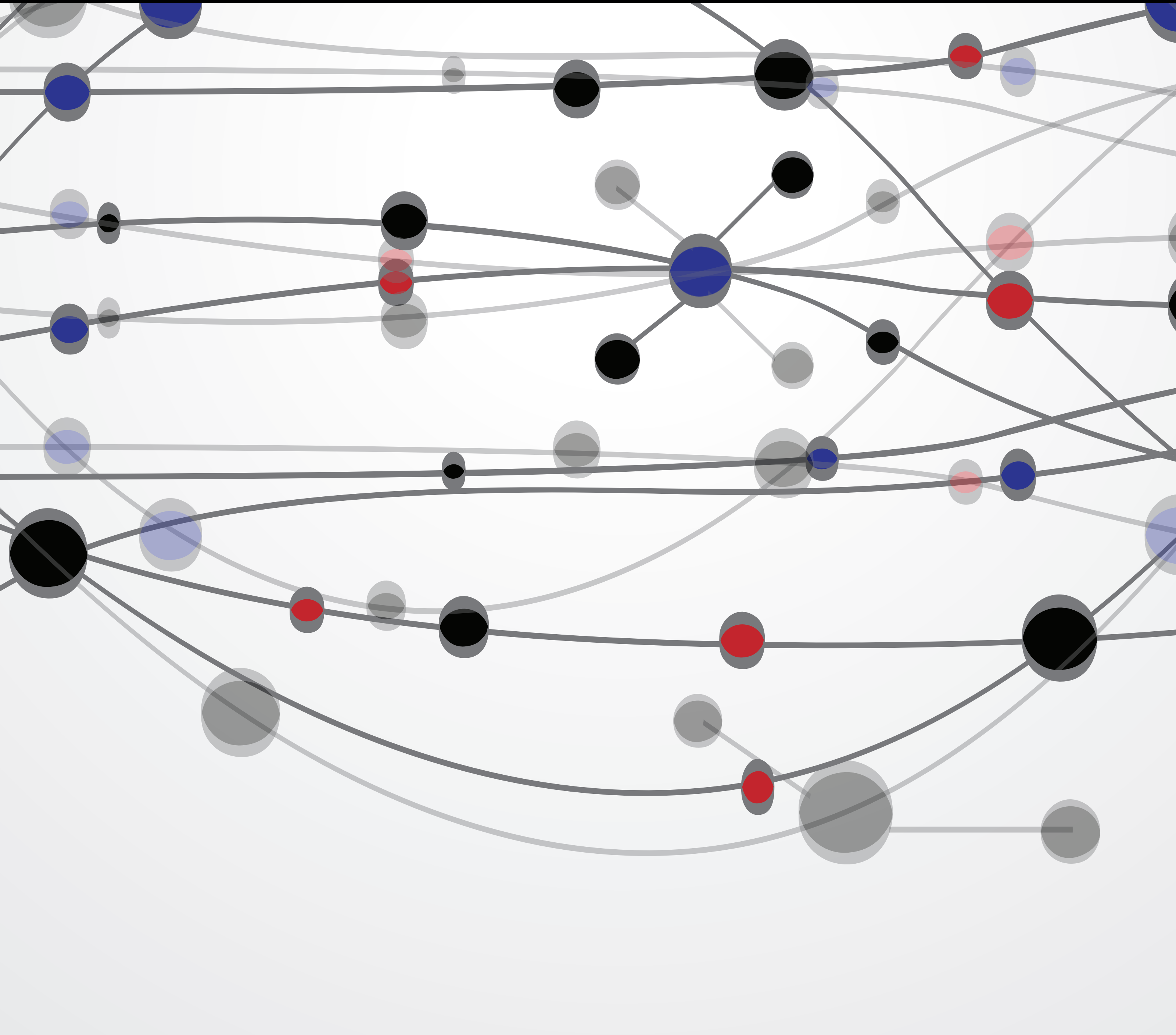
The Scientific World Journal

Machine Learning in Intelligent Video and Automated Monitoring

Guest Editors: Yu-Bo Yuan, Gao Yang David, and Shan Zhao



Machine Learning in Intelligent Video and Automated Monitoring

The Scientific World Journal

Machine Learning in Intelligent Video and Automated Monitoring

Guest Editors: Yu-Bo Yuan, Gao Yang David, and Shan Zhao

Contents

Machine Learning in Intelligent Video and Automated Monitoring, Yu-Bo Yuan, Gao Yang David, and Shan Zhao
Volume 2015, Article ID 570145, 3 pages

Fast Image Search with Locality-Sensitive Hashing and Homogeneous Kernels Map, Jun-yi Li and Jian-hua Li
Volume 2015, Article ID 350676, 9 pages

An Efficient Algorithm for Recognition of Human Actions, Yaser Daanial Khan, Nabeel Sabir Khan, Shoaib Farooq, Adnan Abid, Sher Afzal Khan, Farooq Ahmad, and M. Khalid Mahmood
Volume 2014, Article ID 875879, 11 pages

Part-Based Visual Tracking via Online Weighted P-N Learning, Heng Fan, Jinhai Xiang, Jun Xu, and Honghong Liao
Volume 2014, Article ID 402185, 13 pages

Moving Object Localization Using Optical Flow for Pedestrian Detection from a Moving Vehicle, Joko Hariyono, Van-Dung Hoang, and Kang-Hyun Jo
Volume 2014, Article ID 196415, 8 pages

Estimating Body Related Soft Biometric Traits in Video Frames, Olasimbo Ayodeji Arigbabu, Sharifah Mumtazah Syed Ahmad, Wan Azizun Wan Adnan, Salman Yussof, Vahab Iranmanesh, and Fahad Layth Malallah
Volume 2014, Article ID 460973, 12 pages

Anomaly Detection Based on Local Nearest Neighbor Distance Descriptor in Crowded Scenes, Xing Hu, Shiqiang Hu, Xiaoyu Zhang, Huanlong Zhang, and Lingkun Luo
Volume 2014, Article ID 632575, 12 pages

Real-Time Hand Gesture Recognition Using Finger Segmentation, Zhi-hua Chen, Jung-Tae Kim, Jianning Liang, Jing Zhang, and Yu-Bo Yuan
Volume 2014, Article ID 267872, 9 pages

A DAG Scheduling Scheme on Heterogeneous Computing Systems Using Tuple-Based Chemical Reaction Optimization, Yuyi Jiang, Zhiqing Shao, and Yi Guo
Volume 2014, Article ID 404375, 23 pages

Obscenity Detection Using Haar-Like Features and Gentle Adaboost Classifier, Rashed Mustafa, Yang Min, and Dingju Zhu
Volume 2014, Article ID 753860, 6 pages

Cost-Sensitive Learning for Emotion Robust Speaker Recognition, Dongdong Li, Yingchun Yang, and Weihui Dai
Volume 2014, Article ID 628516, 9 pages

Multilabel Image Annotation Based on Double-Layer PLSA Model, Jing Zhang, Da Li, Weiwei Hu, Zhihua Chen, and Yubo Yuan
Volume 2014, Article ID 494387, 9 pages

An Adaptive Superpixel Based Hand Gesture Tracking and Recognition System,
Hong-Min Zhu and Chi-Man Pun
Volume 2014, Article ID 849069, 12 pages

A High Accuracy Pedestrian Detection System Combining a Cascade AdaBoost Detector and Random Vector Functional-Link Net, Zhihui Wang, Sook Yoon, Shan Juan Xie, Yu Lu, and Dong Sun Park
Volume 2014, Article ID 105089, 7 pages

A Vehicle Detection Algorithm Based on Deep Belief Network, Hai Wang, Yingfeng Cai, and Long Chen
Volume 2014, Article ID 647380, 7 pages

Improving Causality Induction with Category Learning, Yi Guo, Zhihong Wang, and Zhiqing Shao
Volume 2014, Article ID 650147, 8 pages

Novel Web Service Selection Model Based on Discrete Group Search, Jie Zhai, Zhiqing Shao, Yi Guo, and Haiteng Zhang
Volume 2014, Article ID 460593, 6 pages

Web Service Reputation Evaluation Based on QoS Measurement, Haiteng Zhang, Zhiqing Shao, Hong Zheng, and Jie Zhai
Volume 2014, Article ID 373902, 7 pages

Classifying Normal and Abnormal Status Based on Video Recordings of Epileptic Patients, Jing Li, Xiantong Zhen, Xianzeng Liu, and Gaoxiang Ouyang
Volume 2014, Article ID 459636, 6 pages

Iris Recognition Using Image Moments and k-Means Algorithm, Yaser Daanial Khan, Sher Afzal Khan, Farooq Ahmad, and Saeed Islam
Volume 2014, Article ID 723595, 9 pages

Pain Expression Recognition Based on pLSA Model, Shaoping Zhu
Volume 2014, Article ID 736106, 8 pages

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2015, Article ID 570145, 3 pages
http://dx.doi.org/10.1155/2015/570145

Editorial

Machine Learning in Intelligent Video and Automated Monitoring

Yu-Bo Yuan,[1] Gao Yang David,[2] and Shan Zhao[3]

[1] *Department of Computer Science and Engineering, East China University of Science and Technology, Shanghai 200237, China*
[2] *School of Science, Information Technology, and Engineering, Federation University Australia, Mount Helen, VIC 3350, Australia*
[3] *Department of Mathematics, University of Alabama, Tuscaloosa, AL 35487-0350, USA*

Correspondence should be addressed to Yu-Bo Yuan; ybyuan@ecust.edu.cn

Received 14 September 2014; Accepted 14 September 2014

1. Introduction

The primary goal of this special issue is to showcase cutting-edge research on tracking and identifying objects, analyzing motion, and extracting interesting frames from analog or digital video streams automatically. At the same time, we particularly focus on the efficiency of video surveillance systems and machine learning methods which can be used to analyze video and control the machine automatically. Our aim is to unify the machine learning techniques as an integral concept and to highlight the trends in advanced video intelligence and automated monitoring.

2. Contributions and Results

With the developments of computer science, communication technology, and internet engineering, intelligent video surveillance systems have become more and more important in today's life. They can be seen everywhere. Intelligent video surveillance is digital, network-based video surveillance but is different from the general network video surveillance—it is higher-end video surveillance applications. Intelligent video surveillance system can automatically recognize different objects and find anomalies in the monitor screen. Thus, it potentially provides the fastest and best way to alert and provide useful information, which can help security personnel more effectively deal with the crisis. Moreover, intelligent video surveillance system can maximally reduce false positives and false negative phenomena.

The basic information framework can be found in the illustrated Figure 1.

In this special issue, there were 51 submissions from more than 16 countries including China, the United States, Canada, Germany, France, Australia, Japan, Pakistan, Bangladesh, Korea, Malaysia, South Africa, and Romania. Contributions of the accepted papers are summarized as follows.

Based on the studies on the video data sets, innovative results are reported in some papers. Y. D. Khan et al. proposed a sufficiently accurate method while being computationally inexpensive solution to recognize human actions from videos; H. Fan et al. proposed a novel part-based tracking algorithm using online weighted P-N learning; J. Hariyono et al. presented a good pedestrian detection method from a moving vehicle using optical flows and histogram of oriented gradients (HOG); O. A. Arigbabu et al. presented an effective approach for estimating body related soft biometrics and propose a novel approach based on body measurement and artificial neural network for predicting body weight of subjects and incorporate the existing technique on single view metrology for height estimation in videos with low frame rate; X. Hu et al. proposed a novel local nearest neighbor distance (LNND) descriptor for anomaly detection in crowded scenes; R. Mustafa et al. presented a novel method for detecting nipples from pornographic image contents; J. Zhang et al. set up a new image multilabel annotation method based on double-layer probabilistic latent semantic analysis (PLSA); Z. Wang et al. constructed an accurate pedestrian detection system after combining cascade AdaBoost detector

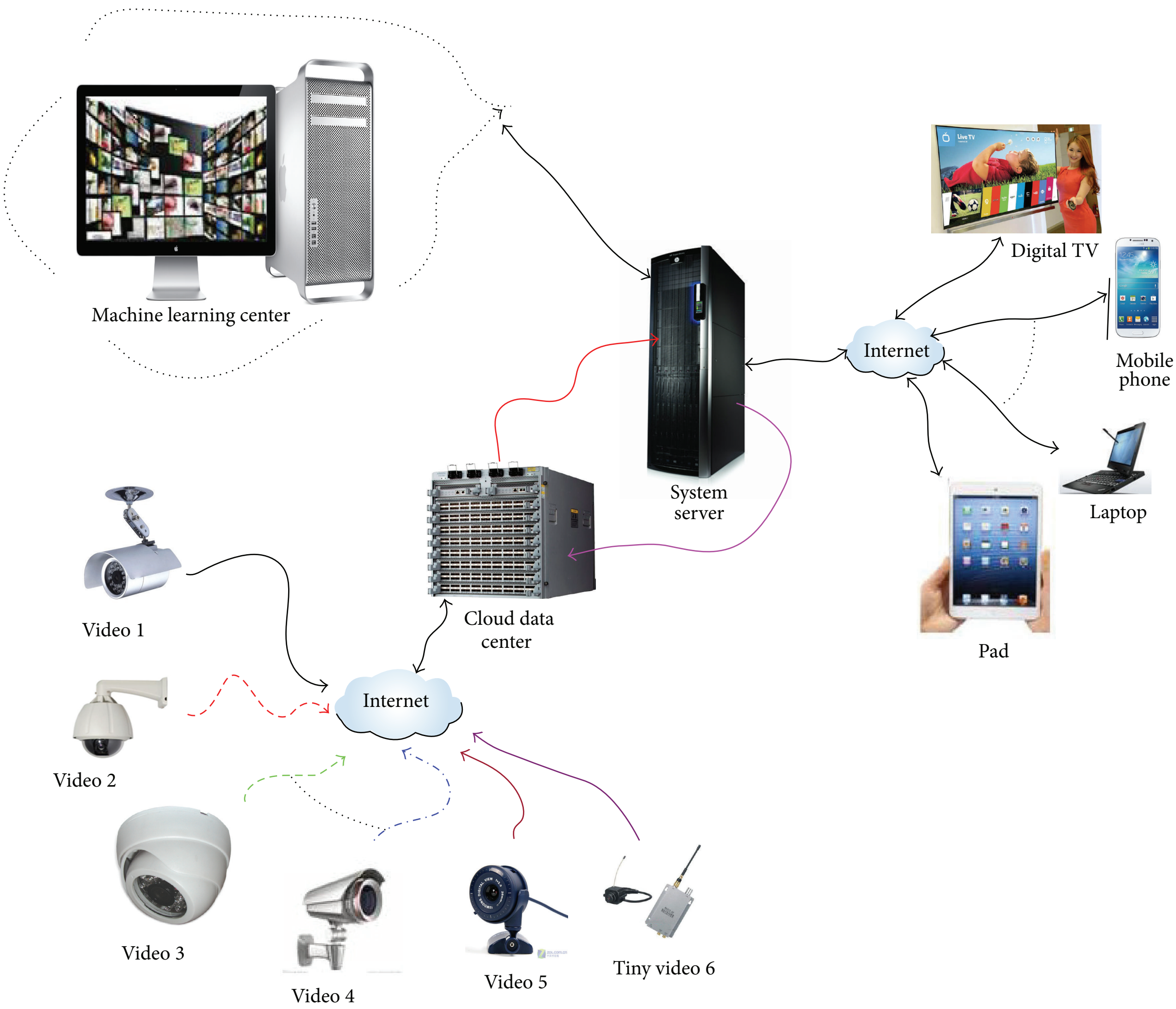


Figure 1: Basic information framework of intelligent video and automated monitoring. More video sources can be collected from video 1 to video 5. In special case, tiny videos are also employed to get video records. The data sets (usually they are big) are submitted to the cloud data center. The services system to handle the videos is the central and important unity. The machine learning system is set up to learn the knowledge or pattern from the special videos according to the users' needs or conditions. In this system, many popular technologies can be employed, such as data mining, manifold learning, kernel learning, image and video processing, and optimization methods and algorithm. In some cases, the machine learning system can transfer the information to users with emails from internet, short messages by mobile communication system, or other dedicated devices (example digital TV sets).

and random vector functional-link net; H. Wang et al. proposed a novel vehicle detection algorithm from 2D deep belief network (2D-DBN) by deep learning framework; J. Li et al. proposed a human action recognition scheme to detect distinct motion patterns and to distinguish the normal status from the abnormal status of epileptic patients after learning video recordings of the movement of the patients with epilepsy; this work is very interesting in the field of health care system of epileptic patients; S. Zhu proposed a new approach to automatically recognize the pain expression from video sequences, which categorize pain as 4 levels: no pain, slight pain, moderate pain, and severe pain.

Some great contributions are devoted to the field of biometrics. Z. Chen et al. presented a novel real-time method for hand gesture recognition using the finger segmentation; D. Li et al. introduced a cost-sensitive learning technology to reweight the probability of test affective utterances in the pitch envelop level and enhanced the robustness in emotion-dependent speaker recognition effectively; H.-M. Zhu and C.-M. Pun proposed an adaptive and robust superpixel based hand gesture tracking system and hand gestures drawn in free air had been recognized from their motion trajectories; Y. Daanial Khan et al. proposed a biometric technique for identification of a person using the iris image.

There are some novel contributions from knowledge management and services selection in the cloud computing. Y. Jiang et al. proposed a tuple molecular structure-based chemical reaction optimization (TMSCRO) method for DAG scheduling on heterogeneous computing systems; Y. Guo et al. proposed a comprehensive causality extraction system

(CL-CIS) integrated with the means of category-learning; J. Zhai et al. proposed a novel cost function and improved the discrete group search optimizer (D-GSO) algorithm; H. Zhang et al. proposed a novel web reputation evaluation method quality of service (QoS) information.

Acknowledgments

The special issue editors would like to thank all of the authors for their contributions and the reviewers for their valuable comments and suggestions. This research has been supported by the National Nature Science Foundations of China (Grants 61402174, 61370174, and 61001200), Open Project Program of the State Key Lab of CAD&CG (Grant no. A1213), and Zhejiang University and Nature Science Foundation of Shanghai Province of China (11ZR1409600).

Yu-Bo Yuan
Gao Yang David
Shan Zhao

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2015, Article ID 350676, 9 pages
http://dx.doi.org/10.1155/2015/350676

Research Article

Fast Image Search with Locality-Sensitive Hashing and Homogeneous Kernels Map

Jun-yi Li[1,2] and Jian-hua Li[1]

[1] *School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China*
[2] *Electrical and Computer Engineering Department, National University of Singapore, Singapore 119077*

Correspondence should be addressed to Jun-yi Li; leejy2006@163.com

Received 30 December 2013; Accepted 19 February 2014

Academic Editors: Y.-B. Yuan and S. Zhao

Fast image search with efficient additive kernels and kernel locality-sensitive hashing has been proposed. As to hold the kernel functions, recent work has probed methods to create locality-sensitive hashing, which guarantee our approach's linear time; however existing methods still do not solve the problem of locality-sensitive hashing (LSH) algorithm and indirectly sacrifice the loss in accuracy of search results in order to allow fast queries. To improve the search accuracy, we show how to apply explicit feature maps into the homogeneous kernels, which help in feature transformation and combine it with kernel locality-sensitive hashing. We prove our method on several large datasets and illustrate that it improves the accuracy relative to commonly used methods and make the task of object classification and, content-based retrieval more fast and accurate.

1. Introduction

In Web 2.0 applications era, we are experiencing the growth of information and confronted with the large amounts of user-based content from internet. As each one can publish and upload their information to the internet, it is urgent for us to handle the information brought by these people from internet. In order to organize and be close to these vision data from Internet, it has caused considerable concern of people. Therefore, the task of fast search and index for large video or image databases is very important and urgent for multimedia information retrieval such as vision search especially now the big data in some certain domains such as travel photo data from the website and social network image data or other image archives.

With the growth of vision data, we focus on two important aspects of problem including nearest neighbor search and similarity metric learning. For metric learning, many of the researchers have proposed some algorithms such as Information-Theoretic metric learning [1]. As for nearest neighbors search, the most common situation and task for us is to locate the most similar image from an image database. Among all the methods, given the similarity of example and query item, the most common method is to find all the vision data among the vision database and then sort them. However time complexity of this algorithm is too large and also impractical. When we handle image or video data, especially, this complexity will not be calculated, because it is very difficult for us to compute the distance of two items in higher dimensional space and also vision datum is sparse, so we cannot complete it by limited time.

Many researchers believe that linear scanning can solve this problem; although we believe it is a common approach and not suitable for computing in large-scale datasets, it promotes the development of ANN. LSH was used in ANN algorithms. To get fast query response for high-dimensional space input vectors [1–5], when using LSH, we will sacrifice the accuracy. To assure a high probability of collision for similar objects, randomized hash function must be computed; this is also referred to in many notable locality-sensitive hashing algorithms [6, 7].

Although, in object similarity search task, the LSH has played an important role, some other issues and problems have been neglected. In image retrieval, recognition, and search tasks, we find that they are very common:

(A) in the sample feature space, traditionally LSH approaches can only let us get a relatively high collision probability for items nearby. As a lot of vision datasets contained much rich information, we can find that the category tags are attached to YouTube and Flickr data and the class labels are attached to Caltech-101 images. However the low-level and high-level of vision samples have great gap, which means that the gap low-level features and high-level semantic information exist. To solve this problem, we intend to utilize the side additional information for constructing hash table;

(B) As to manipulate nonlinear data which is linear inseparable, we commonly use kernel method in vision task because of its popularity. For instance, in vision model, objects are often modeled as BOF and kernel trick is an important approach in classifying these data from low-dimension space to high-dimension space. However, how to create hash table in kernel spaces is a tough problem for us.

To verify our idea, we did several experiments in object search task. For example, we show our results on the Caltech-101 [8] dataset and demonstrate that our approach is superior to the existing hashing methods as our proposed algorithm.

In order to test our algorithm performance on dataset, we design some experiments on certain visual task such as Caltech-101 [8] and demonstrate that the performance of algorithm in our paper is beyond the traditional LSH approaches on the dataset, as hash functions can be calculated beyond many kernels. Arbitrary kernel in ANN is suitable in our scheme; actually we can find that a lot of similarity hashing functions can be accessed in the task of vision search tasks based on content retrieval.

2. Homogeneous Kernel

In our paper, we mainly focus on some similar kernels like intersection, Jensen-Shannon, Hellinger's, and χ^2 kernels. In the fields of machine learning and vision search, we often use these kernels as learning kernels. These kernels have two common attributes: being homogeneous and additive. The idea of kernel signature has been smoothly connected to these kernels in this section. Meanwhile we can use pure functions to represent these kernels. Also these attributes will be applied in Section 3 to obtain kernel feature maps. Through the kernel feature map, we can get their approximate expression.

Homogeneous Kernels. A kernel $k_l : \mathbb{R}_0^+ \times \mathbb{R}_0^+ \rightarrow \mathbb{R}$ is γ-homogenous if

$$\forall m \geq 0 : k_l(ma, mb) = m^{\gamma} k_l(a, b). \tag{1}$$

When $\gamma = 1$, we believe that $k_l(ma, mb)$ is homogeneous. Let $m = 1/\sqrt{ab}$; we can obtain a γ-homogeneous kernel and we can also write the formula as

$$\begin{aligned} k_l(a, b) = m^{-\gamma} k_l(ma, mb) &= (ab)^{\gamma/2} k_l\left(\sqrt{\frac{b}{a}}, \sqrt{\frac{a}{b}}\right) \\ &= (ab)^{\gamma/2} \kappa(\log b - \log a). \end{aligned} \tag{2}$$

Here the pure function

$$\kappa(\lambda) = k_l\left(e^{\lambda/2}, e^{-\lambda/2}\right), \quad \lambda \in \mathbb{R} \tag{3}$$

is called the kernel signature.

Stationary Kernels. A kernel k_s: $\mathbb{R}_0^+ \times \mathbb{R}_0^+ \rightarrow \mathbb{R}$ is called stationary kernels if

$$\forall l \in \mathbb{R} : k_s(l + a, l + b) = k_s(a, b). \tag{4}$$

Let $l = -(a + b)/2$; the $k_s(a, b)$ is represented as

$$\begin{aligned} k_s(a, b) &= k_s(l + a, l + b) \\ &= k_s\left(\frac{b - a}{2}, \frac{a - b}{2}\right) = \kappa(b - a), \end{aligned} \tag{5}$$

$$\kappa(\lambda) = k_s\left(\frac{\lambda}{2}, -\frac{\lambda}{2}\right), \quad \lambda \in \mathbb{R}. \tag{6}$$

Here we call formula (6) kernel feature.

In the field of machine learning or computer vision, most of the homogeneous kernels are composed of the Jensen-Shannon, intersection, χ^2, and Hellinger's kernels. So we can also view them as additive kernels. In the next section, we will focus on these kernels and their kernel maps. Table 1 shows the details [9].

χ^2 Kernel. We define $k(a, b) = 2ab/(a + b)$ as the χ^2 kernel [10, 11]. Here the χ^2 distance is then defined as $D^2(a, b) = \chi^2(a, b)$.

Jensen-Shannon (JS) Kernel. We define $k(a, b) = (a/2)\log_2(a + b)/a + (b/2)\log_2(a + b)/b$ as the JS kernel. Here the JS kernel distance $D^2(a, b)$ can be obtained by $D^2(a, b) = KL(a \mid (a + b)/2) + KL(b \mid (a + b)/2)$, where we import the concept of Kullback-Leibler divergence computed by $KL(a \mid b) = \sum_{l=1}^{d} a_l \log_2(a_l/b_l)$.

Intersection Kernel. We defined $k(a, b) = \min\{a, b\}$ as the intersection kernel [12]. The distance metric $D^2(a, b) = \|a - b\|_1$ is l^1 distance between variants a and b.

Hellinger's Kernel. We defined $k(a, b) = \sqrt{ab}$ as the Hellinger's kernel and specified distance metric $D^2(a, b) = \|\sqrt{a} - \sqrt{b}\|_2^2$ as Hellinger's distance between variants a and b. The function expression $\kappa(\lambda) = 1$ is the signature of the kernel, which is constant.

γ-Homogeneous Parameters. In previous research paper, we can see that the homogeneous kernels are used by parameters $\gamma = 1$ and $\gamma = 2$. When $\gamma = 2$, the kernel becomes $k(a, b) = ab$. Now, in our paper, we can derive the γ-homogeneous kernel by formula (2).

3. Homogeneous Kennel Map

When handling low-dimensional data which is inseparable, we should create kernel feature map $\psi(x)$ for the kernel

Table 1: Common kernels, signature, and their feature maps.

Kernel	$k(a,b)$	Signature $\kappa(\theta)$	$\kappa(w)$	Feature $\psi_\omega(a)$
Hellinger's	$\sqrt{ab}$	1	$\delta(\omega)$	$\sqrt{a}$
χ^2	$\dfrac{2ab}{a+b}$	$\operatorname{sech}\left(\dfrac{\theta}{2}\right)$	$\operatorname{sech}(\pi\omega)$	$e^{iw\log a}\sqrt{a\operatorname{sech}(\pi\omega)}$
Intersection	$\min\{a,b\}$	$e^{-\lvert\theta\rvert/2}$	$\dfrac{2}{\pi}\dfrac{1}{1+4\omega^2}$	$e^{iw\log a}\sqrt{\dfrac{2a}{\pi}\dfrac{1}{1+4\omega^2}}$
JS	$\dfrac{a}{2}\log_2\dfrac{a+b}{a}+\dfrac{b}{2}\log_2\dfrac{a+b}{b}$	$\dfrac{e^{\theta/2}}{2}\log_2\left(1+e^{-\theta}\right)+\dfrac{e^{-\theta/2}}{2}\log_2\left(1+e^{\theta}\right)$	$\dfrac{2}{\log 4}\dfrac{\operatorname{sech}(\pi\omega)}{1+4\omega^2}$	$e^{iw\log a}\sqrt{\dfrac{2}{\log 4}\dfrac{\operatorname{sech}(\pi\omega)}{1+4\omega^2}}$

so that we can map our input data information in low-dimensional space to relatively high-dimensional (Hilbert) information space with $\langle\cdot,\cdot\rangle$:

$$\forall a,b\in R^D : K(a,b)=\langle\psi(a),\psi(b)\rangle. \tag{7}$$

In order to compute the feature maps and get approximate kernel feature maps expression for the homogeneous kernels, we should use Bochner's theorem by expanding the configuration of γ-homogeneous expression. Here we notice that if a homogeneous kernel is Positive Definite [13], its signature will also be Positive Definite expression. The assumption condition is suitable for a stationary kernel. So, depending on formulae (2) and Bochner's theorem (9), we can derive the $k(a,b)$ and closed feature map.

We can compute the kernel density and feature map closed form [9] for most machine learning kernels. Table 1 illustrates the results. Consider

$$\begin{aligned} k(a,b) &= (ab)^{\theta/2}\int_{-\infty}^{+\infty} e^{-iw\lambda}\kappa(\omega)\,d\omega, \quad \theta=\log\frac{b}{a} \\ &= \int_{-\infty}^{+\infty}\left(e^{-iw\log a}\sqrt{a^{\theta}\kappa(\omega)}\right)^{*}\left(e^{-iw\log b}\sqrt{b^{\theta}\kappa(\omega)}\right)d\omega, \end{aligned} \tag{8}$$

$$\psi_w(a)=e^{-iw\log a}\sqrt{a^{\gamma}\kappa(\omega)}. \tag{9}$$

4. Kernelized Locality-Sensitive Hashing

To create and conduct the data association, we take the approach of Kernelized LSH [14] which is also a hash table-based algorithm. KLSH is proposed based on LSH algorithm, which is more efficient and accurate for query search and matching. When searching the input query, the KLSH approach can quickly locate the possible similar and nearest neighbor items in the hash table and match it. In addition, KLSH has another characteristic: traditional LSH methods can only find a part of hashes in the kernel space, while KLSH can locate all the possible hash tables in kernel space. Moreover KLSH has been applied in the vision search tasks by large scale datasets such as Tiny Image and other datasets [14].

Similar to LSH, constructing the hash functions for KLSH has been the key problem for us. That means if we intend to compute the collision probabilities of input query and the database points, we should compute the extent of similarity between them in the database as proposed by [15].

KLSH Principle. Any locality-sensitive hashing algorithm is based on the probability of distribution of hash function clusters. So we should compute the collision probability of a bundle of points, for example, m and n:

$$P_r(h(m)=h(n))=\operatorname{sim}(m,n). \tag{10}$$

We can also view the problem as the issue of computing the similarity of objects between m and n. Here $\operatorname{sim}(m,n)$ in the algorithm is the measure function of calculating the similarity, while $h(m)$ and $h(n)$ are randomly selected from the hash function cluster H. The instinct beyond this is that we find the fact that m and n will collide in the same hash bucket. So those objects which are significantly similar will be more possible to be memorized in the hash table and this eventually results in confliction [1].

We can derive the similarity function expression according to the vector inner product:

$$\operatorname{sim}(m,n)=m^T n. \tag{11}$$

In [15, 16], the definition of LSH function has been extended from formula (10) as

$$h_{\vec{r}}(m)=\begin{cases}1, & \text{if } \vec{r}^{\,T}m\ge 0,\\ 0, & \text{else.}\end{cases} \tag{12}$$

Here we create a random hyper plane vector $\vec{r}$. The distribution of $\vec{r}$ fit has a zero-mean multi-Gaussian $N(0,\Sigma_p)$ distribution. The dimensionality of $\vec{r}$ is the same with the input vector m. This demonstrates that the statistical characteristic of input vector is uniquely matched with each hash function.

Meanwhile this verification has been detailedly reported in the LSH attribute in [17]. When we project on a point m, actually the sigh function we obtain in this process is a hash function and then we repeat it k times; a couple of hashes can be created. We can also call this couple of hashes hash bucket. The hash bucket can be formed as

$$g(m) = \langle h_1(m), \ldots, h_t(m), \ldots, h_k(m) \rangle . \tag{13}$$

From (13), we can see that, after repeating k times, we can get one column of hash bucket (14); then repeating b times, we can finally obtain the hash bucket $g(m)$:

$$g_j(m) = \langle h_1(m), \ldots, h_t(m), \ldots, h_k(m) \rangle \quad (1 < j < b) . \tag{14}$$

When given the value of b, we can get all the the hash functions located in the bucket; we can see the following:

$$g(m) = \left\{ \begin{matrix} h_{1,1\vec{r}}(m) & h_{2,1\vec{r}}(m) & \cdots & h_{s,1\vec{r}}(m) & \cdots & h_{t,1\vec{r}}(m) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ h_{1,j\vec{r}}(p) & h_{2,j\vec{r}}(m) & \cdots & h_{s,j\vec{r}}(m) & \cdots & h_{t,j\vec{r}}(m) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ h_{1,b\vec{r}}(m) & h_{2,b\vec{r}}(m) & \cdots & h_{s,b\vec{r}}(m) & \cdots & h_{t,b\vec{r}}(m) \end{matrix} \right\}, \quad (1 < j < b; 1 < s < t) . \tag{15}$$

Due to the fact that we compute the similarity measure function in high-dimensional kernel space, the similarity function can also be extended and written as

$$\text{sim}\left(\left(m_i, m_j\right)\right) = \kappa\left(m_i, m_j\right) = \phi(m_i)^T \phi\left(m_j\right) . \tag{16}$$

In formula (16), we use kernel function $\phi(m)$ to construct $\kappa(m_i, m_j)$ to complete the kernel mapping for the points of m$_i$ and m_j. And $\phi(m_i)^T\phi(m_j)$ is a product of projection on hash function from the $\Re$ space. The problem is that nothing is known about the data while in kernel space to generate $\vec{r}$ from $N(0, \Sigma_p)$. Therefore, in order to construct the hash function, $\vec{r}$ needs to be created so that we can quickly compute the $\vec{r}^T\phi(m)$ function based on the kernel. Similar to normal $\vec{r}^T$, we could use only the kernel of $\phi(m)$ to approximately compute the function of $\vec{r}^T\phi(m)$. We should select a subset of database to construct $\vec{r}$. By the large number of central limit theory, if we intend to choose parts of database items from the whole database to form the dataset S, the sample of kernel data must be satisfied by the distribution with mean μ and variance Σ. The variable z_a can be written as

$$z_a = \frac{1}{k}\sum_{i \in s} \phi(m_i) . \tag{17}$$

With the growth of variable a, the theory tells us that the vector $\widetilde{z}_a = \sqrt{t}(z_a - \mu)$ has also been satisfied by the distribution of normal Gaussian.

We used the whitening transform to obtain $\vec{r}$:

$$\vec{r} = \Sigma^{-1/2}\widetilde{z}_a . \tag{18}$$

The LSH function has been yielded:

$$h(\phi(m)) = \begin{cases} 1, & \text{if } \phi(m)^T \Sigma^{-1/2}\widetilde{z}_a \geq 0, \\ 0, & \text{else.} \end{cases} \tag{19}$$

As analyzed above, we use kernel function to represent the database data; then the statistical data like variance and mean are uncertain. If we intend to estimate and calculate μ and Σ, we could sample the data from the database by KPCA and eigen decomposition in [18] and we let $\Sigma = V\Lambda V^T$ and $\Sigma^{-1/2} = V\Lambda^{-1/2}V^T$; therefore we can obtain the hash function $h(\phi(m))$:

$$h(\phi(m)) = \text{sign}\left(\phi(m)^T V\Lambda^{-1/2}V^T\widetilde{z}_a\right) . \tag{20}$$

From the above, we can see how to construct the hash function for the kernel matrix input vectors. In this case, we let the kernel matrix input be $K = U\Omega U^T$ by decomposing the K matrix. Here Ω and Λ have the same nonzero eigenvalue; it is also viewed as another form of kernel matrix input. From [18], we compute the projection

$$v_t^T\phi(m) = \sum_{i=1}^{n} \frac{1}{\sqrt{\theta_t}} u_t(i)\,\phi(m_i)^T \phi(m) . \tag{21}$$

Here u_t and v_t are, respectively, the tth eigenvector of the kernel matrix and its covariance matrix.

As mentioned before, we choose n data points from the database to form $\phi(m_i)$; traversing all the t eigenvectors and conducting the computation yields

$$\begin{aligned} h(\phi(m)) &= \phi(m)^T V\Lambda^{-1/2}V^T\widetilde{z}_a \\ &= \sum_{t=1}^{n} \sqrt{\theta_t v_t^T \phi(m)^T}\, v_t^T\widetilde{z}_a . \end{aligned} \tag{22}$$

Substituting (21) into (22) yields

$$\begin{aligned} &\sum_{t=1}^{n} \sqrt{\theta_t}\left(\sum_{i=1}^{n} \frac{1}{\sqrt{\theta_t}} u_t(i)\,\phi(m_i)^T \phi(m)\right) \\ &\quad \cdot \left(\sum_{i=1}^{n} \frac{1}{\sqrt{\theta_t}} u_t(i)\,\phi(m_i)^T \widetilde{z}_a\right) . \end{aligned} \tag{23}$$

```
(1) Process Data
(a) Obtain $K$ matrix from database throughout the $n$ points.
(b) Obtain $e_s$ by randomly sampling a subset from the $\{1, 2, \ldots, n\}$
(c) Project on $a$th subset to obtain $h_a(\phi(m))$.
(d) Obtain $w = K^{1/2} \cdot e_s/a$
(e) Project $w(i)$ onto the points in kernel space
(f) Obtain hash bucket $g_j(m)$
(2) Query Processing
(a) Obtain the same hash bucket in (29) from the database
(b) Use Ann search for query matching.
```

Algorithm 1: KLSH algorithm.

Simplifying (23) yields

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\phi\left(m_i\right)^T\phi\left(m\right)\right)\cdot\left(\phi\left(m_j\right)^T\tilde{z}_a\right)\left(\sum_{t=1}^{n}\frac{1}{\sqrt{\theta_t}}u_t\left(i\right)u_t\left(j\right)\right). \tag{24}$$

Since $K_{ij}^{-1/2} = \sum_{k=1}^{n}(1/\sqrt{\theta_k})u_k(i)u_k(j)$, we further simplify the (24) yields

$$h\left(\phi\left(m\right)\right) = \sum_{i=1}^{n} w\left(i\right)\left(\phi\left(m_i\right)^T\phi\left(m\right)\right), \tag{25}$$

where $w(i) = \sum_{j=1}^{n} K_{ij}^{-1/2}\phi(x_j)^T\tilde{z}_a$.

Through the above derived formula $w(i)$ we can obtain $\vec{r} = \sum_{i=1}^{n} w(i)\phi(x_i)$ which obeys random Gaussian distribution, then we can substitute (17) into $w(i)$:

$$w\left(i\right) = \frac{1}{a}\sum_{j=1}^{n}\sum_{l\in s}K_{ij}^{-1/2}K_{jl}. \tag{26}$$

We neglect the term of $\sqrt{a}$, and finally the simplified $w(i)$ yields (27). e_s represents the unit vector for S.

And therefore hash function for kernel input will finally be

$$w = K^{1/2}\cdot\frac{e_s}{k}, \tag{27}$$

$$h\left(\phi\left(m\right)\right) = \operatorname{sign}\left(\sum_{i=1}^{n} w\left(i\right)\kappa\left(m, m_i\right)\right). \tag{28}$$

κ is the kernel mapping matrix for points m and m_i in space. After several iterations, the hash function will form a hash bucket.

In order to get the suitable parameters in this process, we implement the query matching for several iterations. The detailed algorithm is illustrated finally in Algorithm 1. Consider

$$\begin{aligned} &g_j\left(m\right) \\ &= \left[h_1\left(\phi\left(m\right)\right), h_{2,j}\left(\phi\left(m\right)\right), \ldots, h_{t,j}\left(\phi\left(m\right)\right), \ldots, h_{k,j}\left(\phi\left(m\right)\right)\right], \\ &\qquad\left(1 < l < t\right), \quad \left(1 < j < b\right). \end{aligned} \tag{29}$$

Figure 1: Datasets: Caltech-101 Example.

5. Experimental Result

In the experiment, we proposed the homogenous kernel-hashing algorithm and verified the high efficiency on the dataset. In our scheme, homogenous kernel-KLSH method makes it possible to get the unknown feature embeddings. We use these features to conduct vision search task to locate the most similar items in the database, and the neighbors we find in the task will give their scores on the tags. The method proved to be more effective and accurate than the linear scan search.

In this part, we design and verify our algorithm on the Caltech-101 dataset in Figure 1. Caltech-101 dataset is a benchmark on image recognition and classification, which has 101 categories objects and each category has about 100 images, so 10000 images totally. In recent years, many researchers have done useful research on this dataset such us proposing some important and useful image represent kernels [19]. Also there are many published papers that focused on this dataset, some of which are very valuable and significantly historic. For example, papers [20–22], respectively, state their contribution to the dataset. The author of [21] proposed the matching method for pyramid kernel of images histograms, while Berg [20] proposed and created the CORR kernel of

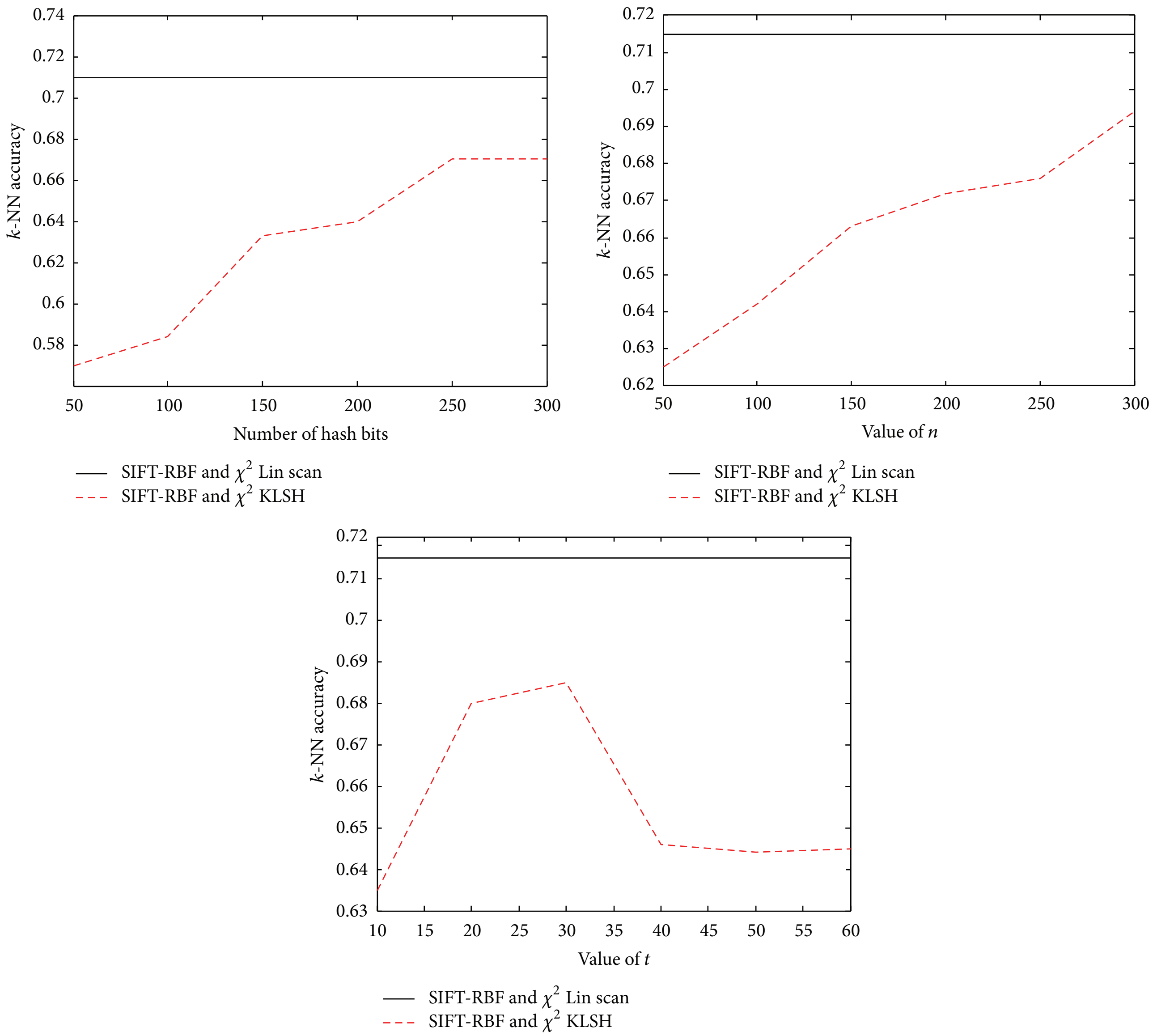


Figure 2: Hashing using a RBF-χ^2 kernel for SIFT based on homogenous kernels χ^2 ($\gamma = 1/2$). We choose $t = 30$, $n = 300$, and $b = 300$ in our experiment.

image local feature using geometric blur for matching local image similarity.

In our paper, we apply our algorithm to complete the vision classification and similar search task. The platform of our benchmark is based on Intel 4 core 3.6 GHZ CPU and 16 GB of memory and 2 TB hard disk.

We used χ^2 kernel for γ-homogeneous kernel maps ($\gamma = 1/2$) and applied the nonlinear RBF-χ^2 kernel designed in [19, 23] to the SIFT-based local feature. Meanwhile we applied and learnt the homogenous kernel map beyond it. Compared with the nonlearnt kernel, our learnt kernel has been more accurate. And we use KNN classifier, respectively, for KLSH and linear scan to compute the accuracy of classification. We also compare it with CORR [24] and the result proves to be better than them, here we use 15 images per class for training task.

From Figure 2 we can see that the growth of parameters is closely related with accuracy. As is seen, the accuracy increased with the increase of n, while it has little relationship with the number of t and b. The value of (n, t, b) is chosen as $n = 300$, $b = 300$, $t = 30$ as the best parameters through a series of experiments.

We find that the combination of these parameters can result in better performance than the large-scale dataset. Meanwhile it can be seen that our approach with homogenous kernel map has higher accuracy than CORR-KLSH with metric learning [25].

Figure 3 illustrates that our method is superior to other existing approaches [25–28] tested on this dataset. Comparing with other kernel classifiers, our classifier with RBF-χ^2 kernel for local features performs better. In Table 2 we can see that the result of ours has higher accuracy with $T = 15$

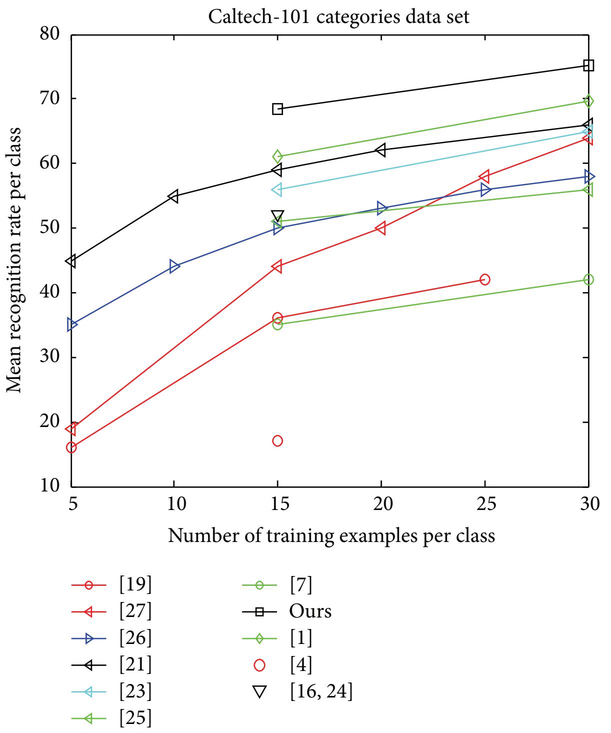


Figure 3: Comparison against existing techniques on the Caltech-101.

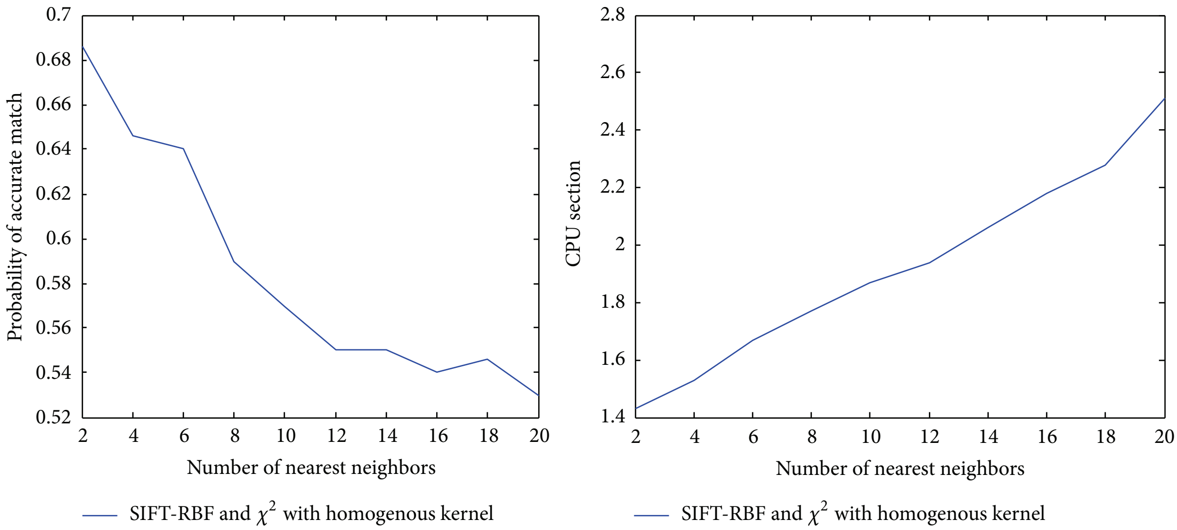


Figure 4: Classification beyond CPU load performance.

and $T = 30$ than other papers' results including better than [24] which obtains the result by 61% for $T = 15$ and 69.6% for $T = 30$. More clearly, it has improved the result by 16% several years ago.

In order to find the best parameters in our experiment for NN search for our scheme, we should take into account the balance between performance and CPU time. Therefore here we conducted to analyze the performance and CPU time of different of k ($k = 2, 3, \ldots, 20$) for NN search. Figure 4 illustrates the accuracy and CPU time by each k in our dataset.

The author of [26] proposed the method by combining KPCA and normal LSH. That means computing hashing beyond the KPCA. However this method has apparent disadvantage because KPCA will bring on the loss of input information although it can reduce the dimensionality in the processing, while KLSH can solve this problem to assure the integrity of input information to compute the LSH. Therefore

Table 2: Accuracy of Caltech-101.

#train	Ours	[18]	[29]	[30]	[31]	[32]	[33]
15	68.5	59.05	56.4	52	51	49.52	44
30	75.2	66.23	64.6	N/A	56	58.23	63

we found that our method has high accuracy and better performance than the algorithm in [26].

6. Conclusions

In our paper, we properly use the concept of homogeneous kernel maps to help us to solve the problem of approximation of those kernels, including those commonly used in machine learning such as χ^2, JS, Hellinger's, and intersection kernels. Combining with the KLSH scheme, it enables us to have access to any kernel function for hashing functions. Although our approach is inferior to linear scan search in time but it can guarantee that the search accuracy will not be affected. Moreover we do not need to consider the distribution of input data; to some extent, it can be applicable for many other databases as Flicker and Tiny Image. Experimental results demonstrate that it is superior to standard KLSH algorithm.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors would like to thank Brian Kulis from UC Berkeley University for helpful comments on experiments and also they should thank Professor Yan Shuicheng for advice on their paper. The work was supported in part by the National Program on Key Basic Research Project (973 Program) 2010CB731403/2010CB731406.

References

[1] B. Kulis, P. Jain, and K. Grauman, "Fast similarity search for learned metrics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2143–2157, 2009.

[2] G. Shakhnarovich, P. Viola, and T. Darrell, "Fast pose estimation with parameter-sensitive hashing," in *Proceedings of the 9th IEEE International Conference on Computer Vision (ICCV '03)*, pp. 750–757, Nice, France, October 2003.

[3] G. Shakhnarovich, T. Darrell, and P. Indyk, Eds., *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice*, The MIT Press, Cambridge, Mass, USA, 2006.

[4] A. Bardera, J. Rigau, I. Boada, M. Feixas, and M. Sbert, "Image segmentation using information bottleneck method," *IEEE Transactions on Image Processing*, vol. 18, no. 7, pp. 1601–1612, 2009.

[5] O. Chum, J. Philbin, and A. Zisserman, "Near duplicate image detection: min-Hash and tf-idf weighting," in *Proceeding of the British Machine Vision Conference (BMVC '08)*, September 2008.

[6] M. S. Charikar, "Similarity estimation techniques from rounding algorithms," in *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC '02)*, pp. 380–388, Montreal, Canada, May 2002.

[7] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *Proceedings of the 25th International Conference on Very Large Data Bases (VLDB '99)*, pp. 518–529, 1999.

[8] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object cateories," in *Proceedings of the Workshop on Generative Model Based Vision*, 2004.

[9] M. Hein and O. Bousquet, "Hilbertian metrics and positive definite kernels on probability measures," in *Proceedings of the 10th Workshop on Artificial Intelligence and Statistics (AISTAT '05)*, January 2005.

[10] J. Puzicha, J. M. Buhmann, Y. Rubner, and C. Tomasi, "Empirical evaluation of dissimilarity measures for color and texture," in *Proceedings of the 7th IEEE International Conference on Computer Vision (ICCV '99)*, pp. 1165–1172, September 1999.

[11] D. R. Martin, C. C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 5, pp. 530–549, 2004.

[12] A. Barla, F. Odone, and A. Verri, "Histogram intersection kernel for image classification," in *Proceedings of the International Conference on Image Processing (ICIP '03)*, pp. 513–516, September 2003.

[13] B. Scholkopf and A. J. Smola, *Learning with Kernels*, The MIT Press, 2002.

[14] B. Kulis and K. Grauman, "Kernelized locality-sensitive hashing for scalable image search," in *Proceeding of the 12th International Conference on Computer Vision (ICCV '09)*, pp. 2130–2137, Kyoto, Japan, October 2009.

[15] Z. Chen, J. Samarabandu, and R. Rodrigo, "Recent advances in simultaneous localization and map-building using computer vision," *Advanced Robotics*, vol. 21, no. 3-4, pp. 233–265, 2007.

[16] M. Weems, *Kernelized locality sensitive hashing for fast landmark association [M.S. thesis]*, 2011.

[17] M. X. Goemans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *Journal of the Association for Computing Machinery*, vol. 42, no. 6, pp. 1115–1145, 1995.

[18] B. Schölkopf, A. Smola, and K. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.

[19] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman, "Multiple Kernels for object detection," in *Proceedings of the 12th International Conference on Computer Vision (ICCV '09)*, pp. 606–613, October 2009.

[20] A. C. Berg, T. L. Berg, and J. Malik, "Shape matching and object recognition using low distortion correspondences," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, 33, p. 26, San Diego, Calif, USA, June 2005.

[21] K. Grauman and T. Darrell, "Discriminative classification with sets of image features," in *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV '05)*, pp. 1458–1465, October 2005.

[22] A. D. Holub, M. Welling, and P. Perona, "Combining generative models and fisher kernels for object recognition," in *Proceedings*

of the 10th IEEE International Conference on Computer Vision (ICCV '05), pp. 136–143, October 2005.

[23] M. Varma and D. Ray, "Learning the discriminative power-invariance trade-off," in *Proceedings of the IEEE 11th International Conference on Computer Vision (ICCV '07)*, October 2007.

[24] H. Zhang, A. C. Berg, M. Maire, and J. Malik, "SVM-KNN: discriminative nearest neighbor classification for visual category recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, pp. 2126–2136, June 2006.

[25] A. Torralba, R. Fergus, and Y. Weiss, "Small codes and large image databases for recognition," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, 2008.

[26] J. Yang, X. Gao, and D. Zhang, "Kernel ICA: an alternative formulation and its application to face recognition," *Pattern Recognition*, vol. 38, no. 10, pp. 1784–1787, 2005.

[27] T. Serre, L. Wolf, and T. Poggio, "Object recognition with features inspired by visual cortex," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, pp. 994–1000, June 2005.

[28] N. Rasiwasia and N. Vasconcelos, "Latent dirichlet allocation models for image classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2665–2679, 2013.

[29] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: spatial pyramid matching for recognizing natural scene categories," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, pp. 2169–2178, June 2006.

[30] A. C. Berg, *Shape matching and object recognition [Ph.D. thesis]*, Computer Science Division, University of California, Berkeley, Calif, USA, 2005.

[31] J. Mutch and D. Lowe, "Multiclass object recognition using sparse, localized features," in *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR '06)*, 2006.

[32] K. Grauman and T. Darrell, "Pyramid match kernels: discriminative classication with sets of image features," Tech. Rep. CSAIL-TR-2006-020, MIT, 2006.

[33] G. Wang, Y. Zhang, and F.-F. Li, "Using dependent regions for object categorization in a generative framework," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, pp. 1597–1604, New York, NY, USA, June 2006.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 875879, 11 pages
http://dx.doi.org/10.1155/2014/875879

Research Article

An Efficient Algorithm for Recognition of Human Actions

Yaser Daanial Khan,[1] Nabeel Sabir Khan,[1] Shoaib Farooq,[1] Adnan Abid,[1] Sher Afzal Khan,[2] Farooq Ahmad,[3] and M. Khalid Mahmood[4]

[1] *School of Science and Technology, University of Management and Technology, Lahore 54000, Pakistan*
[2] *Department of Computer Science, Abdul Wali Khan University, Mardan 23200, Pakistan*
[3] *Faculty of Information Technology, University of Central Punjab, 1-Khayaban-e-Jinnah Road, Johar Town, Lahore 54000, Pakistan*
[4] *Department of Mathematics, University of the Punjab, Lahore 54000, Pakistan*

Correspondence should be addressed to Yaser Daanial Khan; yaser.khan@umt.edu.pk

Received 4 April 2014; Revised 26 June 2014; Accepted 27 June 2014; Published 27 August 2014

Academic Editor: Yu-Bo Yuan

Recognition of human actions is an emerging need. Various researchers have endeavored to provide a solution to this problem. Some of the current state-of-the-art solutions are either inaccurate or computationally intensive while others require human intervention. In this paper a sufficiently accurate while computationally inexpensive solution is provided for the same problem. Image moments which are translation, rotation, and scale invariant are computed for a frame. A dynamic neural network is used to identify the patterns within the stream of image moments and hence recognize actions. Experiments show that the proposed model performs better than other competitive models.

1. Introduction

Human action recognition is an important field in computer vision. The implications of robust human action recognition system, requiring minimal computations, include a wide array of potential applications such as sign language recognition, keyboard or a remote control emulation, human computer interaction, surveillance, and video analysis. Such systems are developed to enable a computer to intelligently recognize a stream of complex human actions being input via a digital camera. It thrives for the need of a multitude of efficiently designed algorithms pertaining to pattern recognition and computer vision. Background noise, camera motion, and position and shape of the object are major impairment factors against the resolution to this problem. This paper presents an efficient and sufficiently accurate algorithm for human action recognition making use of image moments. A comprehensive understanding of image moments describes characteristics information of an image. The proposed system aims to recognize human actions regardless of its position, scale, colors, size, and phase of the human. The paper describes a robust feature extraction and comprehensive classification and training processes. The primary focus is to facilitate video retrieval classified on the basis of featured human action. Inherently it requires methods to identify and discover objects of interest by providing comprehensive features after video segmentation, feature extraction, and feature vector organization. These features are designed such that they are immune to encumbrances such as noise and background view. This calls for methods incessantly capable of tackling video descriptors which are repeatable and most relevant. An efficient computational paradigm for extraction of such descriptors needs to be devised because only those areas of an image are matters of concern, which contain deciphering features. A real-time implementation is realized for detection of nominated human actions. Various researchers have addressed the proposed problem using different methodologies. Tran et al. represent human action as a combination of the movements of the body part [1]. They provide a representation described by a combination of movements of the body part to which a certain action correlate. Their proposed method makes use of polar pattern of the space for representing the movement of the individual parts of the body. In another article Ali and Shah [2] represent kinematic functions computed from optical flow for the recognition of human action in video tribes. These kinematic features represent the spatiotemporal properties of the video. It further performs principal component analysis (PCA) on the kinematic feature volume. Multiple instance learning (MIL) is used for the purpose of classification of human action using succinct data after PCA. Busaryev and Doolittle recognize hand gestures captured from a webcam in real time. Such classification of gestures is applied to control real-world applications. Background subtraction and HSV-based extraction are compared as methods for getting a clean hand image for further analysis. The gesture in each hand image is then determined with Hu moments or a local feature classifier, and each gesture is mapped to a certain keystroke or mouse function [3]. Cao et al. combine multiple features for action detection. They build a novel framework which combines GMM-based representation of STIPs based detection [4]. In order to detect moving objects from complicated backgrounds, Zhang et al. improved Gaussian mixture model, which uses K-means clustering to initialize the model and gets better motion detection results for surveillance videos [5]. They demonstrate that the proposed silhouette representation, namely, "envelope shape," solves the viewpoint problem in surveillance videos. Shao et al. present a method that extracts histogram of oriented gradients (HOG) descriptors corresponding to primitive actions prototype [6]. The output contains only the region of interest (ROI). Using this information the gradient of motion is computed for motion estimation. The gradient vectors are obtained for the partitioning of periodic effect. Once it detects a complete cycle of movement, two key frames are selected for encoding the motion. Finally, the current class action descriptors for the classification of features are extracted while the corresponding classifier is trained offline. Ullah et al. implemented the bag of features (BoF) approach for classification of human actions in realistic videos. The main idea is to segment videos into semantically meaningful regions (both spatially and temporally) and then to compute histogram of local features for each region separately [7, 8].

Certain weaknesses of the recognition algorithm for human actions in video with the kinematic features [8] and multiple instance learning are quite evident. Firstly the kinematic properties selected are not scale, translation, and rotation invariant, as the same action from different angles induces different optical flow. Secondly, occlusion presents serious consequences for the performance of the algorithm, especially in cases where a significant part of the body is closed. Moreover the training step is the slowest part of the algorithm which makes excessive use of memory due to its iterative nature. The method using the HSV model for segmentation of hands will have problems if another object of the same hue is present in the frame. Other methods using sparse representations of human action recognition cannot handle several actions in a video clip. This is because they do not take into account the spatial and temporal orientation of the extracted features. The method discussed in [9, 10] uses color intensities to segment the action by manually selecting a region. Using this approach a region must be selected every time when the scene changes; this undesirably requires human intervention. Furthermore, most of the algorithms work only for a specific illumination; it will fail to give results on high or low illumination. The approach used in [11] is based upon the assumption that each independent observation follows the same distribution. Certainly this approach is bound to fail in case the distribution of the observations is quite the reverse. Although the approach seems to be scale invariant still it is not rotation invariant.

The paper is organized into several sections. Section 1 gives a brief introduction of the problem and the current state of the art. Section 2 gives an overview of the proposed system. Section 3 describes the feature extraction process. Section 4 gives a comprehensive description of the training process. Section 5 provides some detailed results from the model while Section 6 adds some conclusive remarks.

2. An Overview of the Proposed System

The system is designed to retrieve semantically similar video clips for a given criterion video clip, from a large set of semantically different videos. The video dataset contains features of every proposed action and on query, video features will be extracted and matched with the stored features in the feature library. Since gestures are sequence of postures (static frames), therefore the system is expected to recognize gestures by identifying constituent postures one by one. Ultimately a temporal classifier is used to classify the input stream of spatial postures into an action. Figure 1 shows the flow of the initial training process. Firstly, individual frames are extracted from the video input. Secondly each extracted frame is preprocessed to make it suitable for moment extraction. These moments form a feature vector which is initially used for training of the system.

The system is exhaustively trained using the training process described later. A sufficiently trained system is deemed appropriate for classification of the proposed actions. Figure 2 shows the process used for classification of human actions.

Extracted features from a live video feed are fed into a trained dynamic neural network (DNN) for the purpose of classification. The neural network classifies the action performed in a few successive frames. The dynamic neural network is designed such that its behavior varies temporally based on the video input.

3. Preprocessing and Feature Extraction

Initially a number of preprocessing steps must be performed on video frames before moments based features are extracted. Computations of moments require that the image is of monochrome nature. The chromatic frame extracted from the video is firstly binarized using a threshold. The threshold is carefully chosen based on the mean illumination level of the frame. Mean illumination is computed by taking the mean of luminosity value of each pixel in the frame. Once binarized, the image will hold either black or white pixels. Further to remove noise and other impairments dilation and erosion is performed [12]. Figures 3 and 4 show the result of this process on a sample frame.

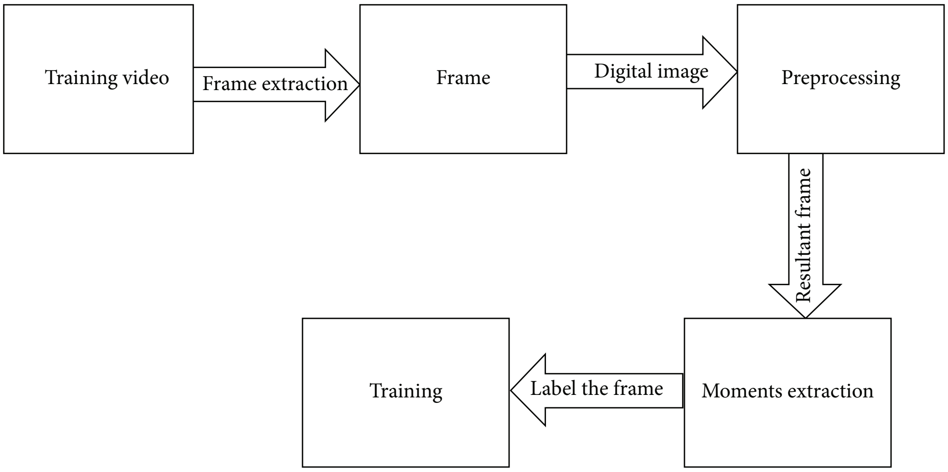


Figure 1: The steps of the training process.

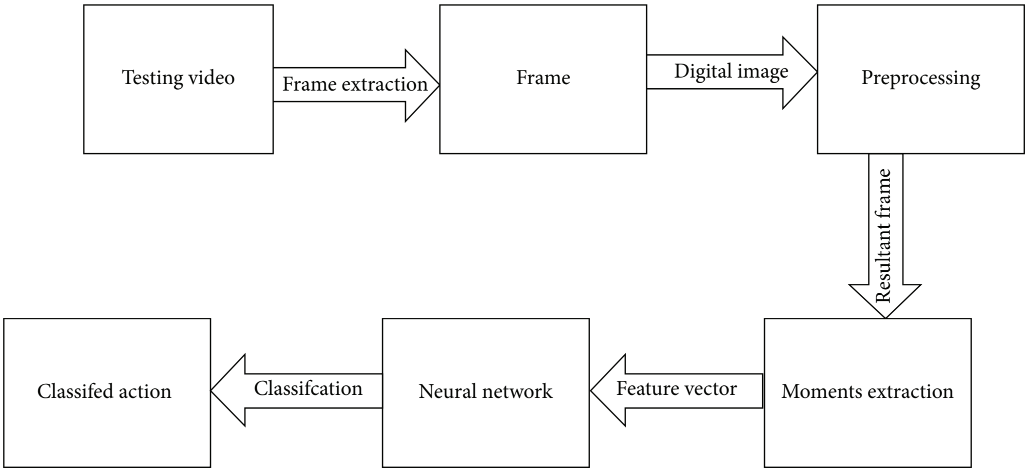


Figure 2: The classification process.

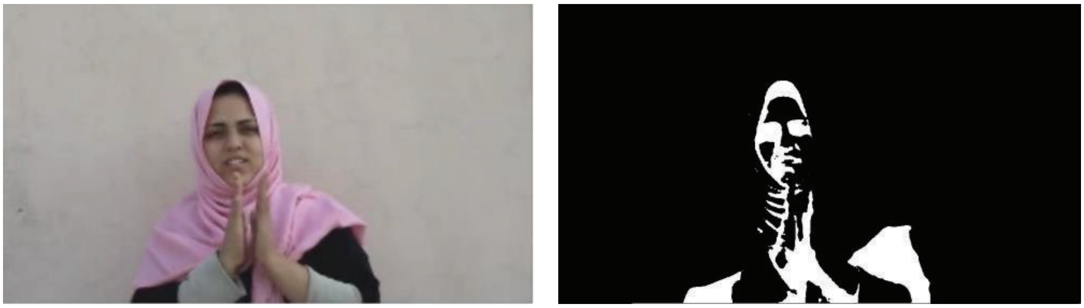
Before any intricate processing is performed on the data set, the background is removed from each frame. Here two alternate approaches are adopted. In the first approach initial few frames are captured without any foreground action containing only the background. Any frame from this initial footage is used as a representative. This frame is subtracted from each frame containing foreground to obtain the filtered foreground. In the other approach each successive frame is XORed. The resultant frame represents the change in action during the period of the latter frame. The difference frame in this case also excludes the background.

3.1. Moments Extraction. Moments are scalar quantities which are used to categorize the shape and its features. They are computed from the shape boundary and its entire region. The concept of moments in images is quite similar to the concept of moments in physics. One major difference between the two is that image moments are inherently two-dimensional in nature. The resolution to the proposed problem is sought with the help of various moments such as raw, central, and scale invariant and rotation invariant moments along with certain corporeal properties of the image like the centroid and eccentricity. Invariant moments are those moments which are impervious to certain deformations in the shape and are most suited for comparison between two images. The scale, location, and rotation invariant moments are used to extract features regardless of size, position, and rotation, respectively.

3.2. Raw Moments. Raw moments are calculated along the origin of the image. Let $f(x, y)$ be a function that defines an image where (x, y) are any arbitrary coordinates of the image. In case of two-dimensional continuous signal the raw moment function M_{pq} for the moment of order $(p+q)$ is given as

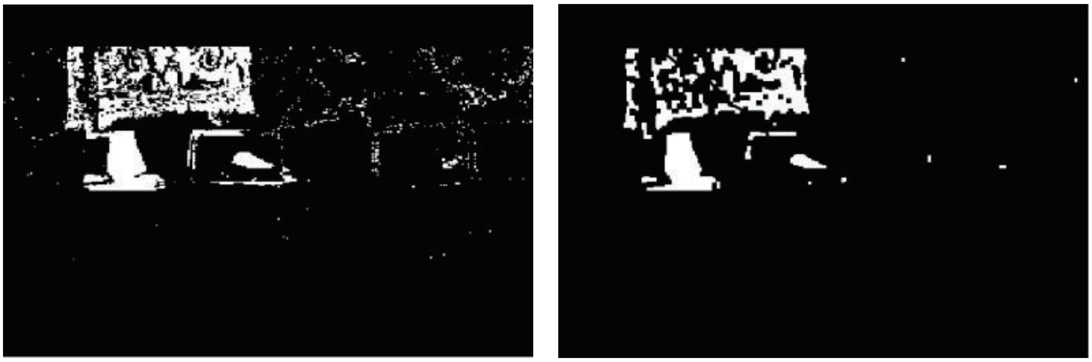
$$M_{pq} = \sum_x \sum_y x^p y^q f(x, y), \tag{1}$$

where $f(x, y)$ is xth pixel along x-axis and yth pixel along y-axis and p, q are the pth and qth indices of the moments. These moments are computed throughout the span of the



(a) (b)

Figure 3: (a) The original frame. (b) The frame after.



(a) (b)

Figure 4: (a) The binarized image. (b) The same image after erosion dilation operations.

image. The raw moments provide information about properties like area and size of the image; for example, the moment M_{00} will give the area of object.

3.3. Central Moments. The moments which are invariant to translation of objects in an image are called central moments as they are computed along the centroid rather than the origin. From the equation of raw moments central moments are calculated such that the first two order moments from (18), that is, M_{10} and M_{01}, are used to locate the centroid of the image.

Let $f(x, y)$ be a digital image; then reducing the coordinates in previous equation by center of gravity ($\overline{x}$ and $\overline{y}$) of the object we get

$$\mu_{pq} = \sum_{x}\sum_{y}(x - \overline{x})^{p}(y - \overline{y})^{q} f(x, y). \tag{2}$$

The coordinates of the center of mass $(\overline{x}, \overline{y})$ are the point of intersection of the lines $x = \overline{x}$ and $y = \overline{y}$, parallel to the x and y-axis, where the first order moments are zero. The coordinates of the center of gravity are the components of the centroid given as follows:

$$\overline{x} = \frac{M_{10}}{M_{00}}, \qquad \overline{y} = \frac{M_{01}}{M_{00}}, \tag{3}$$

while

$$\begin{gathered} \mu_{00} = M_{00}, \\ \mu_{01} = \mu_{10} = 0. \end{gathered} \tag{4}$$

Moments of order up to three are simplified in [13] and are given as follows:

$$\begin{gathered} \mu_{11} = M_{11} - \overline{x}M_{01} = M_{11} - \overline{y}M_{10}, \\ \mu_{20} = M_{20} - \overline{x}M_{10}, \\ \mu_{20} = M_{20} - \overline{x}M_{10}, \\ \mu_{21} = M_{21} - 2\overline{x}M_{11} - \overline{x}M_{20} + 2\overline{x^2}M_{01}, \\ \mu_{12} = M_{12} - 2\overline{y}M_{11} - \overline{x}M_{02} + 2\overline{y^2}M_{10}, \\ \mu_{30} = M_{30} - 3\overline{x}M_{20} - 2\overline{x^2}M_{10}, \\ \mu_{03} = M_{30} - 3\overline{y}M_{02} - 2\overline{y^2}M_{10}. \end{gathered} \tag{5}$$

It is shown in [14] that the generalized form of central moments is

$$\mu_{pq} = \sum_{m}^{p}\sum_{n}^{q}\binom{p}{m}\binom{q}{n}(-\overline{x})^{(p-m)}(-\overline{y})^{(q-n)}M_{mn}. \tag{6}$$

The main advantage of central moments is their invariances to translations of the object. Therefore they are suited well to describe the form or shape of the object while the centroid pertains to information about the location of the object.

3.4. Scale Invariant Moments. The raw moments and the central moments depend on the size of object. This creates a problem when the same object is compared but both the images are captured from different distances. To deal with this encumbrance scale invariant moments are calculated. Moments μ_{ij} are invariant to changes in scale and are obtained by dividing the central moment by scaled (00)th moment as given in the following:

$$\mu_{ij} = \frac{\mu_{ij}}{\mu_{00}^{(1+(i+j)/2)}}. \tag{7}$$

3.5. Rotational Invariant Moments. Rotational moments are those moments which are invariant to changes in scale and also in rotation. Most frequently used are the Hu set of invariant moments:

$$\begin{gathered}
I_1 = \eta_{20} + \eta_{02}, \qquad I_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2, \\
I_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2, \\
I_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} - \eta_{03})^2, \\
\begin{aligned}
I_5 = {} & (\eta_{30} + 3\eta_{12})(\eta_{30} - \eta_{12}) \\
& \times \left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] \\
& + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \\
& \times \left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right],
\end{aligned} \\
\begin{aligned}
I_6 = {} & (\eta_{20} - \eta_{02})\left[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] \\
& + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}),
\end{aligned} \\
\begin{aligned}
I_7 = {} & (\eta_{30} + 3\eta_{12})(\eta_{30} - \eta_{12}) \\
& \times \left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] \\
& - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03}) \\
& \times \left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right].
\end{aligned}
\end{gathered} \tag{8}$$

All the moments discussed in this section are computed for each frame. The collection of the moments is used as a feature vector. This feature vector provides characteristic information about the contents of the frame in numerical form. The variation of patterns formed by periodic frames in a video defines the action being performed. Further a framework is presented capable of recognizing the hidden patterns within the stream of feature vectors for each defined human action [14–16].

4. Training the Network

A drawback of supervised training is that training data needs to be labeled. Initially each frame in the training video is assigned a class number. A specific number is assigned to each class, inherently; the frame related to any class will be given a class number. A target matrix is organized such that each column represents a label of a frame within the training data. Another input matrix is correspondingly organized in which each column contains the extracted moments of the frame. Further a neural network is designed such that neurons in the input layer could be clamped to each element in the obtained feature vector. The neurons in hidden layer are variable and will be changed to fine-tune the results, while the output layer has neurons equivalent to the number of designated classes. Moreover the network is of recurrent nature; that is, the output at output layer is recurrently clamped with the input as shown in Figure 5. Initially all the inputs and outputs of hidden layer are assigned random weights. Back propagation algorithm is used to adjust these weights and converge the output. This algorithm makes use of the sigmoid function (σ) for the training purpose given as

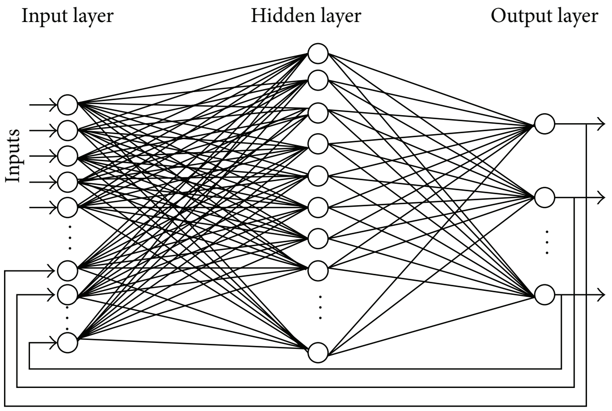


FIGURE 5: A recurrent neural network, notice the output being recurrently fed into the input layer.

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \tag{9}$$

The derivative of this function is given as

$$\frac{d\sigma(x)}{dx} = \sigma(x) \cdot (1 - \sigma(x)). \tag{10}$$

The feature vector for each frame is fed into the input layer and the output is computed. Initially randomly assigned weights are used for each edge. The difference between the actual and labeled output determines the error. Back propagation technique back-tracks this error and readjusts the weights so that the error is reduced. The weights are adjusted in a backward direction. In case of proposed network weights are adjusted in the hidden layer and then the same is done for input layer. Several iterations are performed for each input until convergence is achieved and no appreciable change in weights is perceived.

Let the weight of an edge between an arbitrary neuron i in input layer and an arbitrary neuron in hidden layer j be given as w_{ij} while the weight of an edge between an arbitrary neuron j in hidden and arbitrary neuron k in output layer is given as w_{jk}. For each neuron in input layer the following operations are performed:

$$\psi_j = \sum_{i=1}^{N} x_i w_{ij} - \tau_j, \qquad \chi_j = \frac{1}{1 + e^{-\psi_j}}, \tag{11}$$

where N represents the number of input layer neurons and τ_j the threshold used by the jth neuron in the hidden layer. Outputs at the hidden layer are given as follows:

$$\psi_k = \sum_{j=1}^{M} \chi_j w_{jk} - \tau_k, \qquad \chi_k = \frac{1}{1 + e^{-\psi_k}}, \tag{12}$$

while τ_k is the threshold of the kth neuron at the output layer, χ_k is the neuron output, and M is the number of neurons in hidden layer.

The obtained feature vector for a single video frame is clamped to the neural network in order to produce an output z^k. Here the difference between the desired and actual output is computed as the error ϵ_k given as

$$\epsilon_k = \lambda_k - z_k, \tag{13}$$

while λ_k is the desired output.

Further error gradient is used to determine the incremental iterative change in the weight so that the actual output approaches the expected output. The error gradient is defined as the product of error and the rate of change in the actual output. Analytically it is given as

$$\delta_k = \epsilon_k \frac{\partial \chi_k}{\partial \psi_k}. \tag{14}$$

Using the partial derivative of χ_k and putting it in above equation the following equation is formed:

$$\delta_j = \epsilon_k \chi_j \cdot \left(1 - \chi_j\right). \tag{15}$$

The weight of edges between input and hidden layer also needs to be adjusted. For this purpose the error gradient for hidden layer should also be calculated. In the back propagation techniques the errors are back-tracked. The error gradient at output layer is primarily used to calculate error gradient at hidden layer. Here, the following equation is used to calculate it:

$$\delta_j = \chi_j \cdot \left(1 - \chi_j\right) \sum_{k=1}^{M} w_{jk} \delta_k. \tag{16}$$

Using these error gradients the renewed weights for neuron at each layer are computed. The following equations are used:

$$\begin{aligned} w_{ij} &= w_{ij} + \gamma \cdot x_i \cdot \delta_j, \\ w_{jk} &= w_{jk} + \gamma \cdot \chi_j \cdot \delta_k, \end{aligned} \tag{17}$$

where γ is the learning rate. Generally it is a tiny positive value lesser than 1 and is adjustable according to the learning behavior of the network. Similarly the threshold used for computing the renewed weights should also be recalculated for the next iteration. The following equations are used to recalculate the weights:

$$\theta_k = \theta_k + \gamma \cdot (-1) \cdot \delta_k, \tag{18}$$

$$\theta_j = \theta_j + \gamma \cdot (-1) \cdot \delta_j. \tag{19}$$

Equations (18) and (19) represent the threshold for arbitrary neuron in output and hidden layer, respectively. This method of gradient descent is quite effective and works splendidly for almost all sorts of problems. It has the capability to minimize the error and optimize the network to provide accurate results. Although the training process is iterative, it ultimately needs to terminate. This termination condition is indicated by the convergence of results. The result is said to have converged when no appreciable change in weights is possible. This termination condition is determined using the mean square error given as

$$E = \frac{1}{M} \sum_{k=1}^{M} (\lambda_k - z_k)^2. \tag{20}$$

In the current problem a learning rate of $\alpha = 0.0001$ was used. The output of a recurrent neural network is not only dependent on the current input but also dependent on the previous output. This recurrent nature of these networks makes them useful for problems which require a continuous input of dynamic data changing temporally as well. Identification of an action is not necessarily dependent on a single frame; rather previous and subsequent frames may also tell a story. Hence the use of recurrent network caters for the need for previous temporal information [17].

5. Results and Discussion

A large database of videos was collected containing hundreds of videos of varied length. Each video contained actions like

(i) walking,

(ii) clapping,

(iii) hugging,

(iv) single hand waving,

(v) double hand waving,

(vi) hand shaking.

Figure 6 shows some of the sample actions. Several videos containing different actions were taken under varied conditions in terms of illumination, location, and background. Frame by frame extraction from these videos is performed. Each frame is firstly labeled in accordance with its semantic content manually. Each stream of frames belonging to a specified class is bifurcated and kept in a folder maintaining its sequence. Hence several samples of each action are segmented from the videos manually. Each sample is a stream of videos belonging to specific action. In the next step the background or the effect of background is removed from the frame. Two different strategies are followed for this purpose. With the first method, background is removed by firstly taking a blank frame which only contains the background

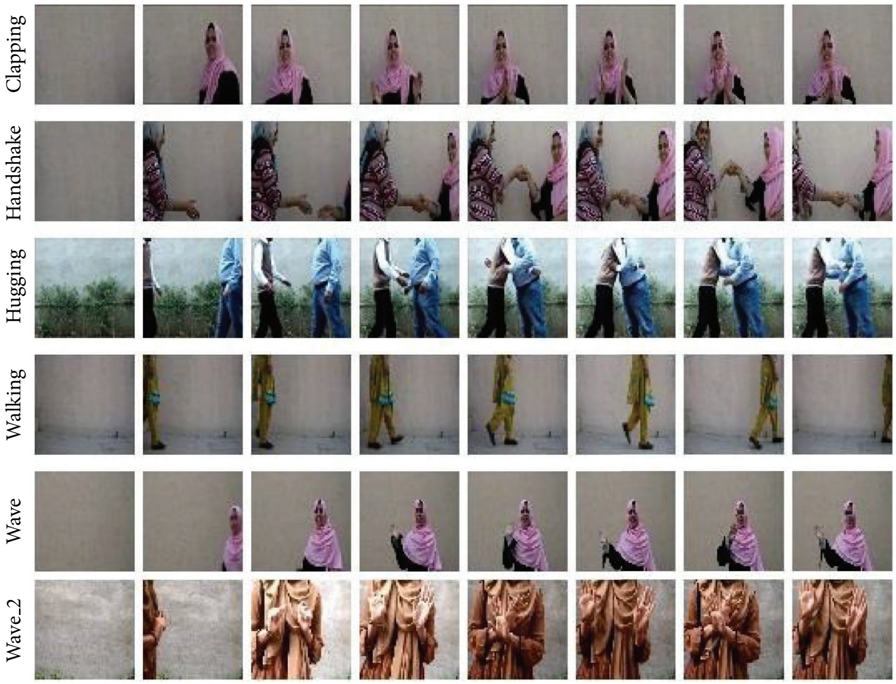


Figure 6: Action database: examples of sequences corresponding to different types of actions and scenarios.

Table 1: The numerical comparison of raw moments for each of the actions.

	Clapping	Handshake	Hugging	Walking	Wave 1	Wave 2
Spatial/raw moments	$1.75E+03$	$4.90E+03$	$7.31E+03$	$6.70E+01$	$1.30E+03$	$9.47E+02$
	$4.46E+05$	$1.07E+06$	$1.32E+06$	$1.59E+04$	$2.98E+05$	$1.15E+05$
	$3.91E+05$	$8.59E+05$	$1.55E+06$	$6.94E+03$	$4.10E+05$	$3.23E+05$
	$9.85E+07$	$2.13E+08$	$3.10E+08$	$1.33E+06$	$9.39E+07$	$3.94E+07$
	$1.16E+08$	$2.63E+08$	$2.82E+08$	$4.16E+06$	$7.34E+07$	$1.63E+07$
	$9.87E+07$	$2.51E+08$	$5.95E+08$	$2.83E+06$	$1.34E+08$	$1.13E+08$
	$2.53E+10$	$5.62E+10$	$6.85E+10$	$3.18E+08$	$2.32E+10$	$5.53E+09$
	$2.48E+10$	$6.61E+10$	$1.20E+11$	$4.80E+08$	$3.08E+10$	$1.37E+10$
	$3.05E+10$	$6.86E+10$	$6.67E+10$	$1.14E+09$	$1.91E+10$	$2.70E+09$
	$2.66E+10$	$9.24E+10$	$2.53E+11$	$1.37E+09$	$4.49E+10$	$3.99E+10$

and then subtracting this frame from the one containing a foreground. As a result background will be eliminated. The other method used for this purpose takes the difference of two successive frames. The resultant frame will contain just the change that occurred due to motion dynamics. Once the effect of background has been countered then for all resultant frames a corresponding feature vector is formed. The feature vector of a frame contains the raw, central, scale, and rotation invariant moments of the image besides its centroid and eccentricity. Tables 1, 2, 3, 4, and 5 show the quantified values of these features. The computed vector for each frame is fed into the recurrent neural network, iteratively training the network as described previously. The training stops when the mean square error is minimized. Not all the database is used for the training purpose. One-fourth of database samples are not used for training; rather they are reserved for testing. At the point when the model has been sufficiently trained it is time to test it. The remaining samples are similarly transformed into feature vectors and fed into the trained model. The accuracy of the model is based on its ability to correctly identify these untrained samples. Figure 7 represents the confusion matrix which shows that the overall accuracy of the system is 80.8%. Also it is noticed that the system is better able to recognize medial frames in an action rather than initial or terminal ones. The accuracy of the system is further increased to 95% if only the accuracy of recognition for the medial frames is considered.

Various experiments were conducted to verify the accuracy, efficiency, and effectiveness of the system in comparison with other competitive models. A technique described in [18] extracts the features in terms of spatial as well as temporal terms. These features are used to train SVM and hence classify the video. The authors in [19] use a technique which significantly reduces the training overhead. A patch based

Table 2: The numerical comparison of central moments for each of the actions.

	Clapping	Handshake	Hugging	Walking	Wave 1	Wave 2
Central moments	1.75E + 03	4.90E + 03	7.31E + 03	6.70E + 01	1.30E + 03	9.47E + 02
	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	−8.96E + 05	2.57E + 07	3.14E + 07	−3.23E + 05	7.40E + 04	−1.81E + 04
	2.25E + 06	2.94E + 07	4.52E + 07	3.77E + 05	5.14E + 06	2.21E + 06
	1.15E + 07	1.01E + 08	2.68E + 08	2.11E + 06	4.60E + 06	2.28E + 06
	9.36E + 06	−1.26E + 09	−2.57E + 09	4.05E + 07	8.61E + 07	−1.40E + 07
	9.23E + 07	2.15E + 09	−9.86E + 08	−1.26E + 08	1.37E + 08	9.95E + 06
	−3.76E + 07	−1.80E + 09	−4.80E + 08	−2.51E + 07	−7.32E + 07	1.84E + 08
	−5.31E + 08	1.30E + 10	1.40E + 10	6.42E + 08	−1.91E + 08	−1.50E + 08

Table 3: The numerical comparison of image orientation for each of the actions.

	Clapping	Handshake	Hugging	Walking	Wave 1	Wave 2
Image orientation	−5.11E + 02	5.24E + 03	4.29E + 03	−4.82E + 03	5.69E + 01	−1.91E + 01
	1.29E + 03	6.00E + 03	6.18E + 03	5.63E + 03	3.95E + 03	2.33E + 03
	6.56E + 03	2.06E + 04	3.66E + 04	3.15E + 04	3.53E + 03	2.40E + 03
	−9.58E − 02	3.12E − 01	1.37E − 01	−1.78E − 01	−1.32E − 01	−2.49E − 01

Table 4: The numerical comparison of scale invariant moments for each of the actions.

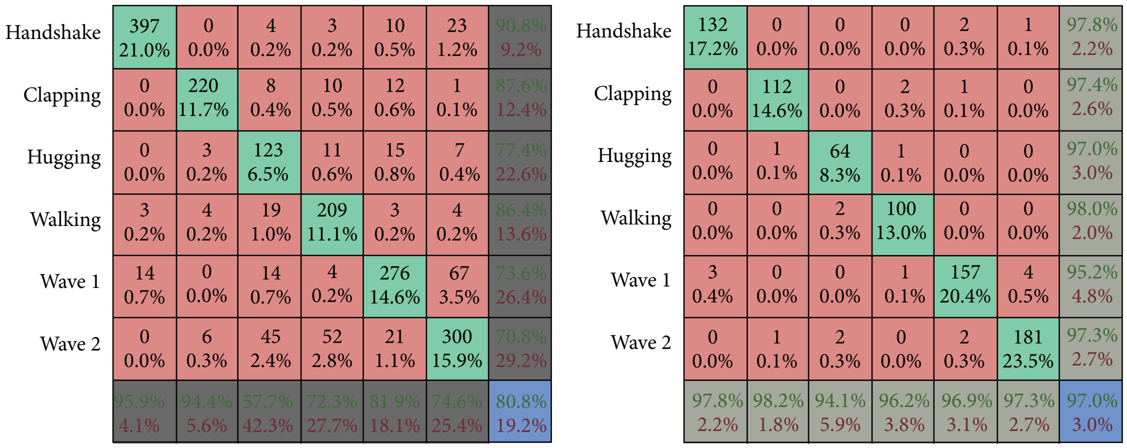
	Clapping	Handshake	Hugging	Walking	Wave 1	Wave 2
Scale invariant	−2.91E − 01	1.07E + 00	5.87E − 01	−7.20E + 01	4.37E − 02	−2.02E − 02
	7.33E − 01	1.22E + 00	8.46E − 01	8.40E + 01	3.04E + 00	2.46E + 00
	3.74E + 00	4.20E + 00	5.01E + 00	4.70E + 02	2.72E + 00	2.54E + 00
	3.04E + 00	−5.27E + 01	−4.81E + 01	9.01E + 03	5.09E + 01	−1.56E + 01
	3.00E + 01	8.99E + 01	−1.84E + 01	−2.81E + 04	8.07E + 01	1.11E + 01
	−1.22E + 01	−7.50E + 01	−8.99E + 00	−5.59E + 03	−4.33E + 01	2.05E + 02
	−1.73E + 02	5.41E + 02	2.61E + 02	1.43E + 05	−1.13E + 02	−1.67E + 02

Table 5: The numerical comparison of rotation invariant moments for each of the actions.

	Clapping	Handshake	Hugging	Walking	Wave 1	Wave 2
Rotation invariants	4.47E + 00	5.43E + 00	5.86E + 00	5.54E + 02	5.75E + 00	5.00E + 00
	9.37E + 00	1.34E + 01	1.87E + 01	1.70E + 05	1.12E − 01	7.13E − 03
	4.35E + 04	6.08E + 05	1.67E + 05	1.97E + 10	1.52E + 05	4.41E + 04
	2.91E + 04	2.39E + 05	4.61E + 04	2.43E + 10	5.23E + 03	8.02E + 04
	1.02E + 09	8.50E + 10	3.89E + 09	5.32E + 20	1.17E + 08	−6.09E + 09
	8.91E + 04	7.40E + 05	1.72E + 05	9.97E + 12	−1.19E + 03	2.21E + 03
	−2.05E + 08	3.26E + 10	1.08E + 09	3.07E + 19	1.07E + 08	−1.96E + 09
	7.61E + 02	2.33E + 05	5.05E + 04	3.95E + 11	−6.43E + 02	3.20E + 03

motion descriptor and matching technique is developed by the author. A concept of transferrable learning distance is introduced which extracts the generic obscure knowledge within patches and is used to identify actions in newer videos. Both of these techniques were implemented. The accuracy of the proposed technique was evaluated in comparison with both these techniques. Figure 8 shows the obtained results while using the assembled action database. It can be seen that the proposed technique performs reasonably well and is more consistent as compared to other competitive techniques.

Figure 9 somehow depicts the efficiency of the system, illustrating the number of frames against time required to classify an action. The graph shows that with the increasing number of frames the computed time for each frame length remains constant. Time required for recognition does not seem to rapidly increase if the number of frames is rapidly



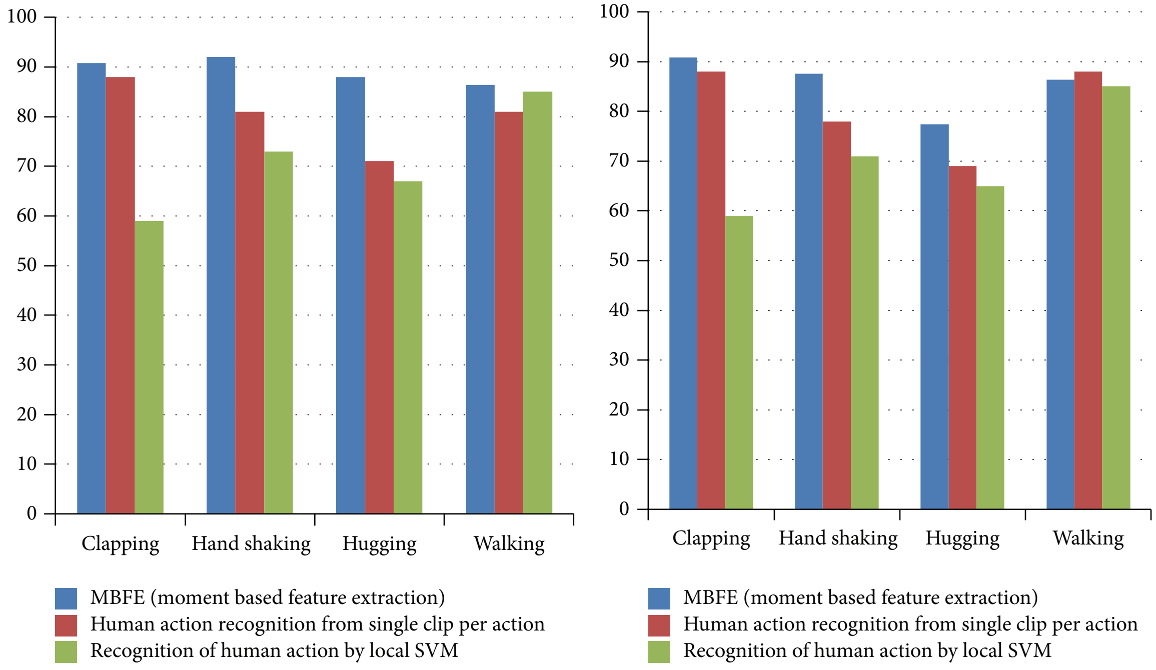
Handshake	397 21.0%	0 0.0%	4 0.2%	3 0.2%	10 0.5%	23 1.2%	90.8% 9.2%
Clapping	0 0.0%	220 11.7%	8 0.4%	10 0.5%	12 0.6%	1 0.1%	87.6% 12.4%
Hugging	0 0.0%	3 0.2%	123 6.5%	11 0.6%	15 0.8%	7 0.4%	77.4% 22.6%
Walking	3 0.2%	4 0.2%	19 1.0%	209 11.1%	3 0.2%	4 0.2%	86.4% 13.6%
Wave 1	14 0.7%	0 0.0%	14 0.7%	4 0.2%	276 14.6%	67 3.5%	73.6% 26.4%
Wave 2	0 0.0%	6 0.3%	45 2.4%	52 2.8%	21 1.1%	300 15.9%	70.8% 29.2%
	95.9% 4.1%	94.4% 5.6%	57.7% 42.3%	72.3% 27.7%	81.9% 18.1%	74.6% 25.4%	80.8% 19.2%

(a)

Handshake	132 17.2%	0 0.0%	0 0.0%	0 0.0%	2 0.3%	1 0.1%	97.8% 2.2%
Clapping	0 0.0%	112 14.6%	0 0.0%	2 0.3%	1 0.1%	0 0.0%	97.4% 2.6%
Hugging	0 0.0%	1 0.1%	64 8.3%	1 0.1%	0 0.0%	0 0.0%	97.0% 3.0%
Walking	0 0.0%	0 0.0%	2 0.3%	100 13.0%	0 0.0%	0 0.0%	98.0% 2.0%
Wave 1	3 0.4%	0 0.0%	0 0.0%	1 0.1%	157 20.4%	4 0.5%	95.2% 4.8%
Wave 2	0 0.0%	1 0.1%	2 0.3%	0 0.0%	2 0.3%	181 23.5%	97.3% 2.7%
	97.8% 2.2%	98.2% 1.8%	94.1% 5.9%	96.2% 3.8%	96.9% 3.1%	97.3% 2.7%	97.0% 3.0%

(b)

Figure 7: (a) The confusion matrix formed for all the frames. (b) The confusion matrix formed for medial frames.



(a)

(b)

Figure 8: (a) The comparison of results using only the medial frames. (b) The comparison using all of the frames.

increased. This shows that the rapidly increasing number of frames does not have much effect on the efficiency of proposed algorithm.

A receiver operating characteristics (ROC) analysis is also performed for videos within the database. Figure 10 gives ROC graph for all the frames in the database while Figure 11 gives the ROC distribution for only the medial frames in the video database. Both graphs suggest that the accuracy of the proposed system is better than the current state of the art. Also the accuracy is greatly increased if only the medial frames in an action video are considered.

6. Conclusions

The paper presents a robust technique for recognition of human actions. It provides a robust framework for feature extraction of frame and training of classifiers. The moments based feature extraction technique proves to be computationally inexpensive while providing higher accuracy than current state-of-the-art approaches. Hence it is more beneficial than other competitive techniques discussed in the paper. Experimental data exhibits that the system has an accuracy of 97% if used for medial frames. Furthermore

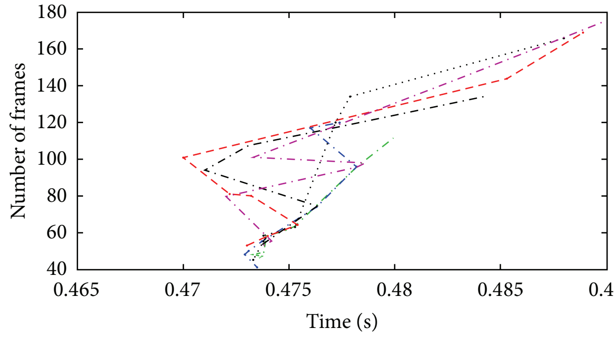


Figure 9: An analysis of number of frames required to identify an action.

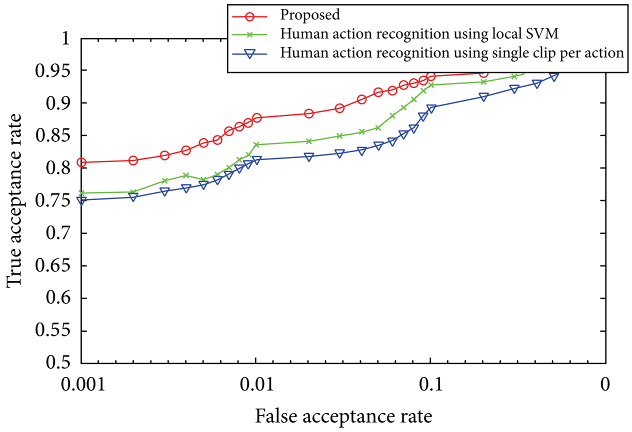


Figure 10: An ROC analysis for the proposed and other competitive techniques using all the frames of video.

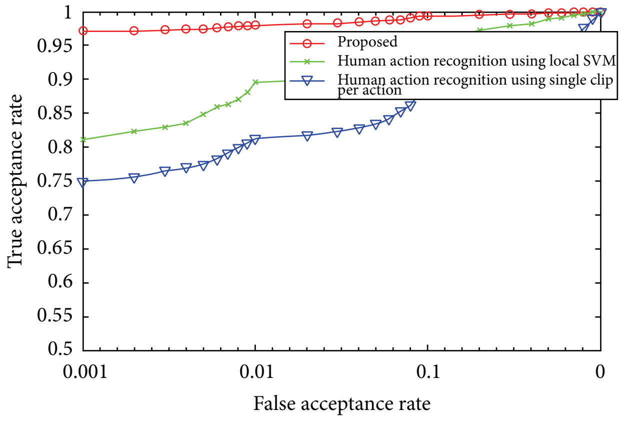


Figure 11: An ROC analysis for the proposed and other competitive techniques using only the medial frames of video.

the experimental results show that the described system is immune to acceptable illumination changes while dealing with indoor and outdoor actions.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] K. N. Tran, I. A. Kakadiaris, and S. K. Shah, "Part-based motion descriptor image for human action recognition," *Pattern Recognition*, vol. 45, no. 7, pp. 2562–2572, 2012.

[2] S. Ali and M. Shah, "Human action recognition in videos using kinematic features and multiple instance learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 2, pp. 288–303, 2010.

[3] O. Busaryev and J. Doolittle, *Gesture Recognition with Applications*, Department of Computer Science and Engineering, The Ohio State University, Columbus, Ohio, USA.

[4] L. Cao, Y. L. Tian, Z. Liu, B. Yao, Z. Zhang, and T. S. Huang, "Action detection using multiple spatial-temporal interest Point features," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '10)*, pp. 340–345, July 2010.

[5] F. Zhang, Y. Wang, and Z. Zhang, "View-invariant action recognition in surveillance videos," in *Proceedings of the 1st Asian Conference on Pattern Recognition (ACPR '11)*, pp. 580–583, Beijing, China, November 2011.

[6] L. Shao, L. Ji, Y. Liu, and J. Zhang, "Human action segmentation and recognition via motion and shape analysis," *Pattern Recognition Letters*, vol. 33, no. 4, pp. 438–445, 2012.

[7] M. M. Ullah, S. N. Parizi, and I. Laptev, "Improving bag-of-features action recognition with non-local cues," in *Proceedings of the British Machine Vision Conference (BMVC '10)*, F. Labrosse, R. Zwiggelaar, Y. H. Liu, and B. Tiddeman, Eds., vol. 10, pp. 95.1–95.11, BMVA Press, September 2010.

[8] D. deMenthon and D. Doermann, "Video retrieval using spatio-temporal descriptors," in *Proceedings of the 11th ACM International Conference on Multimedia (MM '03)*, pp. 508–517, New York, NY, USA, November 2003.

[9] T. Volkmer, *Semantics of Video Shots for Content-Based Retrieval*, 2007.

[10] T. Guha and R. K. Ward, "Learning sparse representations for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 8, pp. 1576–1588, 2012.

[11] Z. Jiang, Z. Lin, and L. Davis, "Recognizing human actions by learning and matching shape-motion prototype trees," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 3, pp. 533–547, 2012.

[12] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2002.

[13] J. Flusser, Z. Barbara, and T. Suk, *Moments and Moment Invariants in Pattern Recognition*, John Wiley & Sons, New York, NY, USA, 2009.

[14] J. Flusser, B. Zitova, and T. Suk, *Moments and Moment Invariants in Pattern Recognition*, John Wiley & Sons, New York, NY, USA, 2009.

[15] H. José Antonio Martín, M. Santos, and J. de Lope, "Orthogonal variant moments features in image analysis," *Information Sciences*, vol. 180, no. 6, pp. 846–860, 2010.

[16] H. Ming-Kuei, "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory*, vol. 8, no. 2, pp. 179–187, 1962.

[17] B. Coppin, *Artificial Intelligence Illuminated*, Jones & Bartlett Learning, Sudbury, Mass, USA, 2004.

[18] C. Schüldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local SVM approach," in *Proceedings of the 17th International Conference on Pattern Recognition (ICPR '04)*, vol. 3, pp. 32–36, IEEE, August 2004.

[19] W. Yang, Y. Wang, and G. Mori, “Human action recognition from a single clip per action,” in *Proceedings of the IEEE 12th International Conference on Computer Vision Workshops (ICCV ’09)*, pp. 482–489, October 2009.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 402185, 13 pages
http://dx.doi.org/10.1155/2014/402185

Research Article

Part-Based Visual Tracking via Online Weighted P-N Learning

Heng Fan,[1] Jinhai Xiang,[2] Jun Xu,[3] and Honghong Liao[4]

[1] *College of Engineering, Huazhong Agricultural University, Wuhan 430070, China*
[2] *College of Science, Huazhong Agricultural University, Wuhan 430070, China*
[3] *Department of Physics, Central China Normal University, Wuhan 430079, China*
[4] *School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China*

Correspondence should be addressed to Jinhai Xiang; jimmy_xiang@163.com

Received 11 April 2014; Accepted 11 June 2014; Published 15 July 2014

Academic Editor: Yu-Bo Yuan

We propose a novel part-based tracking algorithm using online weighted P-N learning. An online weighted P-N learning method is implemented via considering the weight of samples during classification, which improves the performance of classifier. We apply weighted P-N learning to track a part-based target model instead of whole target. In doing so, object is segmented into fragments and parts of them are selected as local feature blocks (LFBs). Then, the weighted P-N learning is employed to train classifier for each local feature block (LFB). Each LFB is tracked through the corresponding classifier, respectively. According to the tracking results of LFBs, object can be then located. During tracking process, to solve the issues of occlusion or pose change, we use a substitute strategy to dynamically update the set of LFB, which makes our tracker robust. Experimental results demonstrate that the proposed method outperforms the state-of-the-art trackers.

1. Introduction

Object tracking is one of the most important components of many applications in computer vision, such as human computer interactions, surveillance, and robotics. However, robust visual tracking is still a challenging problem, which is affected by partial or full occlusion, illumination, scale, and poses variation [1]. The key for the object tracking is to construct an effective appearance model. Many tracking algorithms have been proposed recently, but designing a robust appearance model is still a major challenge, which is affected by both extrinsic (e.g., illumination variation, background clutter, and partial or full occlusion) and intrinsic (e.g., scale and pose variation) factors. In order to handle these problems, a wide range of appearance models based on different visual representations and statistical modeling techniques have been presented by researchers. In general, these appearance models can be categorized into two types: appearance model based on visual representation, such as global-based representation [2–6] and local-based representation [7–11]; appearance model based on statistical modeling, such as generative model [12–16] and discriminative model [5, 6, 17–20].

In this paper, we propose a part-based visual tracking algorithm with online weighted P-N learning. Weighted P-N learning is first proposed by assigning weights (property weight and classification weight) to each sample in training sample set, which can decrease false classification and improve the discriminative power of classifier. Then, we segment object into fragments and select parts of them as local feature blocks to represent the object. Finally, we train classifier for each LFB with weighted P-N learning to obtain the corresponding classifier, respectively, and track each LFB independently within the framework of Lucas-Kanade optical flow [21]. During tracking process, a real-time valid detection method is used for each LFB. If certain LFB is invalid, we use a replacing strategy to update the local feature block set, which can ensure successful tracking.

Contributions. The contributions of this paper include the following.

(i) A part-based visual tracking algorithm with online weighted P-N learning is proposed in this work. Object is represented by LFBs and tracked. When occlusion or distortion happens, a strategy is adopted to replace invalid LFB and keep the new LFB set effective.

(ii) We define the weights (property weight and classification weight) for each sample in training process of P-N learning.

(iii) An online weighted P-N learning is presented by assigning weight to each sample in training sample set, which can improve discriminative power of classifier by decreasing classification errors and increasing the accuracy of tracker.

The rest of the paper is organized as follows. Section 2 reviews the related work of this paper. Section 3 introduces weighted P-N learning. Proposed tracking method is presented in detail in Section 4. Experimental results are shown in Section 5. Section 6 concludes the whole paper.

2. Related Work

Recently, many trackers based on local feature representation have been proposed. Adam et al. [8] present a fragment-based tracking approach, and further, Wang et al. [22] embed the fragment-based method into mean shift tracking framework. This tracking method estimates the target based on voting map of each part via comparing its histogram with the template's. Nevertheless, static template with equal importance being assigned to each fragment obviously lowers the performance of tracker. In order to overcome the shortcomings, Jia et al. [7] propose a fragment-based tracking method using online multiple kernel learning (MKL). All the patches are assigned to different weights based on the importance learned by MKL. However, this strategy may still cause drifting problem. Occlusion especially, which makes part patches invalid, leads to errors in computing voting map, even tracking failure. Wang et al. [10] introduce a tracking method based on superpixel. It only computes the probabilities of superpixels belonging to target, which is prone to drift away in color-similar background and whose tracking results will shrink to the unoccluded part of object when occlusion happens.

Another type of tracking method is based on discriminative appearance model. Tang et al. [23] present a tracking method based on semisupervised support vector machines. This tracker employs a small number of labeled samples for semi-supervised learning and develops a classifier to mark the unlabeled data. Babenko et al. [5] propose a multiple instance learning (MIL) method for visual tracking. This approach solves the problem of slight inaccuracies in the tracker leading to incorrectly labeled training samples and can alleviate drift problem to some extent. However, the MIL tracker might detect positives, which are less important because they do not consider the importance of sample in learning process. Further, Zhang and Song [20] suggest a weighted multiple instance learning (WMIL) tracking method. It assigns weight to each sample based on the corresponding importance. This approach improves the robustness of tracker. Kalal et al. [17] propose a method called P-N learning, which learns from positive samples and negative samples, to construct a classifier. In the meanwhile, the discriminative properties of classifier are improved by two categories of constrains that are termed P-constrains and N-constrains. However, false classification in P-N learning degrades the classifier in some degree.

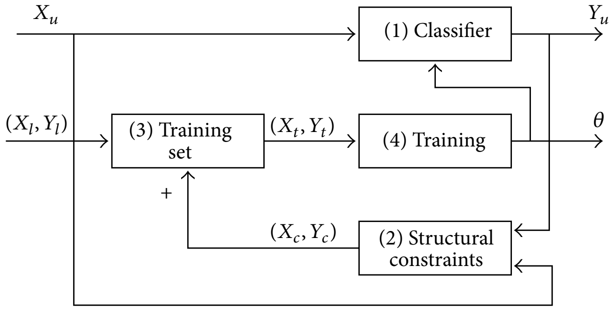


Figure 1: A flowchart in [17] to explain P-N learning. P-N learning algorithm initially develops a classifier from prior knowledge and then iterates over: (1) classify the unlabeled data and label it; (2) reclassify the samples within constrains and label them; (3) expand the training sample set; (4) retrain the classifier.

Another work similar to ours is [9], which utilizes blocks to represent the object. However, the blocks are easily invalided when target appearance changes, which undermines its robustness to nonrigid distortion or occlusion. In our work, we employ LFBs to represent target and use a dynamically updating mechanism to update the local feature block set, which guarantees each LFB in the set is valid when occlusion or deformation occurs. Hence, our tracker is more robust and effective.

3. Weighted P-N Learning

P-N learning is a semisupervised online learning algorithm proposed by Kalal et al. [17, 24–26]. Let x be a sample in feature space $\mathcal{X}$, and let y be a label in label space $\mathcal{Y} = \{1, -1\}$. A set of samples X and corresponding set of labels Y are defined as (X, Y), which is termed a labeled set. The aim of P-N learning is to develop a binary classifier $f : \mathcal{X} \to \mathcal{Y}$ based on a priori labeled set (X_l, Y_l) and improve its discriminative performance by unlabeled data X_u. The flowchart of P-N learning approach in [17] is shown in Figure 1.

3.1. Classifier Bootstrapping. The binary classifier f is a function parameterized by θ. Similar to supervised learning, P-N learning is to estimate parameter θ via training sample set (X_t, Y_t). Nevertheless, it is worth noticing that the training set is iteratively expanded through adding samples, which is screened by constraints from unlabeled data. Initially, classifier and its parameter θ^0 are obtained by training labeled samples. Then, the process proceeds iteratively. In iteration k, all the unlabeled samples are marked by classifier in iteration $k-1$; namely,

$$y_u^k = f\left(x_u \mid \theta^{k-1}\right), \quad x^u \in X_u. \tag{1}$$

Then, the constraints are utilized to revise the classification results and add the corrected labels to training set. Iteration k ends with retraining classifier by using the renewed training sample set.

During training process, the classifier may identify the unlabeled data with wrong labels. Any sample may be screened many times with constraints and hence can be represented mistakenly in the training set repeatedly. Obviously, it can significantly degrade discriminative performance of classifier and therefore lower the accuracy of tracking. In order to further improve the accuracy and robustness of the classifier, we propose a weighted P-N learning method by assigning weight to each sample in training set. Sample j in training set has two categories of weight which are termed P-weight W_j^+ and N-negative W_j^-. P-weight represents the probability of being a positive sample, and N-weight represents the probability of being a negative sample. In iteration k, sample j from training set is represented as positive sample for C_j^+ times and as negative sample for C_j^- times. The positive weight $W1_j^+$ and negative weight $W1_j^-$ are determined by the following formulation:

$$\begin{aligned} W1_j^+ &= \frac{C_j^+}{C_j^+ + C_j^-}, \\ W1_j^- &= \frac{C_j^-}{C_j^+ + C_j^-}. \end{aligned} \tag{2}$$

Besides, the probability of sample j being positive or negative in training set obtained by classifier is defined as classification weights $W2_j^+$ and $W2_j^-$ (in Section 3.3). The P-weight and N-weight of sample j can be then obtained by the following formulation:

$$\begin{aligned} W_j^+ &= W1_j^+ + W2_j^+, \\ W_j^- &= W1_j^- + W2_j^-. \end{aligned} \tag{3}$$

At last, sample j is determined to be either positive or negative via the following formulation:

$$\begin{aligned} &\frac{W_j^+}{W_j^-} \geq 1 \quad \text{positive sample,} \\ &\frac{W_j^+}{W_j^-} < 1 \quad \text{negative sample.} \end{aligned} \tag{4}$$

Figure 2 demonstrates the tracking results with weighted P-N learning. In Figure 2, the left and middle images are the ground truth and tracking results with weighted P-N learning, and the right images are tracking results based on P-N learning.

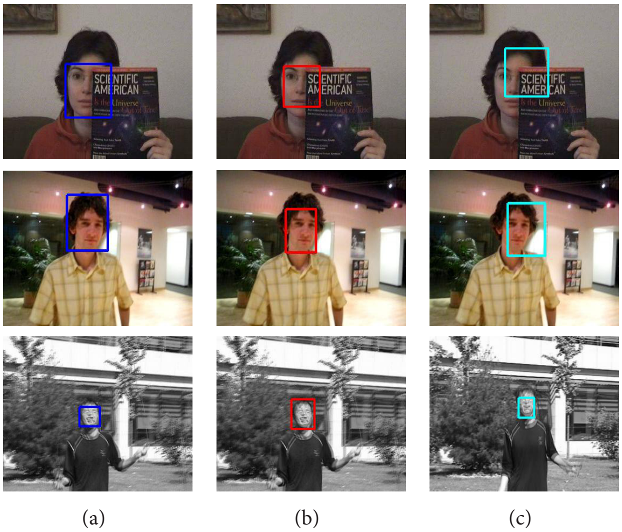


Figure 2: (a) Ground truth. (b) Tracking results with weighted P-N learning. (c) Tracking results based on P-N learning.

3.2. Constraints. In P-N learning, a constraint can be arbitrary function, especially two categories of constraints which we term P and N. P-constrains recognize samples which are labeled negative by the classifier, yet constraints need a positive label. P-constraints add $n^+(k)$ samples to the training set in iteration k. Similarly, N-constraints are employed to identify samples classified as positive but constraints require negative label. In iteration k, N-constraints insert $n^-(k)$ samples to training set.

In iteration k, the error of a classifier is represented by a number of false positives $\alpha(k)$ and a number of false negatives $\beta(k)$. Let $n_c^+(k)$ be the number of samples for which the label is correctly changed to positive in iteration k by P-constraints, and $n_f^+(k)$ is then the number of samples for which the label is incorrectly changed to positive in iteration k. Hence, P-constraints change $n^+(k) = n_c^+(k) + n_f^+(k)$ samples to positive. Similarly, N-constraints change $n^-(k) = n_c^-(k) + n_f^-(k)$ samples to negative, where $n_c^-(k)$ and $n_f^-(k)$ are correct and false assignments. The errors of classifier can be represented as the following formulations:

$$\alpha(k+1) = \alpha(k) - n_c^-(k) + n_f^+(k), \tag{5}$$

$$\beta(k+1) = \beta(k) - n_c^+(k) + n_f^-(k). \tag{6}$$

Equation (5) demonstrates that false positives $\alpha(k)$ decrease if $n_c^-(k) > n_f^+(k)$. In the similar way, false negatives $\beta(k)$ decrease if $n_c^+(k) > n_f^-(k)$. To analyze the convergence of learning process, a model needs to be developed that relates the performance of P-N constraints to $n_c^+(k)$, $n_c^-(k)$, $n_f^+(k)$, and $n_f^-(k)$.

The performance of P-N constraints is represented by four indexes, P-precision P^+, P-recall R^+, N-precision P^-, and

N-recall R^-, determined by the following formulation:

$$\begin{aligned} P(k)^+ &= \frac{n_c^+(k)}{n_c^+(k) + n_f^+(k)}, \\ R(k)^+ &= \frac{n_c^+(k)}{\beta(k)}, \\ P(k)^- &= \frac{n_c^-(k)}{n_c^-(k) + n_f^-(k)}, \\ R(k)^- &= \frac{n_c^-(k)}{\alpha(k)}. \end{aligned} \tag{7}$$

According to formulation (7), it is easy to get

$$\begin{aligned} n_c^+(k) &= R(k)^+ \cdot \beta(k), \\ n_f^+(k) &= \frac{1 - P(k)^+}{P(k)^+} \cdot R(k)^+ \cdot \beta(k), \\ n_c^-(k) &= R(k)^- \cdot \alpha(k), \\ n_f^-(k) &= \frac{1 - P(k)^-}{P(k)^-} \cdot R(k)^- \cdot \alpha(k). \end{aligned} \tag{8}$$

By combining formulations (5), (6), and (8), we can obtain new formulations:

$$\begin{aligned} \alpha(k+1) &= \left(1 - R(k)^-\right) \cdot \alpha(k) + \frac{1 - P(k)^+}{P(k)^+} \cdot R(k)^+ \cdot \beta(k), \\ \beta(k+1) &= \frac{1 - P(k)^-}{P(k)^-} \cdot R(k)^- \cdot \alpha(k) + \left(1 - R(k)^+\right) \cdot \beta(k). \end{aligned} \tag{9}$$

After defining state vector $\vec{x}(k) = [\alpha(k)\ \ \beta(k)]^T$ and transition matrix $\mathbf{M}$ as the following,

$$\mathbf{M} = \begin{bmatrix} 1 - R(k)^- & \dfrac{1 - P(k)^+}{P(k)^+} \cdot R(k)^+ \\ \dfrac{1 - P(k)^-}{P(k)^-} \cdot R(k)^- & 1 - R(k)^+ \end{bmatrix}, \tag{10}$$

hence formulation (9) can be rewritten as the following formulation:

$$\vec{x}(k+1) = \mathbf{M}\vec{x}(k). \tag{11}$$

According to [27], formulation (11) is a recursive equation that is related to a discrete dynamical system. Based on the theory of dynamical systems, the state vector $\vec{x}$ converges to zero if eigenvalues λ_1 and λ_2 of the transition matrix $\mathbf{M}$ meet the condition $\lambda_1 < 1$ and $\lambda_2 < 1$. As pointed in [17], the performance of classifier will be improved constantly, only if the two eigenvalues of transition matrix $\mathbf{M}$ are smaller than one.

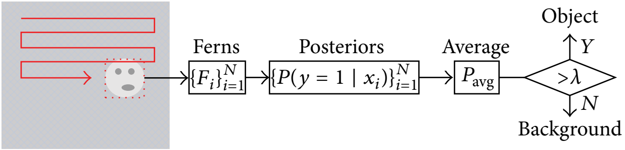


Figure 3: Object detection based on scanning window strategy and randomized forest classifier. The setting of the detector is as follows: 10,000 windows are scanned, 10 ferns per window.

3.3. Object Detecting. In previous subsections, we illustrate the weighted P-N learning method. In this subsection, a classifier will be developed to detect the object. Scanning window strategy is utilized to detect the object in [17]. Similarly, we use this method to detect the object.

In this paper, the randomized forest classifier [28] is adopted. For each input subwindow, classifier consists of N ferns. Each fern i computes the input patch resulting in feature vector x_i, which is used to obtain posterior probability $P(y = 1 \mid x_i)$. The following formulation is defined to discriminate input patch:

$$\begin{aligned} P_{\text{avg}} &> \lambda \quad \text{object}, \\ P_{\text{avg}} &\le \lambda \quad \text{background}, \end{aligned} \tag{12}$$

where $P_{\text{avg}} = \sum_{i=1}^{N} P(y = 1 \mid x_i)$ denotes the average of all posteriors and λ is the threshold which is set to 0.4 in all experiments. The detection process can be illustrated in Figure 3. Actually, $W2_j^+ = P_{\text{avg}}$ and $W2_j^- = 1 - P_{\text{avg}}$. Feature vector is represented by 2-bit Binary Patterns [25] because of their invariance to illumination and efficient multiscale implementation using integral image. In fact, the posteriors $P(y = 1 \mid x_i)$ represent the parameter θ of the classifier and are estimated incrementally through the entire learning process. Each leafnode of fern records the number of positive p and negative n samples changed into it during iteration. The posteriors are then estimated by the following formulation:

$$\begin{aligned} P(y = 1 \mid x_i) &= \frac{p}{p+n} \quad \text{leaf is not empty}, \\ P(y = 1 \mid x_i) &= 0 \quad \text{leaf is empty}. \end{aligned} \tag{13}$$

The classifier is initialized in first frame, and posteriors are initialized to zero and renewed by 500 positive samples produced by affine warping of the selected patch [1]. The classifier is then evaluated on all the patches. In this paper, detections far from the selected patch represent the negative samples and update the posteriors.

4. Tracking

In this paper, the object is represented by independent local feature blocks. The tracking task is then transformed into tracking each local feature block. We train classifier for each LFB with online weighted P-N learning, respectively, and then track each LFB independently within the framework of LK optical flow [21]. During tracking procedure, a real-time

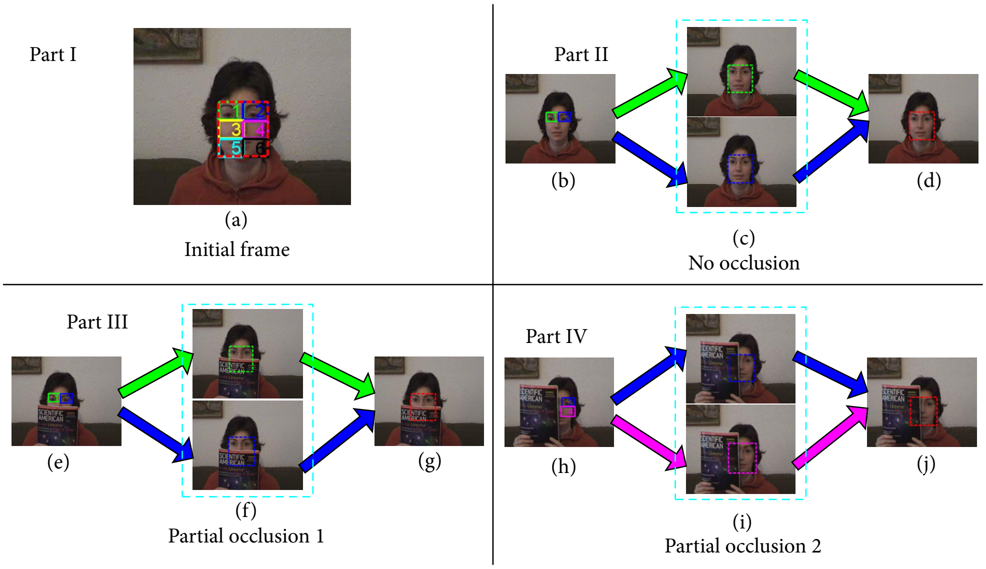


Figure 4: Illustration of tracking process. Part (I): segmentation of object in initial frame (first frame). Object is divided into six blocks. Part (II): tracking object without occlusion. Green block and blue block are selected as LFBs in image (b). Green and blue dotted line rectangles in image (c) are the tracking results of LFBs, and the red dotted line rectangle in image (d) represents the location of object estimated by the tracking results of LFBs. Part (III): tracking object with occlusion yet valid LFBs. Essentially, this process is similar to part (II). Occlusion does not affect the tracking. Part (IV): tracking object with occlusion with invalid LFB. In this part, green LFB is occluded and therefore invalid. We replace it with another new valid LFB, namely, the pink LFB, as shown in image (h). The object is then located in image (j) by tracking results of blue LFB and pink LFB in image (i).

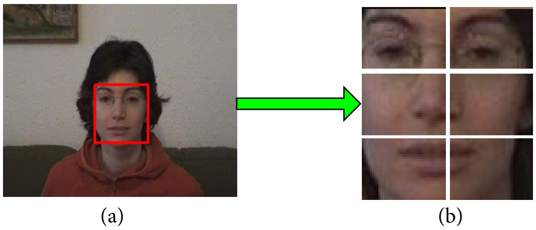


Figure 5: Uniform segmentation of object. Image (a) is object and image (b) is fragments.

valid detection algorithm is utilized for each LFB. If certain LFB is invalid, it will be replaced with an unused block, which makes our tracker robust. Figure 4 illustrates the principle of tracking.

4.1. Set of Local Feature Blocks. Object is represented by LFBs, and thereby, object needs to be segmented into fragments. For simplicity, uniform segmentation is adopted in this paper as shown in Figure 5.

After segmentation, we select part blocks as LFBs. Assume object is divided into K blocks; then we can obtain a candidate set of LFB set $CB = \{b_1, b_2, \ldots, b_K\}$ with K candidate local feature blocks. For candidate LFB b_i, we compute its 2-bit Binary Patterns feature vector x_i. Then, scanning window method is used to compute the similar likelihood between feature vectors of input patch f_i and b_i, and the similarity is represented as L_{ij}. $ML_i = \max(L_{ij})$ represents the highest similarity between b_i and all input patches. Finally, local feature block set $SB = \{sb_1, sb_2, \ldots, sb_M\}$ consists of M ($M < K$) candidate local feature blocks, with smaller ML_i.

4.2. Representations. In frame t, $B^t = (C^t, R^t, W^t, H^t)$ represents the object, where C^t and R^t are coordinates of center position and W^t and H^t are the sizes of object. $sb_k^t = (c_k^t, r_k^t, w_k^t, h_k^t)$ is the kth LFB, where c_k^t and r_k^t are coordinates of center location and w_k^t and h_k^t are the sizes of local feature block. $z_k^t = (oc_k^t, or_k^t, rw_k^t, rh_k^t)$ represents the offset of the kth LFB relative to object, where oc_k^t and or_k^t are the offset of center coordinates and rw_k^t and rh_k^t are the rations of sizes between object and the kth LFB. z_k^t can be determined by the following formulation:

$$\begin{aligned} oc_k^t &= C^t - c_k^t, \\ or_k^t &= R^t - r_k^t, \\ rw_k^t &= \frac{W^t}{w_k^t}, \\ rh_k^t &= \frac{H^t}{h_k^t}. \end{aligned} \tag{14}$$

4.3. Object Tracking. The tracked target is determined in initial frame (first frame) and segmented into fragments according to Section 4.1. Then, we select local feature blocks and compute B^1, sb_k^1, and z_k^1.

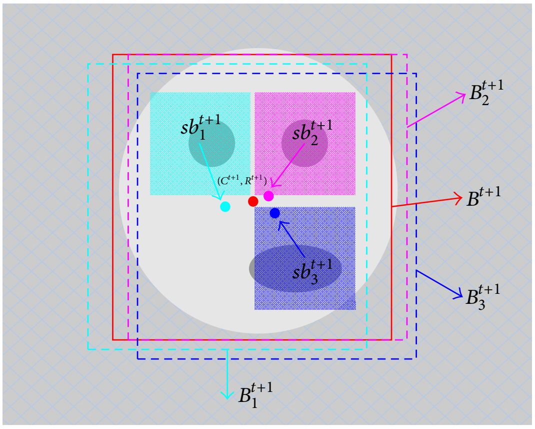


Figure 6: The process of locating object. The green, pink, and blue patches are the local feature blocks sb_1^{t+1}, sb_2^{t+1}, and sb_3^{t+1}, and the green, blue, and pink dotted line rectangles are the corresponding tracking results B_1^{t+1}, B_2^{t+1}, and B_3^{t+1}. The red solid line rectangle represents the final tracking result B^{t+1}, which is located by tracking results of local feature blocks.

Assume current frame is t. Each LFB is corresponded with a classifier via weighted P-N learning, and then we track each LFB. For the kth LFB, $sb_k^{t+1} = (c_k^{t+1}, r_k^{t+1}, w_k^{t+1}, h_k^{t+1})$ is used to represent its tracking result in frame $t+1$. By combining tracking results of each LFB and its offset z_k^t, we can obtain the corresponding object $B_k^{t+1} = (C_k^{t+1}, R_k^{t+1}, W_k^{t+1}, H_k^{t+1})$ via formulation the following formulation:

$$\begin{aligned} C_k^{t+1} &= oc_k^t + c_k^{t+1}, \\ R_k^{t+1} &= or_k^t + r_k^{t+1}, \\ W_k^{t+1} &= rw_k^t \cdot w_k^{t+1}, \\ H_k^{t+1} &= rh_k^t \cdot h_k^{t+1}. \end{aligned} \tag{15}$$

Each LFB determines a related candidate object. Object finally can be located by the following formulation:

$$B^{t+1} = \frac{1}{M} \cdot \sum_{i=1}^{M} B_i^{t+1}, \tag{16}$$

where M is the number of LFBs. The entire process can be explained by Figure 6. An adjustment is needed for offset of each LFB relative to object. We divide new object region B^{t+1} into fragments and compute the new offset of each LFB with prior LFBs based on formulation (14). After this, classifier of each LFB needs to be retrained via weighted P-N learning.

Representation by LFBs has significant advantages. Firstly, compared with entire object, local feature block is more prone to recognition in background, and this guarantees the accuracy and stability of tracking. Besides, object is located by averaging all the tracking results of local feature blocks, which decreases tracking errors by counteracting positive and negative errors; therefore, the robustness of proposed algorithm is improved.

4.4. Updating Set of Local Feature Blocks. During the tracking process, update is essentially adaptive to complex environment variation. In this paper, the learning procedure is online, and then the main problem is how to handle the situation of local feature blocks being invalid. A strategy of replacing is adopted to solve this problem. During the tracking procedure, we make a real-time valid detection (in Section 3.3) for all local feature blocks. If $P_{\text{avg}} \leq \lambda$, local feature block is invalid. When certain local feature block is invalid, it will be replaced with an appropriate block, which is selected from the outside of the LFB set.

Let UB be the unused block set, and $UB = \{ub_1, ub_2, \ldots, ub_P\}$, $P = K - M$, $UB \bigcup SB = CB$. In current frame t, LFB sb_k^t from SB is invalid and needs to be replaced. We first segment object into blocks and obtain UB. For block j from UB, we compute similar likelihood $l_j^t = \text{sim}(x_j^t, x_j^q)$, where sim is function of computing similar likelihood, x_j^t is feature vector of block j in frame t, and x_j^q is feature vector of block j before it is used the last time in frame q (if block j is never used, q equals one). ub_τ is used to replace sb_k^t via the following formulation:

$$ub_\tau = \arg\max l_j^t \left(ub_j\right). \tag{17}$$

The whole update process can be illustrated as shown in Figure 7.

So far, we have introduced the overall procedure of the proposed tracking algorithm as shown in Algorithm 1.

5. Experimental Results

In order to evaluate the performance of our tracking algorithm, we test our tracker on thirteen challenging image sequences. These sequences cover most challenging situations in visual tracking as shown in Table 1. For comparison, we run six state-of-the-art tracking algorithms with the same initial position of object. These algorithms are ℓ_1 tracking [2], FG tracking [8], IVT tracking [3], MIL tracking [5], TLD tracking [26], and CT tracking [6] approaches. Some representative results are shown in this section.

5.1. Quantitative Comparison. We evaluate the above-mentioned trackers via overlapping rate [29] as well as center location error, and the comparing results are shown in Tables 2 and 3.

Figure 8 shows the center location error of utilized tracker on thirteen test sequences. Overall, the tracker proposed in this paper outperforms the state-of-the-art algorithms.

5.2. Qualitative Comparison

Heavy Occlusion. Occlusion is one of the most common yet crucial issues in visual tracking. We test four image sequences (Woman, Subway, PersonFloor, and Occlusion1) characterized in severe occlusion or long-time partial occlusion. Figure 9(a) demonstrates the robustness performance of proposed tracking method in handling occlusion. Object is represented by local feature blocks in proposed algorithm.

```
Initialization:
Segment object into $N$ blocks and obtain $CB = \{cb_i\}_{i=1}^{N}$;
Select the set of LFBs $SB = \{sb_k\}_{k=1}^{M} (M < N)$;
Compute the set of offset in first frame $\{z_k^1\}_{k=1}^{M}$;
Generate positive samples $(\mathcal{X}_k^+, \mathcal{Y}_k^+)$ and negative
  samples $(\mathcal{X}_k^-, \mathcal{Y}_k^-)$ for the $k^{th}$ LFB;
Train classifier $f_k$ for the $k^{\text{th}}$ LFB with data $(X, Y)$ via
  weighted P-N learning, where $X = \{\mathcal{X}_k^+, \mathcal{X}_k^-\}$,
  and $Y = \{\mathcal{Y}_k^+, \mathcal{Y}_k^-\}$;
Object Tracking:
for $t = 2$ to the end of the sequence do
  for $k = 1$ to $M$ do
    Estimate $sb_k^t$ via detecting $k^{\text{th}}$ LFB with classifier
      in LK framework;
    Compute $B_k^t$ via (15);
    Retrain classifier $f_i$;
  end for
  Estimate $B^t$ via (16);
  Adjust the set of offset via (14);
  for $k = 1$ to $M$ do
    Check each LFB with corresponding classifier
    if $P_{\text{avg}}^k < \lambda$ do
      Update $sb_k \leftarrow ub_\tau$ based on Section 4.4;
      Generate positive samples $(\mathcal{X}_k^+, \mathcal{Y}_k^+)$ and
        negative samples $(\mathcal{X}_k^-, \mathcal{Y}_k^-)$ for the $k^{\text{th}}$ LFB;
      Train new classifier $f_k$ for the $k^{\text{th}}$ LFB
      break;
    end if
end for
End
```

ALGORITHM 1: Tracking based on proposed method.

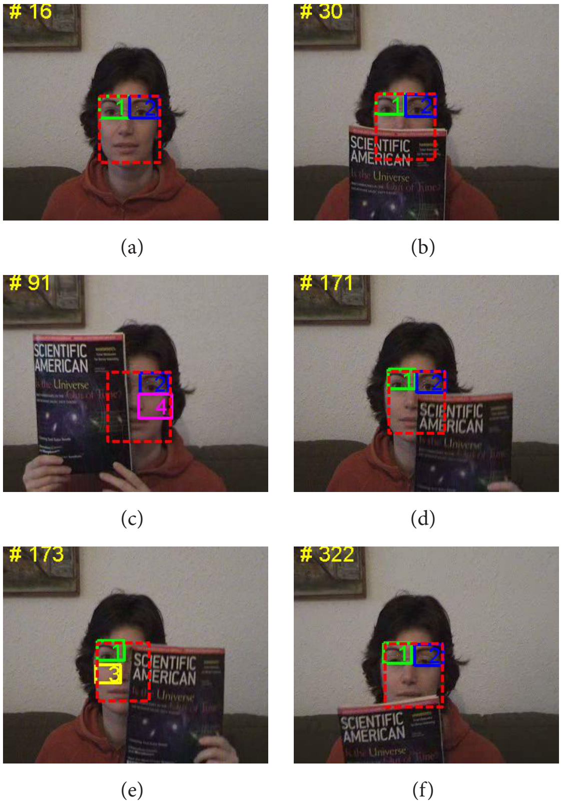


FIGURE 7: The process of updating. The object is located by LFBs in (a) without occlusion. When occlusion occurs, yet each LFB is valid, the object can be tracked by LFBs exactly, as shown in (b). If occlusion happens and certain LFB is invalid, then it will be replaced and target still can be successfully located, as shown in (c), (d), (e), and (f).

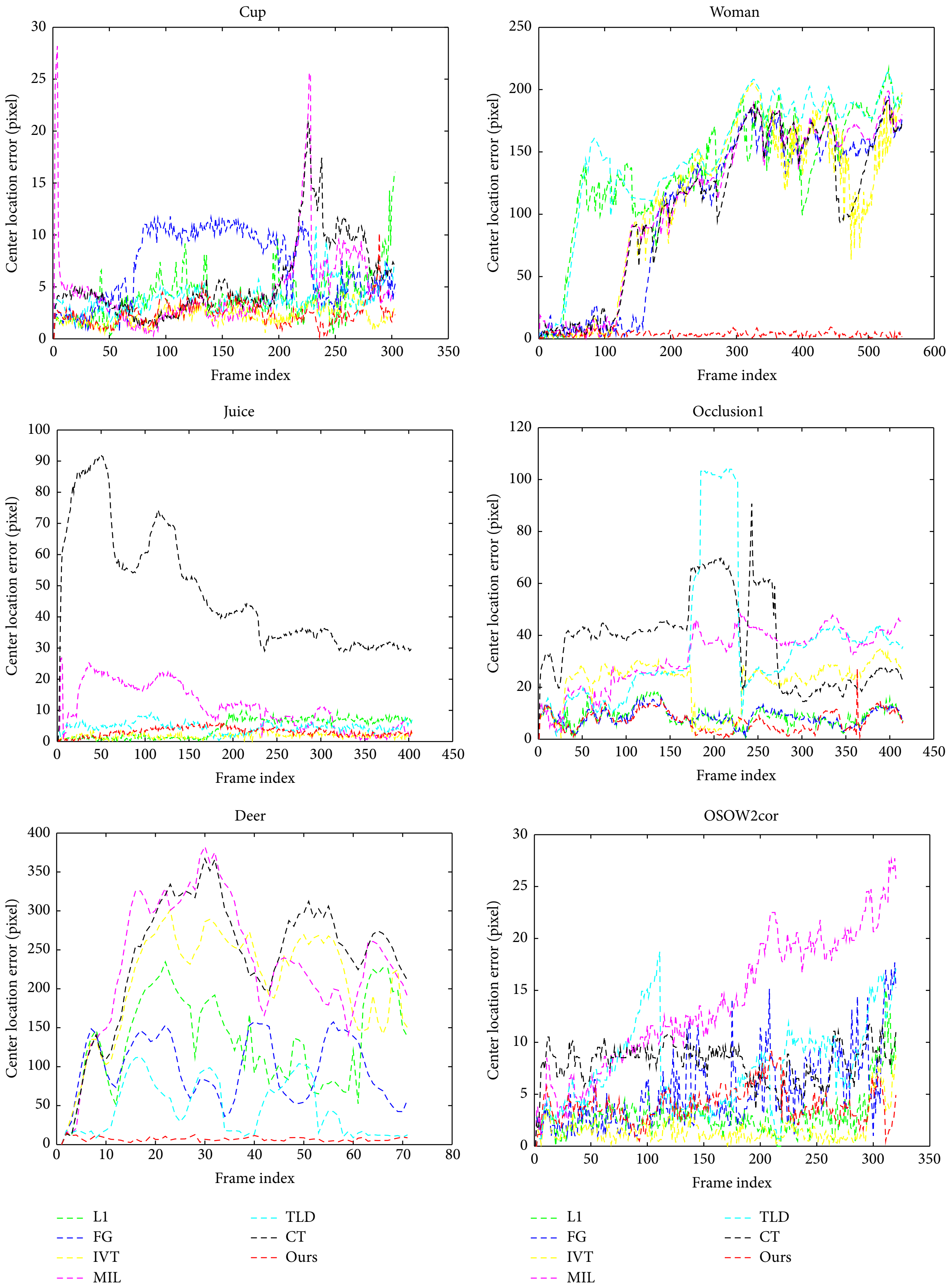
TABLE 1: The tracking sequences used in our experiments.

Sequence	Frames	Main challenges
Cup	303	Direction variation and scale variation
DavidIndoor	770	Illumination, scale variation, and pose change
DavidOutdoor	569	Illumination and scale variation
Deer	71	Fast motion, motion blur, and background clutter
Juice	404	Direction variation and scale variation
Jumping	313	Fast motion and motion blur
Lemming	900	Fast motion, motion blur, and occlusion
Occlusion1	415	Occlusion
OneLSR	559	Scale variation and occlusion
OSOW2cor	320	Scale variation and pose change
PersonFloor	387	Scale variation and occlusion
Subway	175	Occlusion and background clutter
Woman	551	Occlusion, scale variation, and pose change

TABLE 2: Center location errors (in pixels). The best result is shown in bold and the second best in italic fonts.

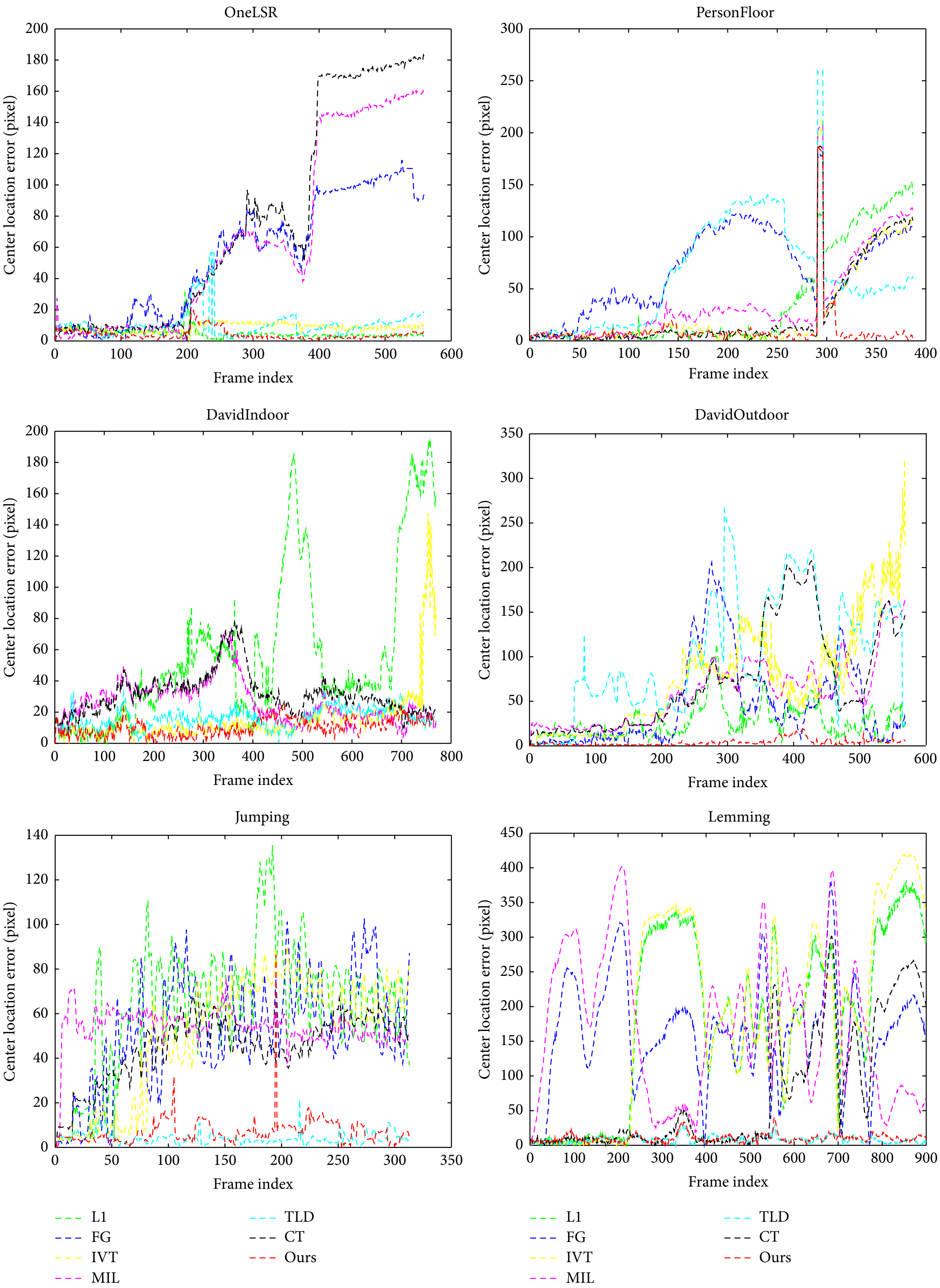
	ℓ_1	FG	IVT	MIL	TLD	CT	Ours
Woman	136.69	*106.21*	110.46	114.02	148.66	107.74	**3.77**
DavidIndoor	57.52	—	*15.78*	26.43	16.19	31.83	**10.34**
Cup	4.09	7.02	**2.23**	4.78	3.86	5.62	*2.47*
Juice	4.39	—	**1.80**	11.07	4.57	47.97	*3.07*
Deer	132.59	97.28	206.68	231.29	*45.83*	240.36	**7.02**
Lemming	181.13	164.87	192.12	168.87	**8.19**	73.28	*10.14*
OneLSR	*4.87*	54.70	8.43	65.48	10.50	76.47	**3.95**
PersonFloor	37.63	68.15	27.06	37.13	62.94	*26.89*	**9.52**
Subway	153.60	8.06	124.60	137.86	*5.26*	11.50	**3.19**
OSOW2cor	*3.06*	5.22	**1.62**	13.83	7.50	8.08	3.97
DavidOutdoor	*26.50*	45.87	75.86	59.77	102.20	73.72	**3.71**
Occlusion1	8.85	*8.60*	21.81	31.91	35.43	37.96	**7.11**
Jumping	64.23	52.34	50.18	54.03	**4.06**	45.11	*6.87*
Average	62.70	56.21	64.51	73.57	*35.01*	60.50	**5.78**

When occlusion happens, other tracking algorithms cannot track object well because they are prone to update background into object. However, our tracker can employ a new unused block to replace the invalid local feature block when occlusion occurs, which can make local feature block set effective to continue tracking.



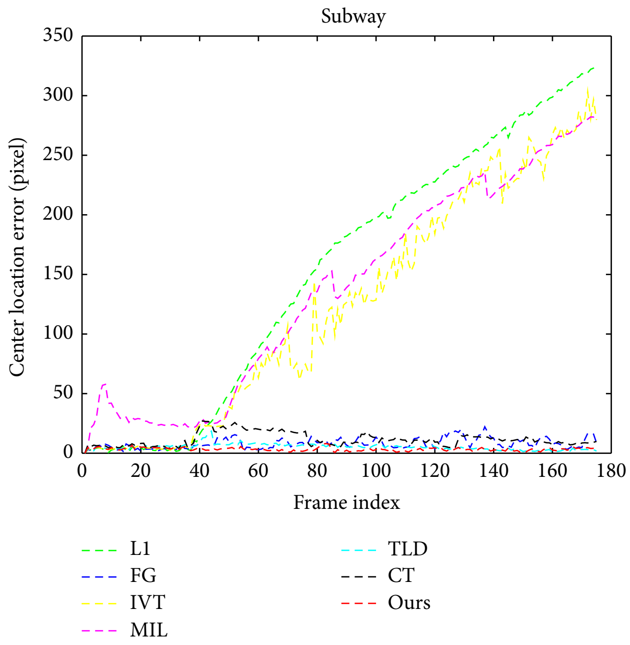
(a)

Figure 8: Continued.



(b)

Figure 8: Continued.



(c)

Figure 8: Quantitative evaluation of the trackers in terms of position errors (in pixels).

Table 3: Overlapping rate. Bold fonts indicate the best performance while the italic fonts indicate the second best ones.

	ℓ_1	FG	IVT	MIL	TLD	CT	Ours
Woman	0.06	*0.19*	0.14	0.15	0.04	0.15	**0.72**
DavidIndoor	0.21	—	0.3	0.45	**0.52**	0.39	*0.47*
Cup	0.64	0.67	0.61	**0.77**	0.63	*0.73*	0.62
Juice	*0.73*	—	**0.79**	0.64	0.66	0.07	0.69
Deer	0.04	0.07	0.03	0.04	*0.34*	0.04	**0.65**
Lemming	0.19	0.05	0.19	0.07	*0.64*	0.43	**0.7**
OneLSR	*0.54*	0.24	0.3	0.26	0.45	0.26	**0.56**
PersonFloor	0.47	0.25	0.53	0.47	0.21	*0.6*	**0.64**
Subway	0.15	0.61	0.17	0.09	**0.68**	0.5	*0.67*
OSOW2cor	0.59	0.43	**0.63**	0.39	0.5	0.4	*0.62*
DavidOutdoor	0.35	*0.42*	0.17	0.3	0.08	0.31	**0.62**
Occlusion1	**0.71**	**0.72**	0.5	0.55	0.48	0.41	0.66
Jumping	0.07	0.1	0.17	0.01	**0.78**	0.07	*0.75*
Average	0.37	0.36	0.35	0.32	*0.46*	0.33	**0.64**

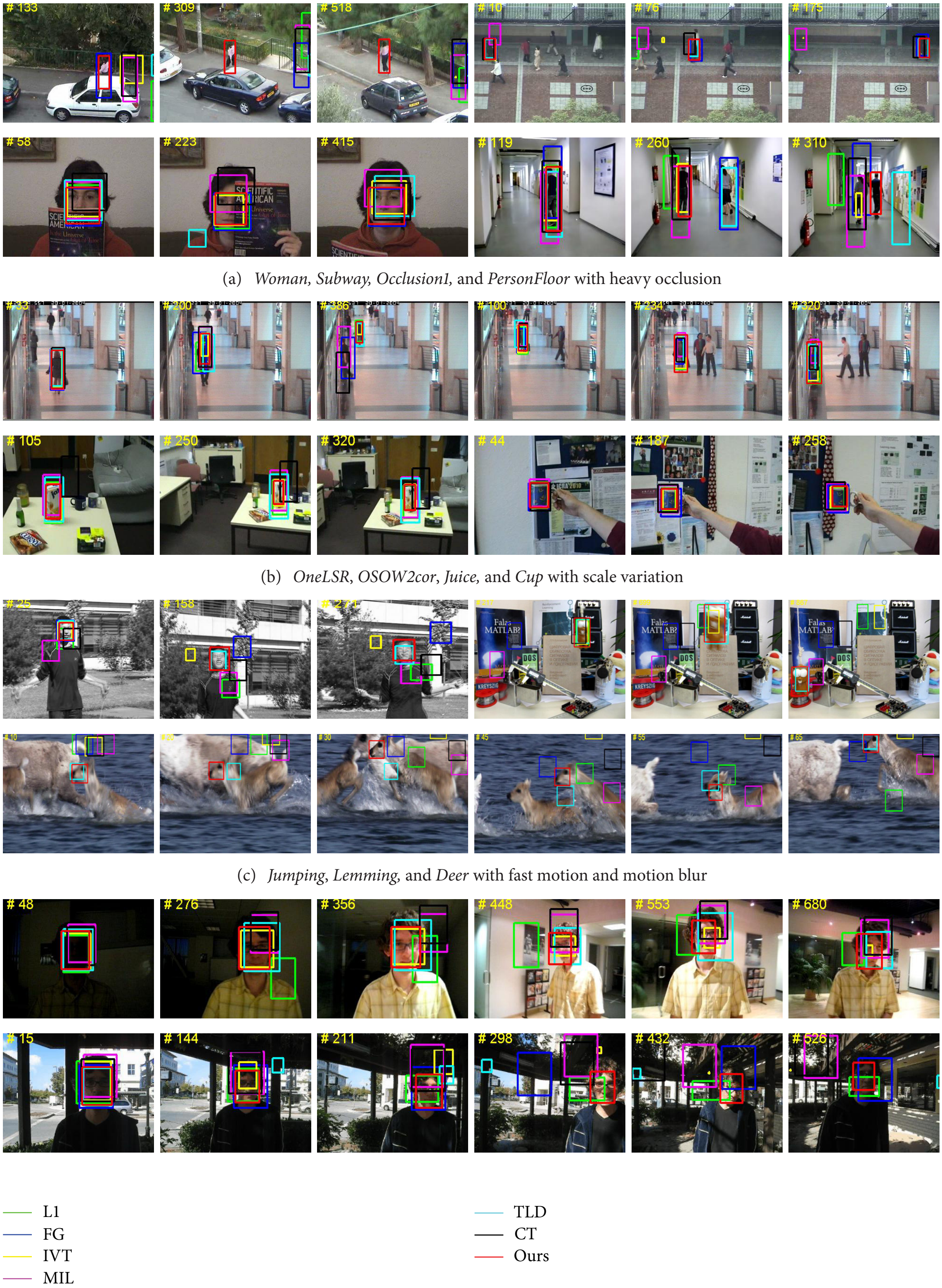
Scale Variation. Figure 9(b) presents the tracking results on four image sequences (OneLSR, OSOW2cor, Juice, and Cup) with large scale variation, even more with slight rotation. Our tracker can tail object throughout the whole sequences, which can be attributed to the discriminative classifier based on weighted P-N learning. We also observe that local feature blocks can better represent object, which makes the tracker focus on the stable part of the object.

Fast Motion and Motion Blur. Figure 9(c) demonstrates experimental results on three challenging sequences (Deer, Lemming, and Jumping). Because the target undergoes fast and abrupt motion, it is more prone to cause blur, which causes drifting problem. It is worth noticing that the suggested approach in this paper performs better than other algorithms. When motion blur occurs, our tracker can guarantee that the object's local feautres are still available. The advantages of using local feature blocks to represent object are shown incisively and vividly. By combining improved P-N learning and local feature blocks, we can obtain a discriminative classifier of stable object parts, which can locate the object. Then we track each local feature block, respectively, and determine the object based on tracking results of local feature blocks.

Illumination Variation. Illumination is a critical factor in visual tracking. Two typical image sequences (DavidIndoor and DavidOutdoor) are employed to test our tracker as shown in Figure 9(d). When illumination varies, some local regions of target are insensitive actually. Our tracker captures these insensitive regions to track local areas of object and further locate entire object via local tracking information.

6. Conclusions

In this paper, we propose a part-based visual tracking algorithm with online weighted P-N learning. An online P-N learning is presented by assigning weight (property weight and classification weight) to each sample in training sample set, which can decrease classification errors and can improve the discriminative power of classifier. Firstly, the target is segmented into fragments, and parts of them are chosen to be local feature blocks to represent object. We



(a) *Woman, Subway, Occlusion1,* and *PersonFloor* with heavy occlusion

(b) *OneLSR, OSOW2cor, Juice,* and *Cup* with scale variation

(c) *Jumping, Lemming,* and *Deer* with fast motion and motion blur

(d) *DavidIndoor* and *DavidOutdoor* with illumination changes

Figure 9: Tracking results on various challenging sequences.

then train classifier for each LFB with weighted P-N learning, obtain the corresponding classifier, respectively, and track each LFB independently within the framework of LK optical flow. In addition, a substitute strategy is adopted to update dynamically the set of LFBs, which ensures robust tracking. Experimental results demonstrate that our algorithm outperforms state-of-the-art trackers. However, our algorithm fails to track object exactly in some scenes. If the tracked target is nonrigid and has an extremely heavy deformation or is fully occluded for long time, the performance of proposed tracker drops.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

Jinhai Xiang is supported by the Fundamental Research Funds for the Central Universities under Grant no. 2013QC024. Jun Xu is supported by the National Natural Science Foundation of China under Grant no. 11204099 and self-determined research funds of CCNU from the colleges' basic research and operation of MOE.

References

[1] Y. Wu, J. Lim, and M. H. Yang, "Online object tracking: a benchmark," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 2411–2418, 2013.

[2] X. Mei and H. Ling, "Robust visual tracking using ℓ 1 minimization," in *Proceedings of the 12th IEEE Conference on Computer Vision and Pattern Recognition(CVPR '09)*, pp. 1436–1443, 2009.

[3] D. A. Ross, J. Lim, R. S. Lin, and M. H. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1–3, pp. 125–141, 2008.

[4] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–577, 2003.

[5] B. Babenko, S. Belongie, and M. Yang, "Visual tracking with online multiple instance learning," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR '09)*, pp. 983–990, Miami, Fla, USA, June 2009.

[6] K. Zhang, L. Zhang, and M. H. Yang, "Real-time compressive tracking," in *Proceedings of the European Conference on Computer Vision (ECCV '12)*, pp. 864–877, Firenze, Italy, October 2012.

[7] X. Jia, D. Wang, and H. Lu, "Fragment-based tracking using online multiple kernel learning," in *Proceedings of the 19th IEEE International Conference on Image Processing (ICIP '12)*, pp. 393–396, October 2012.

[8] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, pp. 798–805, June 2006.

[9] S. M. Shahed Nejhum, J. Ho, and M. H. Yang, "Visual tracking with histograms and articulating blocks," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, June 2008.

[10] S. Wang, H. Lu, F. Yang, and M. Yang, "Superpixel tracking," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 1323–1330, Barcelona, Spain, November 2011.

[11] R. Yao, Q. Shi, C. Shen, Y. Zhang, and A. van den Hengel, "Part-based visual tracking with online latent structural learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 2363–2370, Portland, Ore, USA, 2013.

[12] Q. Yu, B. T. Dinh, and G. Medioni, "Online tracking and reacquisition using co-trained generative and discriminative trackers," in *Proceedings of European Conference on Computer Vision (ECCV '08)*, pp. 678–691, 2008.

[13] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via multi-task sparse learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 2042–2049, June 2012.

[14] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 1269–1276, San Francisco, Calif, USA, June 2010.

[15] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust L1 tracker using accelerated proximal gradient approach," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 1830–1837, Providence, RI, USA, June 2012.

[16] B. Liu, J. Huang, L. Yang, and C. Kulikowsk, "Robust tracking using local sparse appearance model and K-selection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 1313–1320, June 2011.

[17] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-N learning: bootstrapping binary classifiers by structural constraints," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 49–56, June 2010.

[18] H. Grabner, C. Leistner, and H. Bischof, "Semisupervised online boosting for robust tracking," in *Proceedings of European Conference on Computer Vision (ECCV '08)*, pp. 234–247, 2008.

[19] H. Grabner, J. Matas, L. Van Gool, and P. Cattin, "Tracking the invisible: Learning where the object might be," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 1285–1292, June 2010.

[20] K. Zhang and H. Song, "Real-time visual tracking via online weighted multiple instance learning," *Pattern Recognition*, vol. 46, no. 1, pp. 397–411, 2013.

[21] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81)*, vol. 2, pp. 674–679, Vancouver, Canada, 1981.

[22] F. Wang, S. Yu, and J. Yang, "A novel fragments-based tracking algorithm using mean shift," in *Proceedings of 10th International Conference on Control, Automation, Robotics and Vision (ICARCV '08)*, pp. 694–698, December 2008.

[23] F. Tang, S. Brennan, Q. Zhao, and H. Tao, "Co-tracking using semi-supervised support vector machines," in *Proceedings of the IEEE 11th International Conference on Computer Vision (ICCV '07)*, pp. 1–8, October 2007.

[24] Z. Kalal, K. Mikolajczyk, and J. Matas, "Face-TLD: tracking-learning-detection applied to faces," in *Proceedings of the 17th IEEE International Conference on Image Processing (ICIP '10)*, pp. 3789–3792, Hong Kong, September 2010.

[25] Z. Kalal, J. Matas, and K. Mikolajczyk, "Online learning of robust object detectors during unstable tracking," in *Proceedings of the IEEE 12th International Conference on Computer Vision Workshops (ICCV '09)*, pp. 1417–1424, October 2009.

[26] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409–1422, 2012.

[27] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*, Prentice Hall, Upper Saddle River, NJ, USA, 1996.

[28] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[29] M. Everingham, L. V. Gool, C. Williams et al., "Partbased visual trackin g with online latent structural learning," in *Proceedings of the PASCAL Visual Object Classes Challenge (VOC '10) Results*, 2010.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 196415, 8 pages
http://dx.doi.org/10.1155/2014/196415

Research Article

Moving Object Localization Using Optical Flow for Pedestrian Detection from a Moving Vehicle

Joko Hariyono, Van-Dung Hoang, and Kang-Hyun Jo

Graduate School of Electrical Engineering, University of Ulsan, Ulsan 680-749, Republic of Korea

Correspondence should be addressed to Kang-Hyun Jo; acejo@ulsan.ac.kr

Received 9 April 2014; Revised 7 June 2014; Accepted 8 June 2014; Published 10 July 2014

Academic Editor: Yu-Bo Yuan

This paper presents a pedestrian detection method from a moving vehicle using optical flows and histogram of oriented gradients (HOG). A moving object is extracted from the relative motion by segmenting the region representing the same optical flows after compensating the egomotion of the camera. To obtain the optical flow, two consecutive images are divided into grid cells 14 × 14 pixels; then each cell is tracked in the current frame to find corresponding cell in the next frame. Using at least three corresponding cells, affine transformation is performed according to each corresponding cell in the consecutive images, so that conformed optical flows are extracted. The regions of moving object are detected as transformed objects, which are different from the previously registered background. Morphological process is applied to get the candidate human regions. In order to recognize the object, the HOG features are extracted on the candidate region and classified using linear support vector machine (SVM). The HOG feature vectors are used as input of linear SVM to classify the given input into pedestrian/nonpedestrian. The proposed method was tested in a moving vehicle and also confirmed through experiments using pedestrian dataset. It shows a significant improvement compared with original HOG using ETHZ pedestrian dataset.

1. Introduction

Vision-based environment detection methods have been actively developed in robot vision. Detecting pedestrian is one of the essential tasks for understanding environment. Pedestrian detection in images could be used in video surveillance systems and driver assistance systems. It is more challenging to detect moving objects or pedestrian in order to avoid an obstacle and control locomotion of the vehicle in the real-world environment.

In the past few years, moving object and pedestrian detection methods for a mobile robot or moving vehicle have been actively developed. For a practical real-time pedestrian detection system, Gavrila and Munder [1] employed hierarchical shape matching to find pedestrian candidates from moving vehicle. Their method uses a multicue vision system for the real-time detection and tracking of pedestrians. Nishida and Kurita [2] applied SVM with the automated selection process of the components by using AdaBoost. These researches show that the selection of the components and their combination are important to get a good pedestrian detector.

Many local descriptors are proposed for object recognition and image retrieval. Mikolajczyk and Schmid [3] compared the performance of several local descriptors and showed that the best matching results were obtained by the scale invariant feature transform (SIFT) descriptor [4]. Dalal et al. [5, 6] proposed a human detection algorithm using histograms of oriented gradients (HOG) which are similar to the features used in the SIFT descriptor. HOG features are calculated by taking orientation histograms of edge intensity in a local region. They are designed by imitating the visual information processing in the brain and have robustness for local changes of appearances and position. Dalal et al. extracted the HOG features from all locations of a dense grid on an image region and the combined features are classified by using linear SVM. They showed that the grids of HOG descriptors significantly out-performed existing feature sets for human detection. Kobayasi et al. [7] proposed selected feature of HOG using PCA to decrease the number of features. It could reduce the number of features by less than half without lowering the performance.

Moving object detection and motion estimation methods using the optical flow for a mobile robot also have been actively developed. Talukder et al. [8] proposed a qualitative obstacle detection method that was proposed using the directional divergence of the motion field. The optical flow pattern was investigated in perspective camera and this pattern was used for moving object detection. Also, real-time moving object detection method was presented during translational robot motion.

Several researchers also developed methods for egomotion estimation and navigation from a mobile robot using an omnidirectional camera [9, 10]. They tried to measure camera egomotion itself using omnidirectional vision. They used Lucas Kanade optical flow tracker and obtained corresponding features of background in the consecutive two omnidirectional images. The motion of feature points analysis is used to calculate camera egomotion, however they didn't use for moving object detection. They set up an omnidirectional camera on a mobile robot and obtained panoramic image transformed from omnidirectional image. They obtained camera egomotion compensated frame difference based on an affine transformation of two consecutive frames where corner features were tracked by Kanade-Lucas-Tomasi (KLT) optical flow tracker [11]. However, detecting moving objects resulted in a problem that only one affine transformation model could not represent the whole background changes since the panoramic image has many local changes of scaling, translation, and rotation of pixel groups. For this problem, our previous work [12] proposed that each affine transformation of local pixel groups should be tracked by KLT tracker. The local pixel groups are not a type of image features such as corner or edge. We use grid windows-based KLT tracker by tracking each local sector of panoramic image (Figure 2) while other methods use sparse features-based KLT tracker. Therefore, we can segment moving objects in panoramic image by overcoming the nonlinear background transformation of panoramic image [13].

2. Related Works

Proposed method is inspired by the works on pedestrian detection from moving vehicle [1, 8], using optical flow [11] and egomotion estimation [9], we called it is egomotion compensate [12]. Pedestrian as a moving object is extracted from the relative motion by segmenting the region representing the same optical flows after compensating the egomotion of the camera. To obtain the optical flow, image is divided into grid windows and affine transformation is performed according to each window, so that conformed optical flows are extracted. The regions of moving object are detected as transformed objects are different from the previously registered background. Morphological process is applied to get the candidate region of human shape. In order to recognize the object, HOG features were extracted on a candidate region and classified using linear SVM [5, 14]. The HOG feature vectors are used as an input of linear SVM to classify the given input into pedestrian/nonpedestrian. For the performance evaluation, comparative study was presented in this paper.

3. Moving Object Segmentation

This section presents how to detect moving object from the camera mounted on the vehicle. In order to obtain moving object area from video or sequent of images, it is not easy to segment out only moving object area, because the camera moving is also caused by camera egomotion. So, we proposed a method to deal with this situation [12]. We used optical flow analysis to segment independent motion of moving object from egomotion caused by camera. It is called egomotion compensated. The optical flow caused by independent motion of moving object will have different pattern compared with flow caused by egomotion from camera; then, we localize those different pattern as a region of moving object. This region is candidate of detected human/pedestrian after we apply HOG. The overview of the pedestrian detection algorithm is shown in Figure 1.

3.1. Egomotion Compensated.

3.1. Egomotion Compensated. In our previous work [12], we apply KLT optical flow tracker [11] in order to deal with several conditions. Brightness constancy, which is projection of the same point, looks the same in every frame; small motion that points do not move very far and spatial coherence that points move like their neighbors.

The frame difference represents all motions caused by camera egomotion and moving object in the scene. It needs to compensate this effect from frame difference to segment out only independent motion of moving object, so how much the background image has been transformed in two sequences of images. Affine transformation represents the pixel movement between two sequence images as follows:

$$P' = AP + t, \tag{1}$$

where P and P' are pixel location in the first and the second frame. A is transformation matrix and t is translation vector. Affine parameters are calculated by least square method using at least three corresponding features in two images.

In this work, the original input images are converted to grayscale images, and one channel intensity pixel value from the input images is obtained. Then, use two consecutive images which are divided into grid cells of size 14 × 14 pixels; then compare and track each cell in current frame to find corresponding cell in the next frame. The cell that has the most similar intensity value in a group will be selected as corresponding value. Using method from [11], find the motion distance of each pixel in a group of cells, the motion d in x-axis and y-axis of each cell $g_{t-1}(i, j)$, by finding most similar cell $g_t(i, j)$ in the next frame,

$$g_{t-1}(i, j) = g_t\left(i + d_x, j + d_y\right), \tag{2}$$

where d_x and d_y are motion distances in x-axis and y-axis, respectively. At least three corresponding features are used to estimate the affine parameters using the least square method.

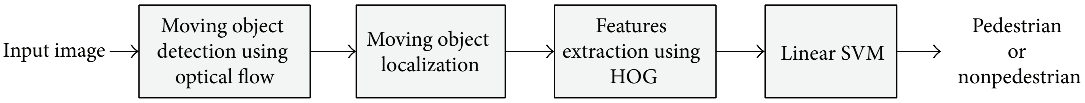

Input image
Moving object detection using optical flow
Moving object localization
Features extraction using HOG
Linear SVM
Pedestrian or nonpedestrian

Figure 1: The overview of the pedestrian detection algorithm.

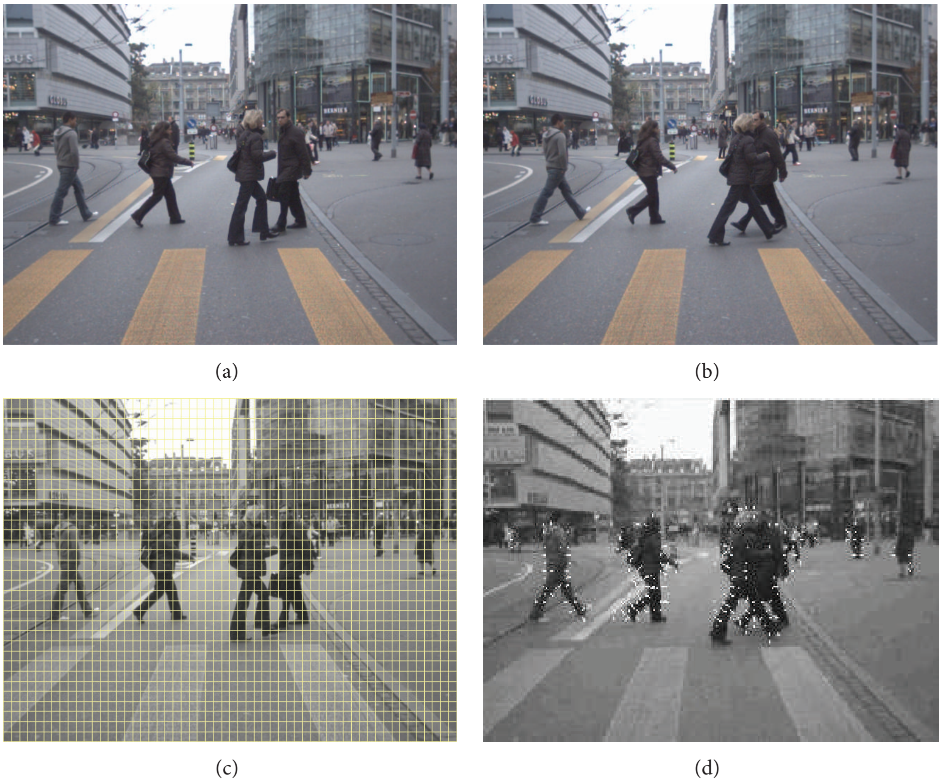

(a)
(b)
(c)
(d)

Figure 2: From two consecutive image sequences (a) and (b), we decide grid windows (c) and track each window in the next consecutive image (d).

Equation (2) is rewritten by affine transformation of each pixel in the same cell as follows:

$$I_t(x, y) = AI_{t-1}(x, y) + d, \quad (3)$$

where $I_t(x, y)$ and $I_{t-1}(x, y)$ are vector 2×1 which represent pixel location in the current and previous frame, respectively; A is 2×2 projection matrix and d is 2×1 translation vector. The results are shown in Figure 3.

To obtain the camera egomotion compensated, frame difference is applied in two consecutive input images by calculating based on the tracked corresponding pixel cells using

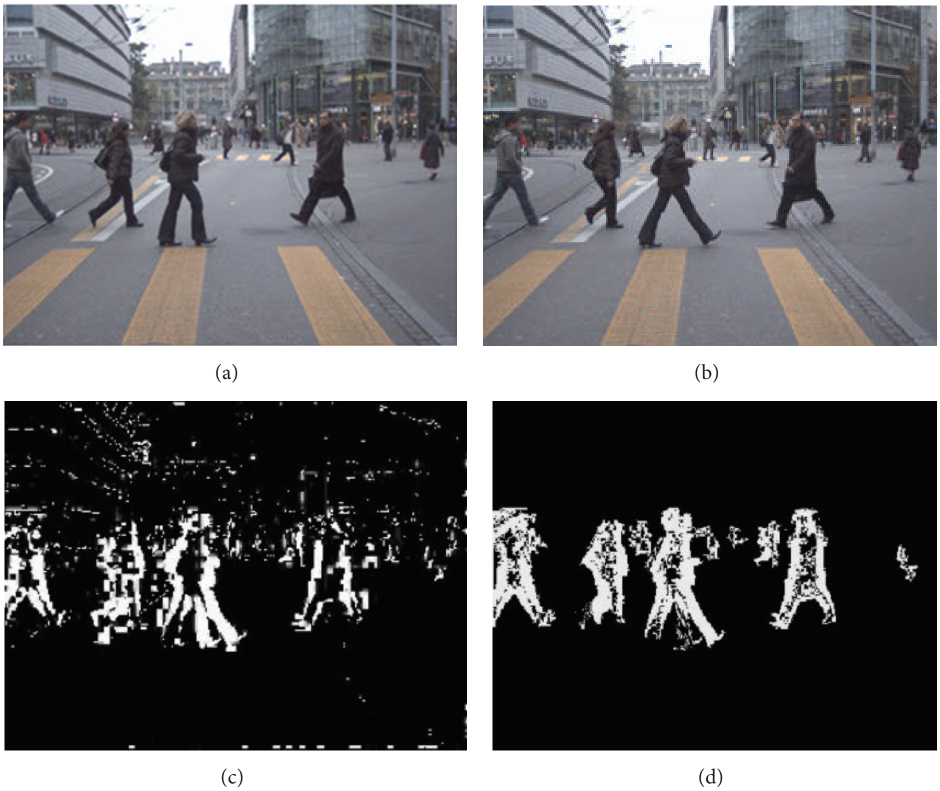
$$I_d(x, y) = \left|I_{t-1}(x, y) - I_t(x, y)\right|, \quad (4)$$

where $I_d(x, y)$ is a pixel cell located at (x, y) in the grid cell.

Suppose that two consecutive images shown in Figures 3(a) and 3(b) cannot segment out moving object using frame difference Figure 3(c), however when we apply frame difference with egomotion compensate could obtain moving objects area shown in Figure 3(d).

3.2. Moving Object Localization. Each pixel output from frame difference using egomotion compensated cannot show clearly as silhouette. It just gives information of motion areas from moving objects. Those moving areas are applied to morphological process to obtain region of moving object and noise removal.

Ideally, we would seek to devise a region segmentation algorithm that accurately locates the bounding boxes of the motion regions in the difference image. Given the sparseness of the data, however, accurate segmentation would involve the enforcement of multiple constraints, making fast implementation difficult. To achieve faster segmentation, we assumed the fact that humans usually appear in upright positions and conclude that segmenting the scene into vertical strips is sufficient most of the time. In this work, we define detected moving objects that are represented by the position


(a)

(b)

(c)

(d)

Figure 3: From two consecutive images (a) and (b), we applied frame difference (c) and comparing when we applied frame difference with egomotion compensated (d).

in width in x-axis. Using projection histogram h_x by pixel voting vertically projects image intensities into x-coordinate.

Adopting the region segmentation technique proposed in [15], we define the region using boundary saliency. It measures the horizontal difference of data density in the local neighborhood. The local maxima, which correspond to where maximal change in data density occurs, are candidates for region boundaries of pedestrian in moving object detection.

4. Feature Extraction

In this section, we present how we extract feature from candidate region obtained from previous section. In this work, we use histogram of oriented gradients (HOG) to extract features from moving object area localization. Local object appearance and shape usually can be characterized well by the distribution of local intensity gradients or edge direction. HOG features are calculated by taking orientation histograms of edge intensity in local region.

4.1. HOG Features. In this work, we extract HOG features from 16×16 local regions as shown in Figure 4. At first, we use Sobel filter to obtain the edge gradients, and orientations were calculated from each pixel in this local region. The gradient magnitude $m(x, y)$ and orientation $\theta(x, y)$ are calculated using directional gradients $dx(x, y)$ and $dy(x, y)$ computed by Sobel filter as

$$m(x, y) = \sqrt{dx(x, y)^2 - dy(x, y)^2}, \tag{5}$$

$$\theta(x, y) = \begin{cases} \tan^{-1}\left(\dfrac{dy(x, y)}{dx(x, y)}\right) - \pi, \\ \quad \text{if } dx(x, y) < 0, dy(x, y) < 0, \\ \tan^{-1}\left(\dfrac{dy(x, y)}{dx(x, y)}\right) + \pi, \\ \quad \text{if } dx(x, y) < 0, dy(x, y) > 0, \\ \tan^{-1}\left(\dfrac{dy(x, y)}{dx(x, y)}\right), \\ \quad \text{otherwise}. \end{cases} \tag{6}$$

The local region is divided into small spatial or cell, each size is 4×4 pixels. Histograms of edge gradients with 8 orientations are calculated from each of the local cells. Then the total number of HOG features becomes $128 = 8 \times (4 \times 4)$ and they constitute a HOG feature vector. To avoid sudden changes in the descriptor with small changes in the position of the window and to give less emphasis to gradients that are far from the center of the descriptor, a Gaussian

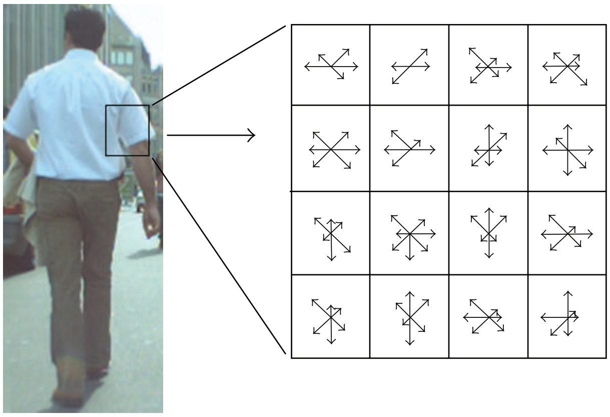


Figure 4: Extraction process of HOG features. The HOG features are extracted from local regions with 16 × 16 pixels. Histograms of edge gradients with 8 orientations are calculated from each of the 4 × 4 local cells.

weighting function with σ equal to one-half of the width of the descriptor window is used to assign a weight to the magnitude of each pixel.

A vector of HOG feature represents local shape of an object, it has edge information at plural cells. In flatter regions like a ground or a wall of a building, the histogram of the oriented gradients has flatter distribution. On the other hand, in the border between an object and background, one of the elements in the histogram has a large value and it indicates the direction of the edge. Even though the images are normalized to position and scale, the positions of important features will not be registered with the same grid positions. It is known that HOG features are robust to the local geometric and photometric transformations. If the translations or rotations of the object are much smaller than the local spatial bin size, their effect is small.

Dalal and Triggs [5] extracted a set of HOG feature vectors from all locations in an image grid and that are used for classification. In this work, we just extract the HOG features from all locations on the candidate region localization from an input image as shown in Figure 5.

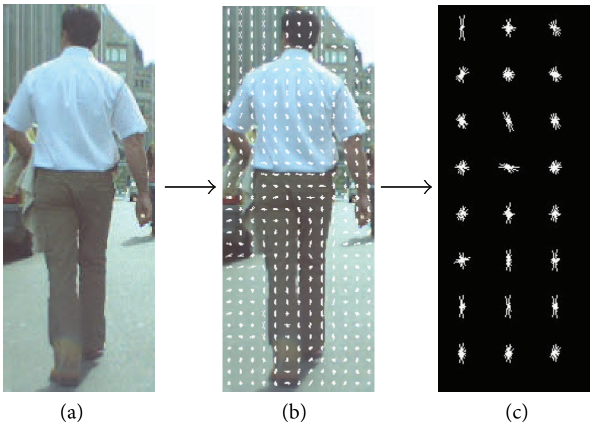


Figure 5: From a candidate input image of size 150 × 382 (a), HOG features are extracted from all locations on the candidate region of an input image with 16 × 16 pixels region (b), and the result is shown in (c).

4.2. Linear SVM Classifier. In the human detection algorithm proposed by Dalal and Triggs [5], the HOG features are extracted from all locations of a dense grid and the combined features are classified using linear support vector machine (SVM). HOG shows significantly outperformed existing feature sets for human detection. This work also used the linear SVM to perform work in various data classification tasks. Let $\{f_i, t_i\}_{i=1}^{N} (f_i \in R^D, t_i \in \{-1, 1\})$ be the given training sample in D-dimensional feature space. The classification function is given as

$$z = \operatorname{sign}\left(\omega^T f_i - h\right), \quad (7)$$

where ω and h are the parameters of the model. For the case of soft-margin SVM, the optimal parameters are obtained by minimizing

$$L(\omega, \xi) = \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{N}\xi_i \quad (8)$$

under the constraints

$$\xi_i \geq 0, \quad t_i\left(\omega^T f_i - h\right) \geq 1 - \xi_i \quad (i = 1, \ldots, N), \quad (9)$$

where ξ_i (≥ 0) is the error of the ith sample measured from the separating hyperplane and C is the hyperparameter which controls the weight between the errors and the margin. The dual problem of (8) is obtained by introducing Lagrange multipliers $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_N), \alpha_k \geq 0$ as

$$L_D(\boldsymbol{\alpha}) = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j t_i t_j f_i^T f_i \quad (10)$$

under the constraints

$$\sum_{i=1}^{N}\alpha_i t_i = 0, \quad 0 \leq \alpha_i \quad (i = 1, \ldots, N). \quad (11)$$

By solving (10), the optimum function is obtained as

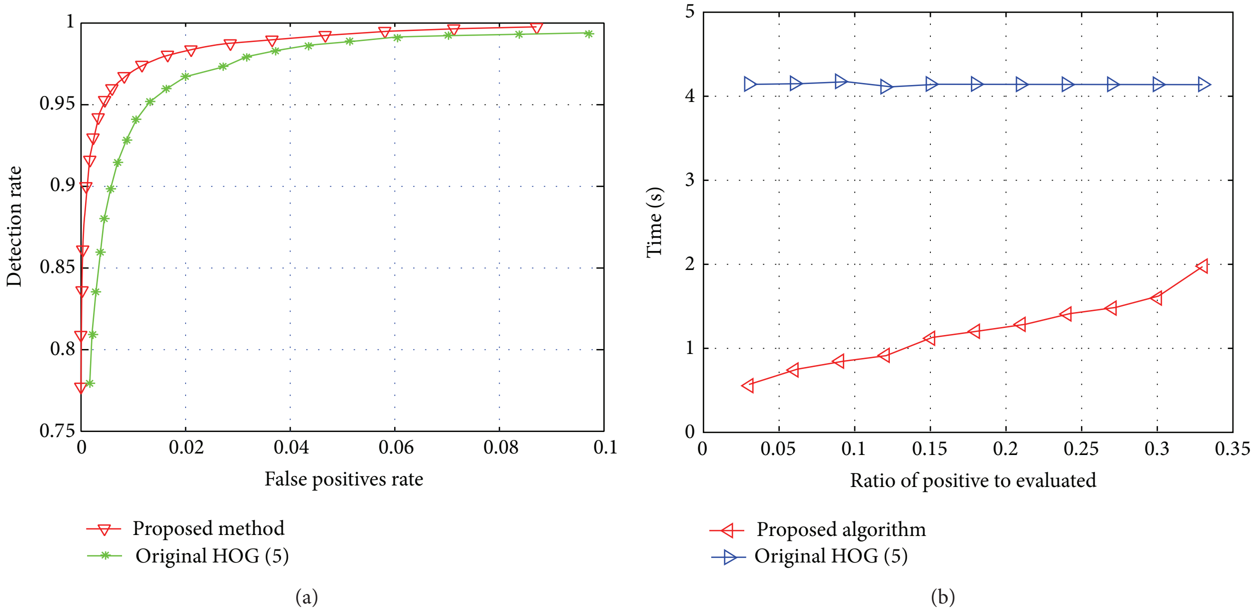
$$z = \operatorname{sign}\left(\sum_{i \in S}\alpha_i^* t_i f_i^T f_i - h^*\right), \quad (12)$$

where S is the set of support vectors.

To get a good classifier, we have to search the best hyperparameter C. The cross-validation is used to measure the goodness of the linear SVM classifier.

5. Experimental Results

In this work, our vehicle system is run in outdoor environment with speed that varies from around 0 to 50 kilometers



(a)

(b)

Figure 6: Comparison result when we tested our proposed method and original HOG by Dalal et al. (a) Comparison of detection rate and (b) comparison of time consumption.

per hour and detected object moving surround its path. Proposed algorithm was programmed in MATLAB and executed on a Pentium 3.40 GHz, 32-bit operating system with 8 GB random access memory. The proposed algorithm was evaluated by using five sequences of images from ETHZ pedestrian datasets which contain around 5,000 images of pedestrians in city scenes [15]. It contains only front or back views with relatively limited range of poses and the position and the height of human in the image are almost adjusted. The size of the image is 640 × 480 pixels. For the training process, we used person INRIA datasets [5]. These images were used for positive samples in the following experiments. The negative samples were originally collected from images of sky, mountain, airplane, building, and so forth. The number of images is 3,000. From these images, 1,000 person images and 2,000 negative samples were used as training samples to determine the parameters of the linear SVM. The remaining 100 pedestrian images and 200 negative samples were used as test samples to evaluate the recognition performance of the constructed classifier.

We studied methods for detecting human, and one of the objectives of this work is that we want a method that can detect people reliably whether they are moving or not. We were concerned that it might be sensitive to the relative proportion of moving and of static people in the videos. We check reliability of the proposed method that the combination of optical flow and HOG not only on the pure video contain of moving object, but also on objects without moving on the sequent images again with static object flows being zero. The results are diluting the fraction of motion regions naturally reduces the advantage of the combination of methods relative to the static ones; however, using the combination of methods, the relative ranking of the methods remains unchanged. Table 1 shows that when we used on relatively the objects without moving on the images for which there are a less flow field, the best combination of methods detectors do marginally better than the best of original HOG detectors done.

The reliability of our moving object detection system was evaluated whether it still works well in the case if the vehicle ran in varying speed. Outdoor application with speed of vehicle that varies from around 0 to 50 kilometers per hour was performed; then we evaluated the proposed window cells based flow estimation which are still visible at several levels. We tested reliability of the window cells for optical flow tracking in several sizes; it will determines from the relative distance of the object from the camera, so that we consider to choose the flow field window tracking which is more accurate for larger people and also well tracking for smaller people in the image. As a counterweight parameter, computational cost was considered for performance balancing. Table 2 shows the miss detection rate and computational cost of several windows size. However, 10 × 10 cells are the lowest on the miss detection rate but the slowest in computational cost; size 14 × 14 is selected based on low in miss detection rate and faster computational speed.

After all, we implemented original HOG by Dalal et al. using those datasets; the recognition rate for test dataset is 98.3%. Then, we test the combination of methods based on optical flow and HOG feature. HOG feature vectors were extracted from all locations of the grid for each training sample. Then, the selected feature vectors were used as input of the linear SVM. The selected subsets were evaluated by cross validation. Also, we evaluated the recognition rates of the constructed classifier using test samples.

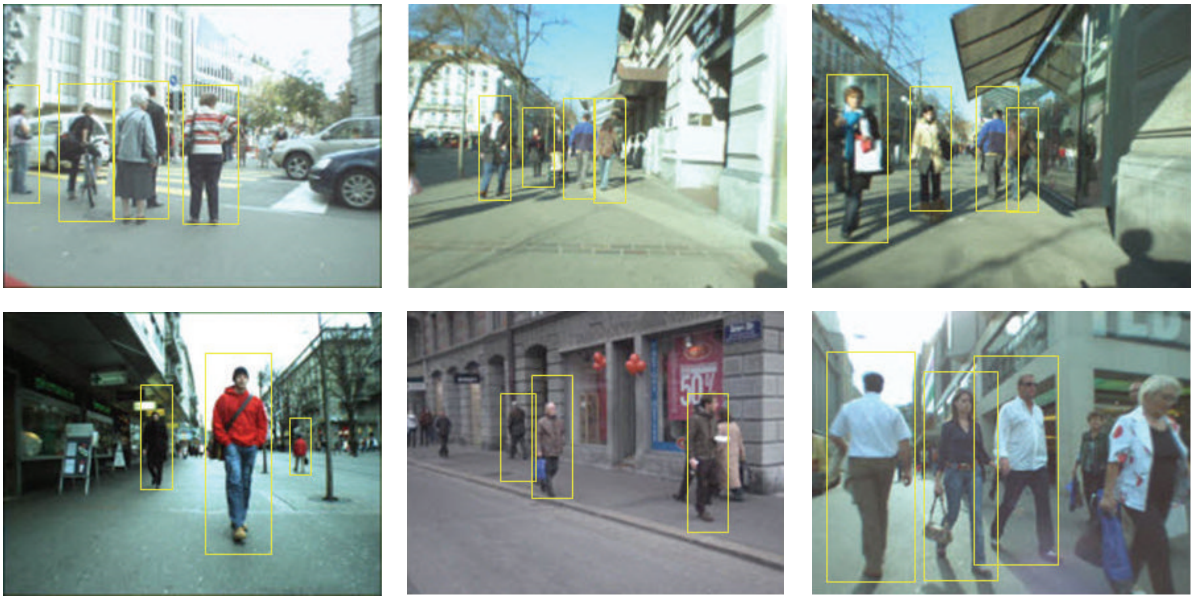
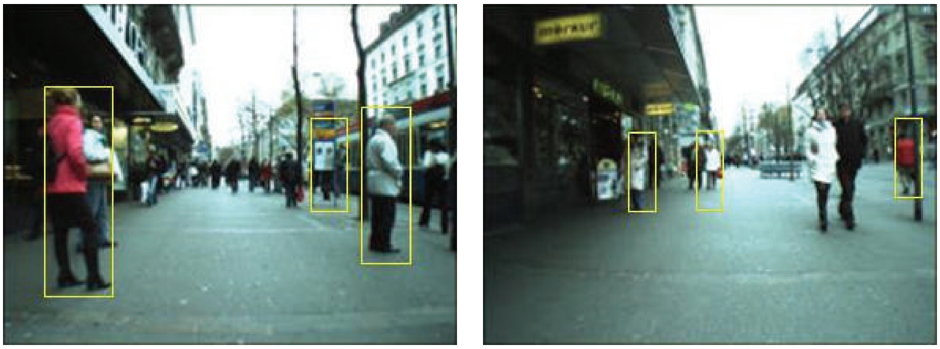


Figure 7: Successful moving objects detection results.



(a)

(b)

Figure 8: (a) False positives detection and (b) false negative detection.

Table 1: The detection rates of various detectors.

False positive rate	0.02	0.04	0.06	0.08	0.1
HOG	0.965	0.980	0.985	0.990	0.992
Proposed method	0.980	0.986	0.989	0.992	0.994

Table 2: The miss rates of various cell windows size.

Cell window size	False positive rate (0.09)	Computational cost (fps)
10 × 10	0.023	11.89
12 × 12	0.024	11.97
14 × 14	0.024	12.35
16 × 16	0.026	12.55
18 × 18	0.028	12.83

The relation between the detection rates and the number of false positive rate is shown in Figure 6. The best recognition rate, 99.3%, was obtained at 0.09 false positive rates. It means that we obtain higher detection rate with smaller false positives rate. The computational cost also reduces eight times better when we use small ratio of positive to evaluated data. However, if we increase the number of ratios it also reduces time consuming significantly. The detection results are shown in Figure 7 and false detection is shown in Figure 8.

6. Conclusion

This paper addressed the problem for detecting pedestrian from moving vehicle using optical flow and HOG. The moving object is segmented out through the relative evaluation of optical flows to compensate egomotion of camera. Morphological process is applied to get the candidate region of pedestrian. In order to recognize the object, HOG features were extracted on a candidate region and classified using linear SVM. The HOG feature vectors are used as an input of linear SVM to classify the given input into pedestrian/nonpedestrian. The proposed algorithm achieved comparable results compared with original HOG and also reduces computational cost significantly using moving object localization. In the future work, we consider the combination methods [16] compared with modification of HOG, such as LBP HOG and feature selection HOG.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This research was supported by the 2014 Research Fund of University of Ulsan.

References

[1] D. M. Gavrila and S. Munder, "Multi-cue pedestrian detection and tracking from a moving vehicle," *International Journal of Computer Vision*, vol. 73, no. 1, pp. 41–59, 2007.

[2] K. Nishida and T. Kurita, "Boosting soft-margin SVM with feature selection for pedestrian detection," in *Proceeding of International Workshop on Multiple Classifier Systems*, vol. 13, pp. 22–31, 2005.

[3] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.

[4] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[5] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, pp. 886–893, San Diego, Calif, USA, June 2005.

[6] N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," in *Proceedings of the 9th IEEE European Conference on Computer Vision (ECCV '06)*, Graz, Austria, 2006.

[7] T. Kobayasi, A. Hidaka, and T. Kurita, "Selection of histograms of oriented gradients features for pedestrian detection," in *Proceedings of the 14th International Conference on Neural Information Processing (ICONIP '08)*, vol. 4985 of *Lecture Notes in Computer Science*, pp. 598–607, Kitakyushu, Japan, 2008.

[8] A. Talukder, S. Goldberg, L. Matthies, and A. Ansar, "Real-time detection of moving objects in a dynamic scene from moving robotic vehicles," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1308–1313, Las Vegas, Nev, USA, October 2003.

[9] R. F. Vassallo, S. Victor, and H. Schneebeli, "A general approach for egomotion estimation with omnidirectional images," in *Proceedings of the 3rd Workshop on Omnidirectional Vision*, pp. 97–103, Copenhagen, Denmark, 2002.

[10] H. Liu, N. Dong, and H. Zha, "Omni-directional vision based human motion detection for autonomous mobile robots," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, vol. 3, pp. 2236–2241, Hawaii, Hawaii, USA, October 2005.

[11] C. Tomasi and T. Kanade, "Detection and tracking of point features," *International Journal of Computer Vision*, vol. 9, pp. 137–154, 1991.

[12] J. Hariyono, V.-D. Hoang, and K.-H. Jo, "Human detection from mobile omnidirectional camera using ego-motion compensated," in *Intelligent Information and Database Systems*, vol. 8397 of *Lecture Notes in Computer Science*, pp. 553–560, 2014.

[13] J. Hariyono, L. Kurnianggoro, D. C. Wahyono, and K. H. Jo, "Ego-motion compensated for moving object detection in a mobile robot," in *Proceedings of the 27th IEA-AEI International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems*, vol. 8482 of *Lecture Notes in Computer Science*, pp. 289–287, Kaohsiung, Taiwan, 2014.

[14] V.-D. Hoang, M.-H. Le, and K.-H. Jo, "Hybrid cascade boosting machine using variant scale blocks based HOG features for pedestrian detection," *Neurocomputing*, vol. 135, pp. 357–366, 2014.

[15] A. Ess, B. Leibe, and L. Van Gool, "Depth and appearance for mobile scene analysis," in *Proceedings of the IEEE 11th International Conference on Computer Vision (ICCV '07)*, Rio de Janeiro, Brazil, October 2007.

[16] Y.-Y. Lu and H.-C. Huang, "Adaptive reversible data hiding with pyramidal structure," *Vietnam Journal of Computer Science*, pp. 1–13, 2014.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 460973, 12 pages
http://dx.doi.org/10.1155/2014/460973

Research Article

Estimating Body Related Soft Biometric Traits in Video Frames

Olasimbo Ayodeji Arigbabu,[1] Sharifah Mumtazah Syed Ahmad,[1] Wan Azizun Wan Adnan,[1] Salman Yussof,[2] Vahab Iranmanesh,[1] and Fahad Layth Malallah[1]

[1] *Department of Computer and Communication Systems Engineering, Universiti Putra Malaysia (UPM), 43400 Serdang, Selangor, Malaysia*
[2] *Department of Systems and Networking, Universiti Tenaga Nasional, Jalan IKRAM-Uniten, 43000 Kajang, Malaysia*

Correspondence should be addressed to Olasimbo Ayodeji Arigbabu; oa.arigbabu@gmail.com

Received 22 April 2014; Revised 18 June 2014; Accepted 22 June 2014; Published 9 July 2014

Academic Editor: Yu-Bo Yuan

Soft biometrics can be used as a prescreening filter, either by using single trait or by combining several traits to aid the performance of recognition systems in an unobtrusive way. In many practical visual surveillance scenarios, facial information becomes difficult to be effectively constructed due to several varying challenges. However, from distance the visual appearance of an object can be efficiently inferred, thereby providing the possibility of estimating body related information. This paper presents an approach for estimating body related soft biometrics; specifically we propose a new approach based on body measurement and artificial neural network for predicting body weight of subjects and incorporate the existing technique on single view metrology for height estimation in videos with low frame rate. Our evaluation on 1120 frame sets of 80 subjects from a newly compiled dataset shows that the mentioned soft biometric information of human subjects can be adequately predicted from set of frames.

1. Introduction

Many advances have been made in the domain of biometrics using physical or behavioral attributes like face, iris, fingerprint, hand geometry, voice, and signature for recognizing individuals [1, 2]. Traditional biometrics possesses high level of uniqueness, permanence, and distinctiveness. As a result, studies carried out on these traits have led to several significant and interesting findings. For instance, iris and fingerprint have been found out to be the most reliable biometric traits used for recognition. Likewise, face recognition is now a very popular and widely used mode of recognition across various fields [3]. In addition, many state-of-the-art algorithms have been proposed with optimum accuracy [4–6]. However, the level of intrusiveness, human compliance, computational cost, and time are amongst the disadvantages of these attributes.

Recently, soft biometrics has been introduced as a recognition technique that can make use of labels, measurements, and descriptions of individuals in surveillance or long distance videos for recognition in an unobtrusive and nonuser compliant way [7, 8]; although they cannot uniquely identify a particular individual due to lack of distinctiveness and permanence [9]. Advantages of soft biometrics include low computational cost and time. The data collection is user friendly and identification is enrolment free. More so, they can be used for filtering a large database, identification and reidentification of individuals, and the ability of being used at a distance in a less constrained environment. Utilizing a single soft biometric trait can be rather vague to an extent. But, in an event where traditional biometrics cannot be deployed due to constrain of distance between the subject and acquisition system, also, in a limited or group specific application whereby the number of users is relatively small and/or optimum accuracy is not generally required, combining a variety of soft biometric attributes can act as an intelligent identifier without the consent of the individuals. Therefore, they can be applied to limit the search for identity to a small category from a large pool of subjects, as such reducing the problem of distinctiveness and permanence, and at the same time providing detailed descriptions for recognition.

In previous literatures, soft biometrics has been mainly categorized into face-based, body-based, and accessory-based [10]. In the work carried out by Dantcheva et al. [11], the authors made use of several face related soft biometrics (like eye color, hair color, skin color, beard, and moustache) for face recognition. The traits were used to place subjects into several groups; then, the probability of collision between two randomly chosen authentication groups was used to measure the recognition rate of the system. Also, combination of face and soft biometric information (like skin and clothes color) were used to perform continuous authentication by [12]; since biometric systems only perform initial authentication at the beginning of a session, it can subject the system to possible impostor threat. As such, soft biometrics was used to occasionally update the identity of the users when they are not making any serious interaction with system. In addition, frontal to side face reidentification was carried out in [13], using both face- and accessory-based soft biometrics (like hair color, skin color, and clothes color).

While facial traits are more suited to fast recognition from close distance, body related attributes have been more adapted to surveillance environments. Denman et al. [14] proposed an approach for identification of individuals using size and clothes color across several camera views. The authors segmented the images into head, torso, and leg region for the estimation of height, relative size, and also color information. Velardo et al. [15] used estimated height, weight, and clothes color for identification across two camera views with protection of individuals' privacy. The previous works have merely utilized the body related soft biometric information for recognition, without actually considering the level of reliability of the prediction or estimation of the attributes, which can be important in matching the correct identity.

In this study, we focus on predicting two main body related soft biometrics traits, which are height and weight in an indoor environment. The possible application of the proposed method could be adapted to visual based patient caring and security guarding [16, 17]. The contribution in this paper is to evaluate the reliability of estimation of the attributes under changes in appearances and low video frame rate. We adopt the concept of single view metrology for height estimation and we propose a new computational approach to contactless weight estimation from video frames, whereby the subjects have varying appearances in an unconstrained indoor environment. Height and weight have been specifically focused on in this study because they are attributes that can be inferred from distance when facial information cannot be easily constructed. The rest of the paper is structured as follows. Section 2 presents the experimental dataset and describes the technical details and strategies adopted for compiling the data. In Section 3, the materials and methods utilized for the proposed system are explained in detail. Section 4 shows the experimental results achieved, and we summarize our work in Section 5.

2. Experimental Dataset

Our approach to this data collection is to compile soft biometric information in a free and naturalistic way in an unconstrained indoor environment using commercially available single view camera. Previous data used for soft biometrics research are usually annotated from gait dataset like CASIA [18] and video surveillance datasets like PETS 2006 [19] and VIPER [20], whereby subjects are tracked across multiple camera networks. Although the use of multiple cameras provides additional information about the scene and object being tracked, the practicality of using multiple acquisition systems for visual surveillance, especially in an indoor environment, incurs more computational cost and complexity. A single view camera system can be easily implemented and very affordable and the computational complexity involved is much minimized, whilst providing adequate information for recognition.

Moreover, there is currently no publicly available dataset to specifically evaluate the performance of full body measured information such as height and body weight model estimated from video.

2.1. Data Acquisition. The dataset, UPM SOFTBIO, was captured and recorded in the Computer and Embedded Systems Laboratory, Universiti Putra Malaysia. The dataset was compiled between May and June 2013, in an indoor environment under uncontrolled illumination. It involved 101 subjects, who willingly volunteered to participate. The entire process was divided into 3 sessions in order to accommodate more intra- and intersession variability, as shown in Figure 1.

In the first session, we assigned a user ID and registered each subject with their metadata like age and gender and manually measured each subject's respective height and weight. After registration was completed, the only instruction given to the subjects was to walk towards and across the camera view at their individual normal stride rate. In the second and third sessions, the subjects were captured in their changed appearances, for instance, carrying backpack and changed clothes, shoes, and hair style.

There was an overall 15-day period between each subsequent session. All the 101 subjects successfully completed the first session, while 80 subjects appeared in two sessions, and 70 subjects were able to participate in the third session. The camera is a single view camera, recording at 50 frame rate per second, with resolution of 1440 × 1080. The height of the camera from ground plane is 120 cm, positioned at a distance of 700 cm to the subjects. The walk across the camera was performed to cover a distance of 600 cm, which depicts full profile of left and right side pose. The walk towards the camera covered 400 cm, representing full profile of frontal and back views and we allowed an extra 100 cm for subjects to be able to make a turn to another direction. Some landmark points were placed on the floor of the camera field of view to indicate where the subjects should start and end the walk.

2.2. Technical Details. The dataset has been annotated and organized based on pose, appearance, and time variability. The first set is composed of frame sequence and video recordings from the first session recordings. The second set composing different appearances from the second session is divided into two groups; articulation with backpack and articulation without backpack. Also, the third set, containing another different appearance from the third session, is divided in the same way as the second set. An empty background was recorded for each participant at the end of the walk.

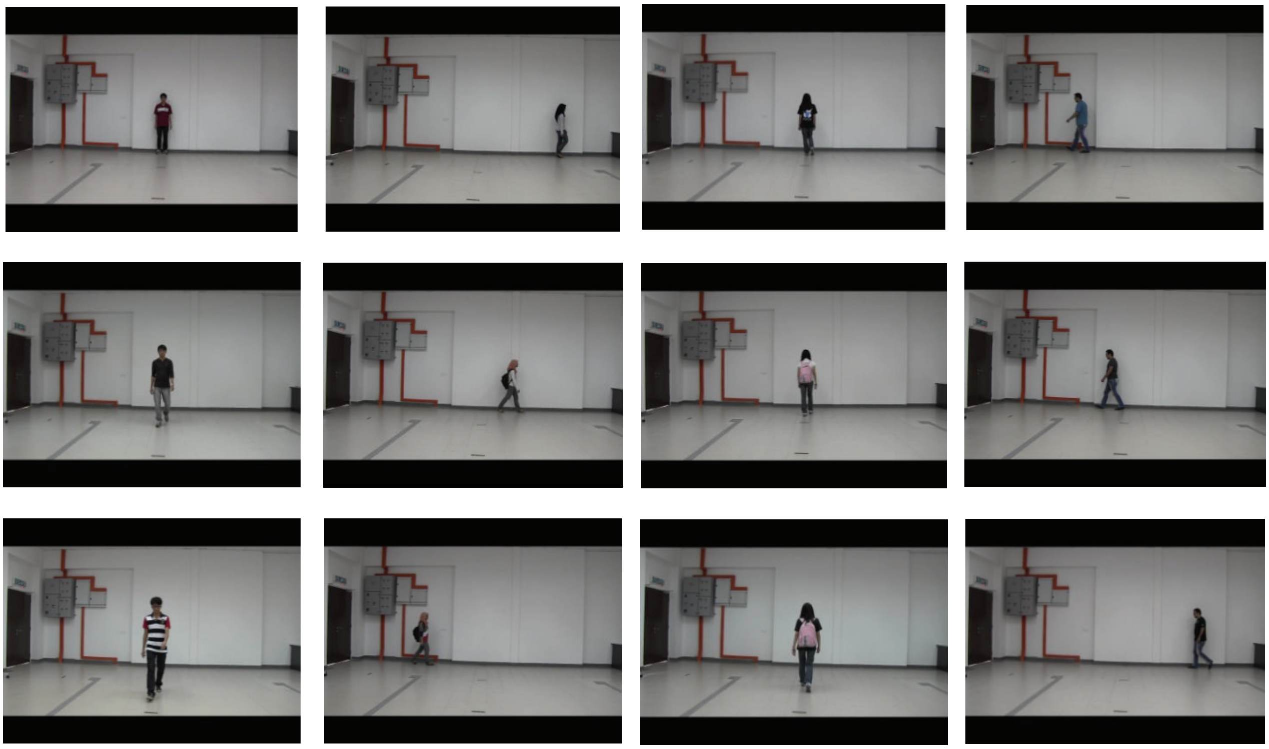


Figure 1: Sample images of four subjects appearing at three different times, covering four different directions, with variation in appearances.

The time required for a participant to complete the data collection task is ~4 mins, starting from the manual collection of the metadata like height, weight, gender, age, and ethnicity to the completion of the walk. The recording of the walk of each participant elapsed between 35 sec~1 min 10 sec, depending on the stride rate. The minimum number of frames is ~1750, while the maximum number of frames is ~3500. The camera has a storage capacity of 64 GB; as such the transfer of data was performed at the completion of each session. The frames and metadata for each participant have been extracted and stored on the desktop and external hard drive with a folder for each subject, which we annotated with user numbers.

2.3. Demographic Details. Demographic distribution is a very important tool in soft biometrics research. The 101 subjects that participated in the data compilation were from different ethnicities, gender, and age group. 51% were from South Asia (29% Chinese, 22% Malay), 28% were from the Middle East, 16% were from Central Asia, and 6% were Africans. 58% were male and 43% female and 62% were between 20 and 30 years, while 39% were between 31–45 years. The height range was between 144 and 197 cm and the weight was between 40 and 119 kg. The distribution of the subjects is depicted in Figure 2.

3. Materials and Methods

The techniques we propose are to compute the weight of moving subjects using features from the body and feed forward neural network (FFNN), and also to adapt the‘concept of single view metrology to low frame rate video for estimating upright human height.

3.1. Object Extraction. Extraction of the object as silhouette from the background is a very important step for the proposed system. In this work, the steps for the object extraction include background subtraction and shadow removal from the extracted object. Consider an input frame of an empty background, bg, and a current frame, curr, containing a moving subject, where all the pixels are within the RGB color space. The difference, D, between the two frames is computed using

$$D(x,y,i) = \sqrt{\text{sum}[(b(\text{bg}(x,y,i))) - (b(\text{curr}(x,y,i)))]^2}, \quad (1)$$

where b is the R, G, B band of each frame and D is a difference image resulting from the subtraction of the two frames, as shown in Figure 3.

Then, to determine the foreground F_M, Otsu thresholding [21] technique is used to automatically define a threshold T_{thres}. But, in this experiment, we introduced a threshold suppression parameter, c, in (2), which is a value greater than 1:

$$F_M(x,y,i) = \begin{cases} 1, & \text{if } D(x,y,i) > \dfrac{T_{\text{thres}}}{c} \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

The reason for introducing parameter, c, is to enhance the contrast of the foreground image. Furthermore, the determined threshold T_{thres} is observed to be too strict, as

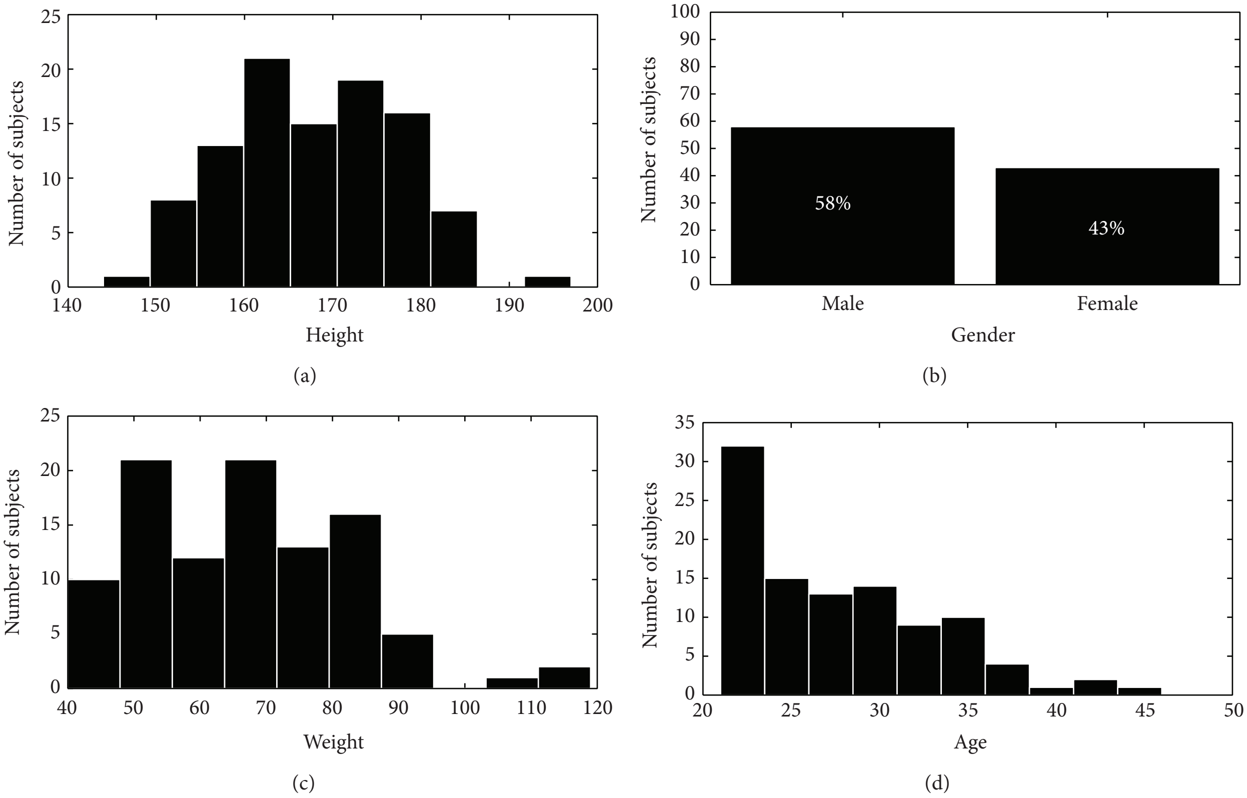


Figure 2: Plot of metadata of subjects according to height, gender, weight, and age distribution.

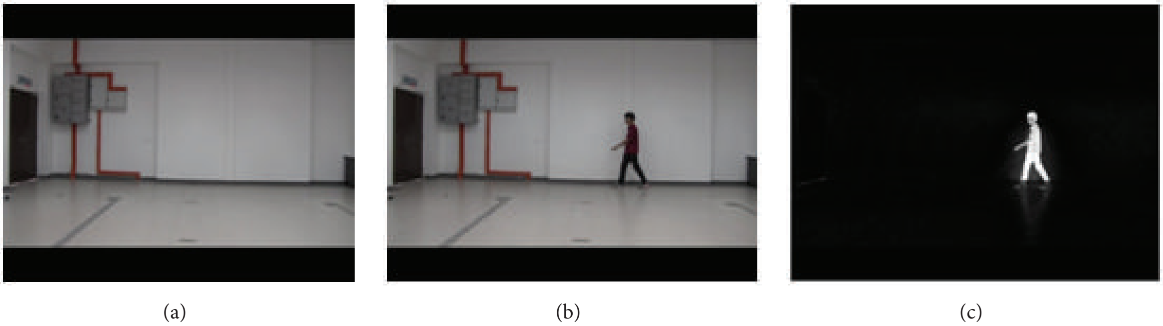


Figure 3: Process of object extraction. (a) Empty background frame. (b) New frame with walking subject. (c) Difference image.

it segments the leg region of most subjects as part of background; an example is shown in Figure 4(a). This is because Otsu technique mainly depends on bimodal distribution of histograms of the two classes. Thus, effect of varying illumination can significantly influence the computation of class variance.

Although improving the contrast consequently results in considering incidental pixels, due to shadow cast, as part of the foreground image, in order to remove the shadows that are extracted with the object, a color and brightness difference, B_D, is computed as [22]

$$B_D = \left| \log \left(\frac{I_b}{I_f} \right) \right| + D_{\mathrm{FB}}, \tag{3}$$

where D_{FB} is the difference between the new frame pixels and the background frame in the normalized GB space, while I_b is background brightness, and I_f is new frame brightness, in RGB space. If $B_D < T$, which is empirically estimated, it is regarded as lighting changes, so the pixels are removed. The output is a brightness difference image, shown in Figure 5(a).

But to highlight the brightness difference around the moving object, as shown in Figure 5(b), we used a simple logical AND operator, in the following expression:

$$\text{Shadow} = F_M \text{ AND } B_D. \tag{4}$$

Finally, to extract the well enhanced foreground image, F_{img}, the following expression is utilized:

$$F_{\mathrm{img}} = F_M \text{ AND } (1 - \text{shadows}). \tag{5}$$

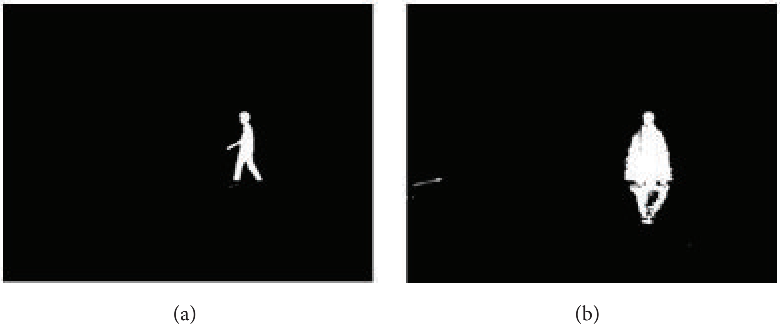


Figure 4: Threshold suppression. (a) Initial segmentation using Otsu threshold. (b) Segmentation by adding parameter c.

The resultant is a clear foreground image, F_{img}, with some random noises, considered as small regions that are post processed by applying morphological operations, as shown in Figure 6. The described technique performs effectively well for our task since the experiment is limited to indoor environment. Though, we note that, in more challenging outdoor scene, object detectors based on more robust local features can be deployed [23, 24].

3.2. Height Estimation. Several methods have been proposed for estimation of human height. There are two main techniques, one of which includes the use of intrinsic parameters of the camera retrieved from camera calibration [25, 26]. The other technique usually referred to as uncalibrated technique includes the use of information from the scene as extrinsic parameters of the camera. The aim in this method is to accurately locate the unknown camera centre or position in the image view with respect to a known reference frame. The most popular one based on 3D affine measurements was presented by Criminisi et al. [27]. Our estimation is mainly based on the technique proposed by Criminisi et al. (see [27]), but quite notable from their work is that a single image has been used for height estimation of stationary objects. This experiment incorporates the technique for estimating height of moving objects in frames of video that possess extremely low frame rate and we used only the height of the acquisition camera as a reference height. Also, the object extraction and vanishing point estimation are performed automatically.

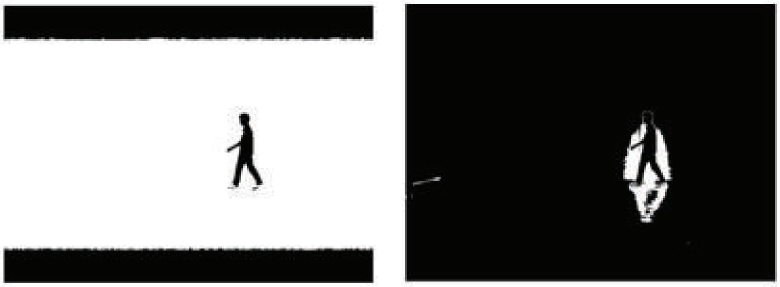
In the concept of single view metrology, if we consider the top and bottom points x_t and x_b of an object in 3D real world view and, also, consider another point y in the same field of view, the distance d between y and x_b represents the height of an object. Distance, d, in this case is regarded as the reference height in real world measurement. Briefly note here that points x_t and x_b are on different planes. Hence, a line projection from the two planes, which tend to infinity, results in another point v, referred to as the vanishing point. As a result, x_t, x_b, y, and v mark a set of collinear points in the world coordinate. Therefore, a simple cross-ratio between the points can be used to obtain the objects height H in real world view. However, in the 2D image view, the projection of points x_t, x_b, and v denote image points X_T, X_B, and V. Hence, the main problem is estimating V, which is also the camera position in the image. If V is known, then the object's height O_H can be estimated, by calculating the proportion of the camera with respect to points X_T, X_B, at the current position of the object in the image view. Once we are able to represent the camera position in image, then, we computed height as follows:

$$O_H = \left(\frac{H_C}{V}\right) * (X_T - X_B), \qquad (6)$$

where X_T is top of the ROI boundary, X_B is bottom of the ROI boundary, $V = X_B - Y$ is coordinate of the horizontal vanishing line, and H_c is height of camera (reference height).

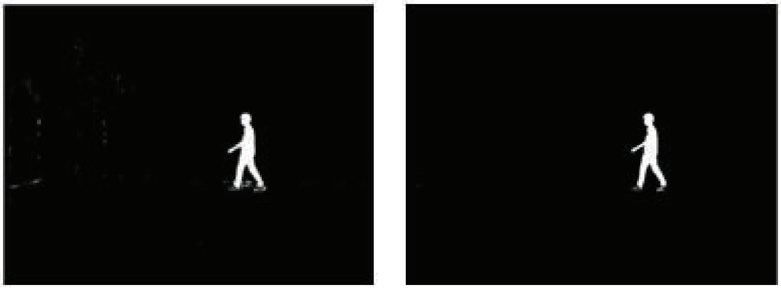
3.3. Weight Estimation. Human weight is another body based attribute which represents the stature along with height. Motivated by recent advances in computer based image analysis, whereby there is a common interest in predicting human measures directly from image, conventional methods make use of weighing scale for measuring body weight. This method is not useful in many conditions whereby the estimate of human weight is considered vital information, for instance, situations whereby an offender is described by a human observer based on the estimate of the body weight [28] or in visual surveillance reidentification [15]. Therefore, to predict the body weight, the only measures that can be performed are restricted to the image frames acquired by the camera. It is important to emphasise that precise measures can be significantly influenced by noise, since the features are extracted from image.

Nevertheless, we gain an insight into this problem using computational intelligence methods. To the best of our knowledge, related works on weight estimation from image are very limited. Previously, Velardo and Dugelay [29] presented a method for weight estimation, by mapping manually extracted anthropometric measures of the whole human body to their respective weight using linear regression model for static subjects from frontal-to-side view. More recently, Labati et al. [30] proposed a computational intelligence approach to weight estimation, whereby the length, area, and volume measurements were automatically extracted across eight segments of the body relative to height at each segment as features. Neural network was used for mapping between measurements and weight of subjects. The estimation was performed with two calibrated cameras for 20 subjects. Most of the previous experimentations were carried out on



(a) (b)

Figure 5: Highlighting brightness difference around moving foreground pixels.



(a) (b)

Figure 6: Resultant foreground image. (a) Foreground with random noise (b) Postprocessed foreground image.

limited datasets captured at the same time (day). Hence, the robustness of their techniques against appearance changes is not proven. The proposed method in this paper takes into consideration the effect of appearance changes as well as large fluctuation in strides of subject as a result of low frame rate.

We take advantage of the background subtraction by segmenting the foreground image into head, torso, and leg regions [14], as shown in Figure 7. In order to determine the regions of segmentation, we first calculate the horizontal projection of all foreground pixels of the object using (4). Consider

$$P_H(j) = \sum_{i=0}^{M} B(i, j), \tag{7}$$

where P_H is the horizontal projection and B is the binary image. Then, the regions of segmentation are automatically located as the minimum points using the following equations [14]:

$$\begin{aligned} H_{\text{region}} &= \operatorname*{argmin}_{i=0.1*I}^{0.3*I} \left(P_H(j)\right) \\ T_{\text{region}} &= \operatorname*{argmin}_{i=0.35*I}^{0.7*I} \left(P_H(j)\right), \end{aligned} \tag{8}$$

where H_{region} is the head region, which is located between the first 10 and 30 percent of P_H, while T_{region} represents the torso region, located between 35 and 70 percent of P_H, as illustrated in Figure 7.

From the three regions, twelve additional features are computed, in addition to the result from object height estimation, O_H. For each image I of a subject, the pixel densities H_{feat}, T_{feat}, and L_{feat} of head, torso, and leg region are calculated using the following:

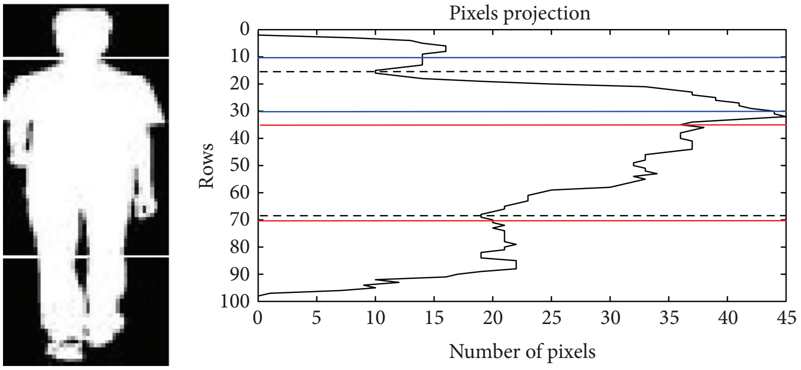
$$\begin{aligned} H_{\text{feat}} &= \sum_{n=1}^{n_i} \frac{(I_H)}{L_{HP}} \\ T_{\text{feat}} &= \sum_{n=1}^{n_i} \frac{(I_T)}{L_{TP}} \\ L_{\text{feat}} &= \sum_{n=1}^{n_i} \frac{(I_L)}{L_{LP}}, \end{aligned} \tag{9}$$

where L is the length of the segmented region in the image. Further, the size of the object, W_{feat}, is computed by dividing the area of the silhouette by difference of the head and bottom point in pixel:

$$W_{\text{feat}} = \frac{\text{Area}(I)}{(X_T - X_B)}. \tag{10}$$

The weighted ratio of the pixels of the head, torso, and leg region, R_H, R_T, and R_L, is calculated by giving more significance to the numerator:

$$\begin{aligned} R_H &= \frac{(H_{\text{feat}})^2}{T_{\text{feat}} + L_{\text{feat}}} \\ R_T &= \frac{(T_{\text{feat}})^2}{H_{\text{feat}} + L_{\text{feat}}} \\ R_L &= \frac{(L_{\text{feat}})^2}{H_{\text{feat}} + T_{\text{feat}}}. \end{aligned} \tag{11}$$



(a)

(b)

Figure 7: The region segmentation of an object. (a) represents the head torso and leg region and (b) depicts the horizontal projection, with the minimum points of head and torso highlighted with the dotted lines.

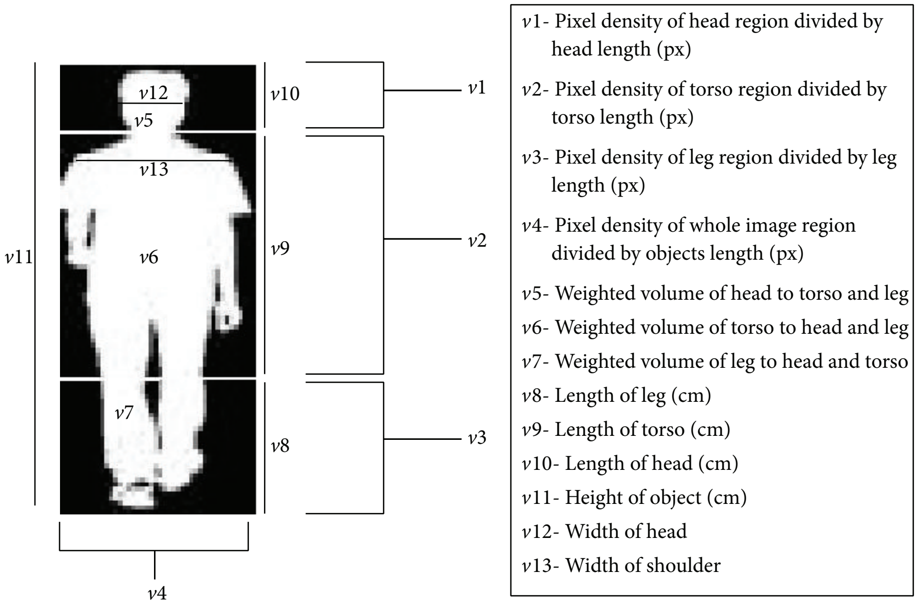

v12
v5
v10
v1
v13
v11
v6
v9
v2
v7
v8
v3
v4
v1- Pixel density of head region divided by head length (px)
v2- Pixel density of torso region divided by torso length (px)
v3- Pixel density of leg region divided by leg length (px)
v4- Pixel density of whole image region divided by objects length (px)
v5- Weighted volume of head to torso and leg
v6- Weighted volume of torso to head and leg
v7- Weighted volume of leg to head and torso
v8- Length of leg (cm)
v9- Length of torso (cm)
v10- Length of head (cm)
v11- Height of object (cm)
v12- Width of head
v13- Width of shoulder

Figure 8: The feature set for weight computation.

Also, the lengths of head, torso, and leg regions and objects height in centimeters (cm) are computed by using the following expressions:

$$L_L = \left(\frac{H_C}{V}\right) \times (L_{LP})$$

$$L_T = \left(\frac{H_C}{V}\right) \times (L_{TP} + L_{LP}) \qquad (12)$$

$$L_H = \left(\frac{H_C}{V}\right) \times ((X_T - X_B) - (L_{TP} + L_{LP})).$$

Finally, the width of the head W_H and width of shoulder W_S of the subjects are calculated by using region label techniques, whereby the pixels which belong to white are searched based on the connected components and the maxima of summations of all connected components are selected. For head, the search is limited to the midlevel of the head region, while, for the shoulder region, the search is restricted to the first 10 percent of the torso binary image, as shown in Figure 8.

As a result, 13 feature sets $[v1 \cdots v13]$ are considered for weight estimation. Furthermore, we exploited a computational intelligence method by passing the extracted feature vector to FFNN, to compute the weight of each subject represented in kg. Before modeling FFNN, the technique described in [31] is adopted for normalizing the features between the range of 0 and 1.

4. Experimental Results

We implemented the proposed system using 1120 video frames of 80 subjects from our newly compiled dataset, UPM SOFTBIO, with each subject walking across and towards the camera representing four different poses. For this experiment, the frames are selected at 1 frame per second (fps)

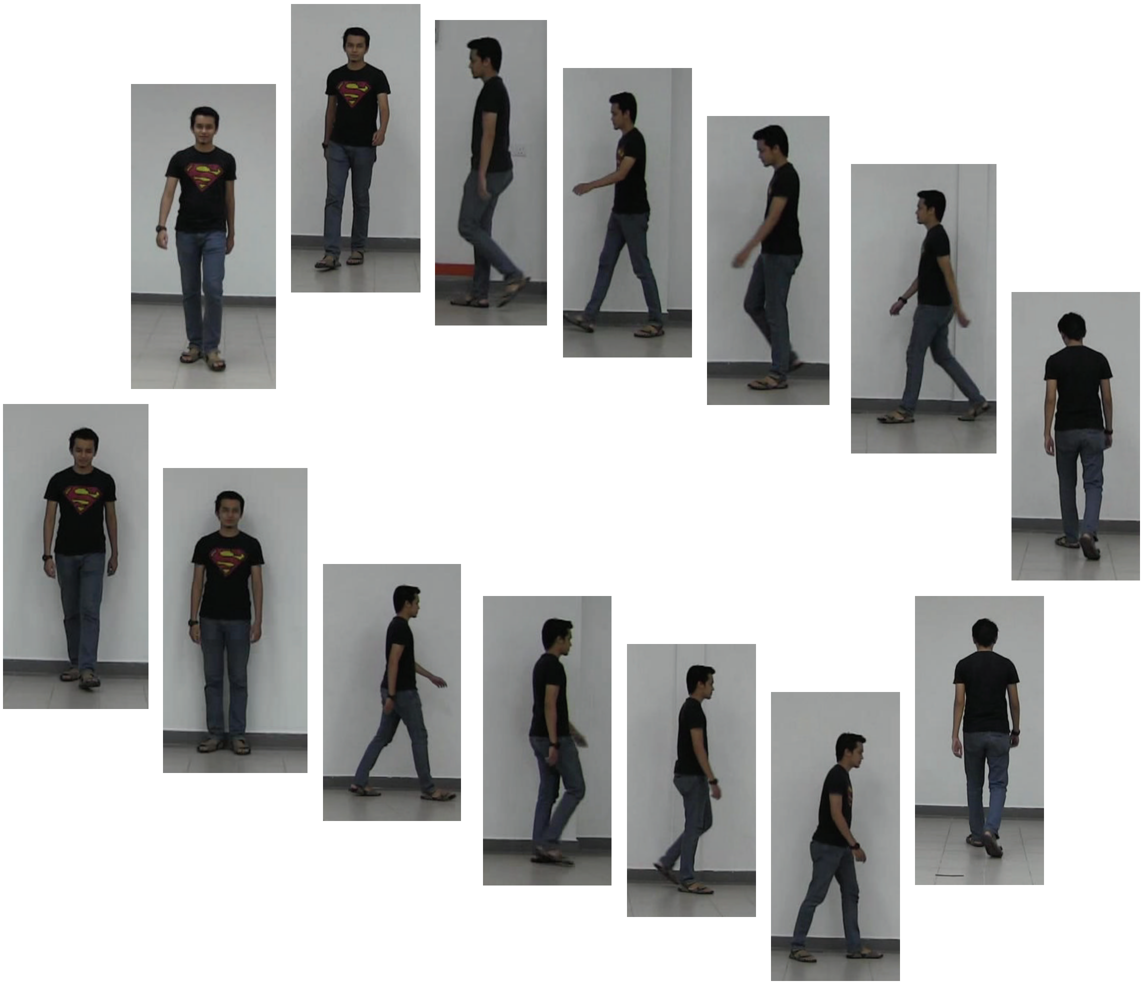


Figure 9: Fourteen frame sets of a subject selected at 1 fps showing four different poses.

representing as low frame rate as possible, in order to provide a large fluctuation in the strides of the subject as shown in Figure 9.

4.1. Height Estimation. In order to evaluate the height estimation using single view metrology, the object extraction is utilized to detect the ROI to retrieve the top and bottom coordinates of the subjects, before computing the height using (3). Based on that, the model attained a mean absolute error (MAE) of 1.57 cm and standard deviation of 3.6 cm for the 80 subjects. This denotes that even though there is large fluctuation in the strides of the subjects due to low frame rate, the errors attained in a particular frame can be well compensated by the accurate prediction in the subsequent frames. Besides, the result is very comparable to the related works on height estimation in video with single view camera, by considering the error attained with respect to the frame rate of the video as shown in Table 1. In addition, the scatter plot of distribution of estimated height against actual height for all 80 subjects in the database is represented in Figure 10 and the results achieved for the first 20 subjects are presented in Table 2.

Table 1: Comparison of results of height estimation in video.

Technique	Frame rate (fps)	Error (cm)	Standard deviation (cm)
BenAbdelkader [32]	20	—	3.5
Nguyen and Hartley [33]	—	1.89	—
Jeges et al. [34]	—	—	4.3
Hansen et al. [35]	15**	3.4	—
Our method	1	1.57	3.6

**Frame rate is presented in Hertz (Hz).

4.2. Weight Estimation. For body weight estimation, the data from session 1 is used for training FFNN, with single hidden layer. The Levenberg-Marquardt training function is used for back propagation. The hidden layer uses the logistic sigmoid (logsig) activation function, while linear (purelin) activation function is utilized in the output layer. During training, the data is randomly divided into 70 and 30 percent as training and validation set, respectively. The criteria for stopping the

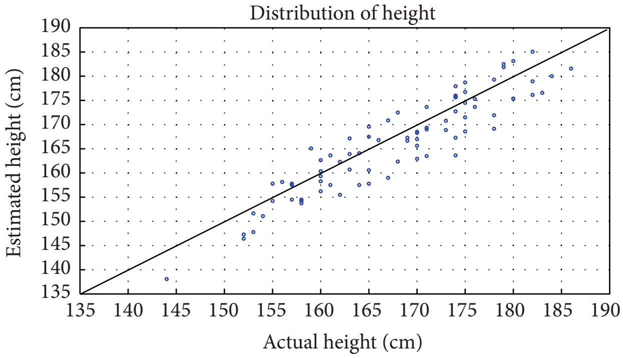


Figure 10: Plot of actual against estimated height by incorporating single view metrology to low video frame rate (1 fps).

training, in order to avoid over fitting, are based on any of the following three conditions accordingly:

(i) in every iteration, if the optimizer is able to reach convergence, once validation is performed, the training should be stopped;

(ii) if the validation error starts to increase for six consecutive attempts, the training should be stopped;

(iii) the training should be stopped if the maximum number of epoch, 100 is attained.

Then, after training is completed, we tested the ability of neural network to learn and adapt from recognized patterns, even in the presence of noise as a result of changes in appearance of the subjects, using a different data acquired in session 2, with a difference of 15 days to the training data.

It is more important to note that, in order to determine the best parameter for the body weight estimation using FFNN, an initial experimentation is performed using different number of nodes from 1 to 40, whereby the experiment on each node is run 10 times, with different seeds and the results of the performance of the nodes are plotted in Figure 11. We present our results in terms of mean absolute error (MAE) and standard deviation. The final result for each node is the average of the results from the 10 runs. The best result for weight estimation for all 80 subjects in the database is MAE of 4.66 kg and standard deviation of 3.48 kg. The result is attained with single hidden layer neural network of 27 nodes.

The scatter plot of distribution of predicted against actual body weight for the whole 80 subjects in the database is shown in Figure 12. Also, in Table 2, is the result for the first 20 subjects in the database.

Velardo and Dugelay [29] have presented a baseline body weight estimation model and confirmed that their system could predict the body weight of 20 static subjects with an error of ±5 to the real weight of the subjects from image, while Labati et al. [30] performed weight estimation of 25 subjects with two calibrated cameras. The authors reported their result with *mean error*, as they attained a mean error of 0.07 kg and standard deviation of 2.3 kg. However, the dataset used for the experiments is not publicly available.

Moreover, since the proposed method in this paper is based on single view camera, therefore, to offer a fair comparison of our approach, the proposed method is benchmarked against the baseline model of Velardo and Dugelay [29], by implementing their feature extraction technique and model on our dataset. The result is highlighted in Table 3. With regard to that, the proposed technique is more robust and outperformed the technique presented in [29] by a significant factor in terms of fluctuation in strides of the walking subjects and changes in clothes appearance. The mean absolute error (MAE) and standard deviation of the two models are highlighted in Table 3.

Some important observations that can be inferred which indicate the advantage of the proposed method over the model in [29] are pointed out as follows.

(i) The proposed technique is implemented on dataset of walking subjects, whereas in [29] frontal and side view image pair of static subjects were used for deriving the body weight model. Therefore, the robustness of their work on data with fluctuations in strides of the subjects is not analyzed, which is evident in the result attained when we implemented the technique on UPM SOFTBIO, as shown in Table 3. In addition, our proposed method performs reliably despite changes in appearance of the subjects.

(ii) The model in [29] requires a very precise and accurate localization of feature points due to the need to measure the geometric length or width between the interest points. Despite the fact that the geometric measures are performed manually, we discovered that an accurate and precise localization can be significantly affected by large clothes of the subjects and self-occlusion caused by arm's swing while the subjects are in motion. But the proposed method in this paper does not necessitate a precise localization of feature points, since it utilizes an automatic procedure of coarse segmentation of the subjects into 3 main body regions.

In order to offer completeness of the proposed technique and ensure generalization of our analysis, an additional experimentation is performed using the optimal number of nodes, 27, for the hidden layer of FFNN to evaluate whether the model can predict the body weight of an unknown subject. For this purpose, session 1 data is randomly divided into equal halves, whereby the first 50% (560 frames) is used for training and the remaining (560 frames) is used for testing. The process is repeated 100 times with different partitions in each run and the final result is the average of 100 runs, presented in Table 4.

Finally, a further experiment is carried out by combining height and weight for a simple human identification at a distance. We used the matcher described in [36] for similarity matching of session 1 and session 2 data. In fact, the results are promising; even though the attributes are not predicted with optimum accuracy, the combination of the two attributes could attain a top rank identification of 51% and a rank 5 identification rate of 93%, while, at rank 10, the identification rate is 100%. Definitely, a reliable identification system cannot be modeled using height and weight alone, but the two

Table 2: Results of height and weight computation of the first 20 subjects in the database.

ID	Real height	Mean absolute error	Standard deviation	Real weight	Mean absolute error	Standard deviation
1	171	2	0	60	2.26	5.13
2	160	3.79	3.49	45	6.64	5.76
3	152	2.71	2.73	53	3.53	5.88
4	157	2.5	3.98	56	4.57	5.68
5	154	2.93	2.79	48	0.96	2.27
6	158	0.29	2.79	65	2.5	2.83
7	175	0.5	2.74	54	4.35	5.22
8	169	2.36	4.01	56	2.97	4.56
9	159	0.07	4.08	60	4.48	3.63
10	174	3.93	3.22	55	5.62	3.36
11	162	1.5	2.07	67	5.33	3.96
12	152	0.57	2.28	47	1.62	1.88
13	165	1.43	3.01	60	1.82	2.16
14	168	1.64	3.95	55	2.84	2.6
15	175	0.43	2.9	63	6.2	2.68
16	180	1.64	4.38	74	6.38	2.84
17	182	3.07	3.63	83	2.79	3.95
18	157	0.79	2.64	55	5.66	3.76
19	163	2.29	2.55	75	8.65	2.96
20	163	0.93	2.06	55	2.39	2.64

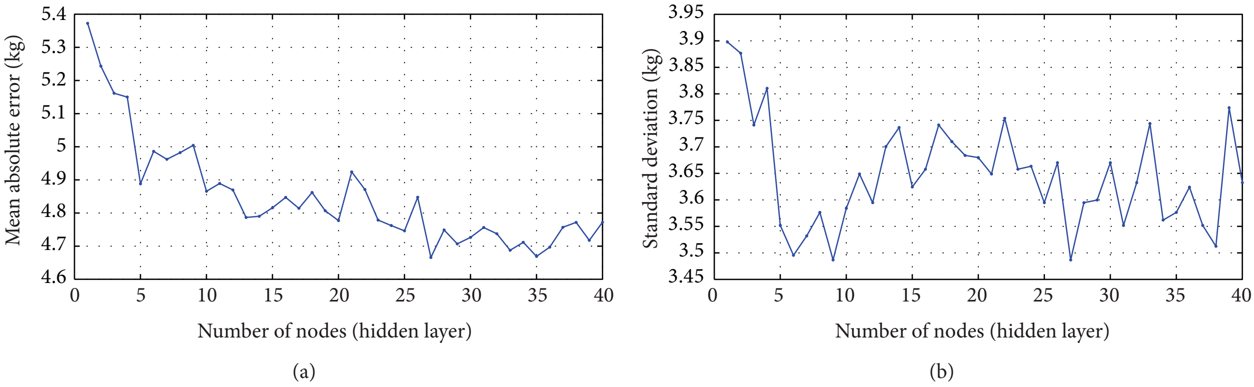


Figure 11: Performance of different nodes. (a) Mean absolute error and (b) standard deviation.

Table 3: Comparison of weight estimation with baseline model.

Technique	Feature extraction	Prediction model	MAE (kg)	Standard deviation (kg)
Velardo and Dugelay [29]	7 body measures	Linear regression	7.21	6.24
Proposed method	13 body measures	FFNN	4.66	3.48

Table 4: Results of weight estimation for unknown subjects.

Number of nodes	Training sample	Testing sample	MAE (kg)	Standard deviation (kg)
27	560 frames	560 frames	6.39	5.1

attributes could serve in improving the performance of other biometrics such as face and gait recognition, either as a prescreening filter or by score fusion.

5. Conclusions

We implemented an approach for weight computation and height estimation from video frames in an unobtrusive manner. We experimented on 1120 frames of 80 subjects in a new dataset, UPM SOFTBIO, compiled in our laboratory which contains 101 subjects, walking in an indoor environment, under uncontrolled lightning conditions describing

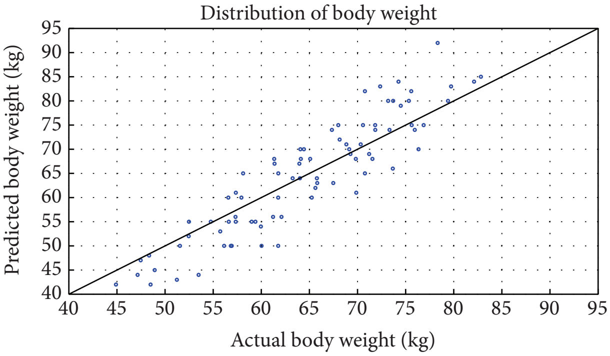


Figure 12: Plot of actual against estimated body weight attained at 27 nodes.

four different directions and varying appearances. The height estimation was based on existing techniques, but we incorporated the technique into video recording with 1 fps, thereby showing significant fluctuation in the walk of the subjects. Moreover, we utilize the camera's real height as the reference height and automatically located the camera position as the distance between the horizontal vanishing line and the feet of the object at any position in the field of view. The result indicates that, under varying strides as a result of low frame rate, the errors attained in a particular frame can be well compensated by the accurate prediction other frames.

For weight computation, several literatures on soft biometrics have suggested the possibility of using body weight as additional biometric information. However, only few literatures can be referred to which have actually gained an insight into this possibility using an image. Therefore, this paper demonstrated a technique for weight computation. We extracted features from the body segments of each individual after silhouette segmentation into head, torso, and leg regions and then used neural network to estimate their respective weights. It is worth noting that even though the result shows the ability of neural network to adapt to significant changes in objects' appearance, based on the MAE of 4.66 kg, we attribute the huge error attained to the effect of clotheses of the subjects. For instance, a subject, whose original body weight is 55 kg could be predicted as 59.66 kg, due to the change of clothes of the subject. Furthermore, the model could predict the body weight of an unknown individual with a MAE of 6.39 kg. This basically shows that the model performs well to a reasonable extent when only single image is available. The limitation of our work is that the experiment has been carried out only in indoor environment, although our target application is directed towards indoor visual surveillance. In future work, we will be incorporating more feature sets to the model and also compile additional datasets in outdoor environments to implement the proposed method.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The authors would like to acknowledge the Malaysian Ministry of Higher Education for the provision of Exploratory Research Grant Schemes through which this research was made possible.

References

[1] A. K. Jain and A. Ross, *Introduction to Biometrics*, 2005.

[2] A. M. Al-Juboori, W. Bu, X. Wu, and Q. Zhao, "Palm vein verification using multiple features and locality preserving projections," *The Scientific World Journal*, vol. 2014, Article ID 246083, 11 pages, 2014.

[3] A. Uçar, "Color face recognition based on steerable pyramid transform and extreme learning machines," *The Scientific World Journal*, vol. 2014, Article ID 628494, 15 pages, 2014.

[4] R. Min, A. Hadid, and J.-L. Dugelay, "Efficient detection of occlusion prior to robust face recognition," *The Scientific World Journal*, vol. 2014, Article ID 519158, 10 pages, 2014.

[5] J. Daugman, "How iris recognition works," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 21–30, 2004.

[6] A. K. Jain, S. Prabhakar, and S. Pankanti, "On the similarity of identical twin fingerprints," *Pattern Recognition*, vol. 35, no. 11, pp. 2653–2663, 2002.

[7] A. K. Jain, S. C. Dass, and K. Nandakumar, "Soft biometric traits for personal recognition systems," in *Biometric Authentication: Proceedings of the 1st International Conference, ICBA 2004, Hong Kong, China, July 15–17, 2004*, vol. 3072 of *Lecture Notes in Computer Science*, pp. 731–738, 2004.

[8] O. A. Arigbabu, S. M. S. Ahmad, W. A. W. Adnan, and S. Yussof, "Recent advances in facial soft biometrics," *The Visual Computer*, 2014.

[9] A. K. Jain, S. C. Dass, and K. Nandakumar, "Can soft biometric traits assist user recognition?" in *Biometric Technology for Human Identification*, Proceedings of SPIE, pp. 561–572, usa, April 2004.

[10] A. Dantcheva, C. Velardo, A. D'Angelo, and J. Dugelay, "Bag of soft biometrics for person identification: new trends and challenges," *Multimedia Tools and Applications*, vol. 51, no. 2, pp. 739–777, 2011.

[11] A. Dantcheva, J.-L. Dugelay, and P. Elia, "Person recognition using a bag of facial biometrics (BoFSB)," in *Proceedings of the IEEE International Workshop on Multimedia Signal Processing (MMSP '10)*, pp. 511–516, Saint Malo, France, October 2010.

[12] K. Niinuma, U. Park, and A. K. Jain, "Soft biometric traits for continuous user authentication," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 4, pp. 771–780, 2010.

[13] A. Dantcheva and J. Dugelay, "Frontal-to-side face re-identification based on hair, skin and clothes patches," in *Proceedings of the 8th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS '11)*, pp. 309–313, Klagenfurt, Austria, September 2011.

[14] S. Denman, C. Fookes, A. Bialkowski, and S. Sridharan, "Soft-biometrics: unconstrained authentication in a surveillance environment," in *Proceedings of the Digital Image Computing: Techniques and Applications (DICTA '09)*, pp. 196–203, Melbourne, VIC, Australia, December 2009.

[15] C. Velardo, C. Araimo, and J. Dugelay, "Synthetic and privacy-preserving visualization of video sensor network outputs," in *Proceedings of the 5th ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC '11)*, pp. 1–5, Ghent, Belgium, August 2011.

[16] J. Li, X. Zhen, X. Liu, and G. Ouyang, "Classifying normal and abnormal status based on video recordings of epileptic patients," *The Scientific World Journal*, vol. 2014, Article ID 459636, 6 pages, 2014.

[17] H. Ailisto, M. Lindholm, S. Mäkelä, and E. Vildjiounaite, "Unobtrusive user identification with light biometrics," in *Proceedings of the 3rd Nordic Conference on Human-Computer Interaction (NordiCHI '04)*, pp. 327–330, October 2004.

[18] S. Yu, D. Tan, and T. Tan, "A framework for evaluating the effect of view angle, clothing and carrying condition on gait recognition," in *Proceedings of the 18th International Conference on Pattern Recognition (ICPR '06)*, vol. 4, pp. 441–444, Hong Kong, August 2006.

[19] J. M. Ferryman, "Performance evaluation of tracking and surveillance," in *Proceedings of the 9th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS '06)*, 2006.

[20] D. Gray and H. Tao, "Viewpoint invariant pedestrian recognition with an ensemble of localized features," in *Proceedings of the 10th European Conference on Computer Vision*, pp. 262–275, 2008.

[21] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[22] P. Kelly, C.Ó. Conaire, and D. Monaghan, "Performance analysis and visualisation in tennis using a low-cost camera network," in *Multimedia Grand Challenge Track at ACM Multimedia*, pp. 1–4, 2010.

[23] Z. Wang, S. Yoon, S. J. Xie, Y. Lu, and D. S. Park, "A high accuracy pedestrian detection system combining a cascade AdaBoost detector and random vector functional-link net," *The Scientific World Journal*, vol. 2014, Article ID 105089, 7 pages, 2014.

[24] J. Li and N. M. Allinson, "A comprehensive review of current local features for computer vision," *Neurocomputing*, vol. 71, no. 10-12, pp. 1771–1787, 2008.

[25] K.-Z. Lee, "A simple calibration approach to single view height estimation," in *Proceedings of the 9th Conference on Computer and Robot Vision (CRV '12)*, pp. 161–166, Toronto, Canada, May 2012.

[26] F. Lv, T. Zhao, and R. Nevatia, "Camera calibration from video of a walking human," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 9, pp. 1513–1518, 2006.

[27] A. Criminisi, I. Reid, and A. Zisserman, "Single view metrology," *International Journal of Computer Vision*, vol. 40, no. 2, pp. 123–148, 2000.

[28] D. Reid, M. Nixon, and S. Stevenage, "Soft biometrics; human identification using comparative descriptions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 6, pp. 1216–1228, 2013.

[29] C. Velardo and J. Dugelay, "Weight estimation from visual body appearance," in *Proceeding of the 4th IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS '10)*, pp. 1–6, Washington, DC, USA, September 2010.

[30] R. D. Labati, A. Genovese, V. Piuri, and F. Scotti, "Weight estimation from frame sequences using computational intelligence techniques," in *Proceedings of the IEEE International Conference on Computational Intelligence for Measurement Systems and Applications (CIMSA '12)*, pp. 29–34, Tianjin. China, July 2012.

[31] S. Aksoy and R. M. Haralick, "Feature normalization and likelihood-based similarity measures for image retrieval," *Pattern Recognition Letters*, vol. 22, no. 5, pp. 563–582, 2001.

[32] C. BenAbdelkader, "Person identification using automatic height and stride estimation," in *Proceedings of the 16th International Conference on Pattern Recognition*, pp. 377–380, 2002.

[33] N. H. Nguyen and R. Hartley, "Height measurement for humans in motion using a camera: a comparison of different methods," in *Proceeding of the 14th International Conference on Digital Image Computing Techniques and Applications (DICTA '12)*, pp. 1–8, Fremantle, Australia, December 2012.

[34] E. Jeges, I. Kispal, and Z. Hornak, "Measuring human height using calibrated cameras," in *Proceedings of the Conference on Human System Interactions*, pp. 755–760, 2008.

[35] D. M. Hansen, B. K. Mortensen, P. T. Duizer, J. R. Andersen, and T. B. Moeslund, "Automatic annotation of humans in surveillance video," in *Proceedings of the 4th Canadian Conference on Computer and Robot Vision (CRV '07)*, pp. 473–480, May 2007.

[36] H. Ailisto, E. Vildjiounaite, M. Lindholm, S. Mäkelä, and J. Peltola, "Soft biometrics-combining body weight and fat measurements with fingerprint biometrics," *Pattern Recognition Letters*, vol. 27, no. 5, pp. 325–334, 2006.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 632575, 12 pages
http://dx.doi.org/10.1155/2014/632575

Research Article

Anomaly Detection Based on Local Nearest Neighbor Distance Descriptor in Crowded Scenes

Xing Hu, Shiqiang Hu, Xiaoyu Zhang, Huanlong Zhang, and Lingkun Luo

School of Aeronautics and Astronautics, Shanghai Jiao Tong University, Dongchuan Road, No. 800, Shanghai, China

Correspondence should be addressed to Shiqiang Hu; sqhu@sjtu.edu.cn

Received 11 April 2014; Accepted 10 June 2014; Published 3 July 2014

Academic Editor: Yu-Bo Yuan

We propose a novel local nearest neighbor distance (LNND) descriptor for anomaly detection in crowded scenes. Comparing with the commonly used low-level feature descriptors in previous works, LNND descriptor has two major advantages. First, LNND descriptor efficiently incorporates spatial and temporal contextual information around the video event that is important for detecting anomalous interaction among multiple events, while most existing feature descriptors only contain the information of single event. Second, LNND descriptor is a compact representation and its dimensionality is typically much lower than the low-level feature descriptor. Therefore, not only the computation time and storage requirement can be accordingly saved by using LNND descriptor for the anomaly detection method with offline training fashion, but also the negative aspects caused by using high-dimensional feature descriptor can be avoided. We validate the effectiveness of LNND descriptor by conducting extensive experiments on different benchmark datasets. Experimental results show the promising performance of LNND-based method against the state-of-the-art methods. It is worthwhile to notice that the LNND-based approach requires less intermediate processing steps without any subsequent processing such as smoothing but achieves comparable event better performance.

1. Introduction

Due to the fact that anomaly is a potentially hazardous source in crowded scenes, anomaly detection in video surveillance is an important task for public security and safety and attracts more and more researchers' attentions recently. It is also a challenge task, because it requires inspecting an excessive number of pedestrians or moving objects and their activities and overcomes some difficult problems such as frequent occlusions, illumination change, noisy, and deformation. The primary task of anomaly detection is to detect the event or interaction that deviated from the expected [1]. Figure 1 shows some examples of video anomalies detection in crowded scenes.

Anomalies can be classified into single anomalous event and anomalous interaction involving multiple events. Single anomalous event is defined by native information of the event such as anomalous speed, direction, and appearance. Anomalous interaction is defined by spatial context around the event with respect to another events occurring at the same time. There is no doubt that the interaction between anomalous event and the other event is anomalous, even between two normal events may be anomalous. For example, one man appears in front of running car. In most previous works, video event is typically defined by native information within a predefined spatiotemporal region and characterized by low-level feature descriptors. The commonly used low-level feature descriptors include optical flow-based [2–11], gradient-based [7, 12, 13], dynamic texture-based [14–17], and frequency-based [18, 19] descriptors. These feature descriptors have been proven that can well characterize the video event in crowded scenes. These feature descriptors have two things in common. First, they only contain the native information of the event itself without respect to the contextual information around the event. Second, they are all high-dimensional descriptors.

In video scenes analysis, contextual information refers to the spatiotemporal relationships between it and other events or its located surroundings [23], which consist of spatial context and temporal context. Spatial context is defined as the relationships with respect to its located surroundings or nearby events occurring at the same time. Temporal

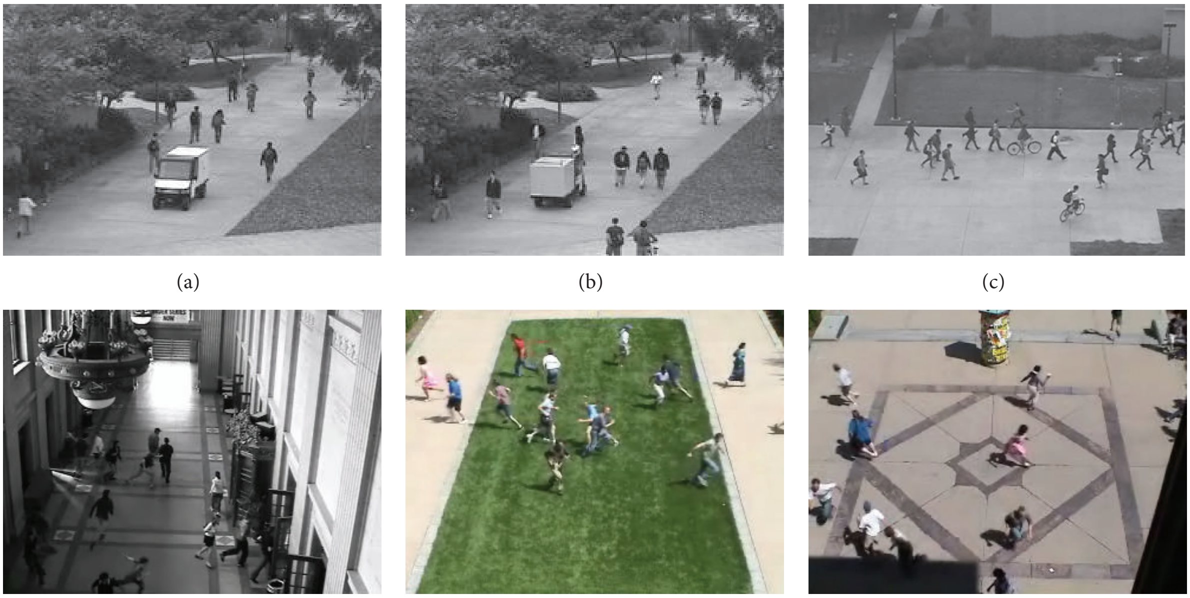


Figure 1: Some examples of abnormal event in crowded scenes.

context is defined as the relationships with respect to the history of the event in the past. An event which is normal only considering the temporal context may be perceived as highly anomalous when it cooccurs with another event in a certain region and period of time or locates in a certain surrounding. Hence, accounting for spatial context of the event is very important for detecting anomalous interaction. However, most feature descriptors in the previous works do not incorporate the contextual information. In order to detect anomalous interaction, they often resort to training a spatiotemporal model for learning contextual information, such as in [3, 4, 6, 8, 10, 12], spatiotemporal Markov random filed (ST-MRF) model, spatiotemporal conditional random filed (ST-CRF) model, spatial saliency detector, and cascade topic model, which were applied for learning the contextual information. Although the contextual information can be well learned and inferred by the spatiotemporal model, not only are the appropriate model constructing and parameters setting not easy, but also the Bayesian inference of these models is often computationally expensive. Only a few works [5, 9, 11] that the authors developed multiscale histogram of optical flow (MHOF) as descriptor to incorporate the contextual information. However, due to the fact that this descriptor was constructed by simply assembling the event with its six neighbors, the relationships between two events are not reflected, and its dimensionality is high, such as 112 dimension in [5, 9] and 168 dimension in [11].

Due to the richer information content contained in the high-dimensional feature descriptor, video events are typically characterized by high-dimensional feature descriptor for adapting its diversity and complexity. In [3–5, 9–11, 13–17], the proposed or adopted feature descriptors are uniformly high-dimensional. Using high-dimensional feature descriptor inevitably suffers from its inherent limitations. First, high-dimensional feature descriptor contains much redundant information and noise that would degrade the performance of detection model. Second, training detection model using high-dimensional descriptors is prone to overfitting and curse of dimensionality. Third, for the detection methods with offline fashion, such as in [2, 4, 8, 10, 12, 17, 19], large amounts of memory are required for storing and computation time for training. Although many dimensionality reduction methods can alleviate these limitations, such as linear method, principle component analysis (PCA), and nonlinear methods including manifold learning techniques, such as Laplacian Eigenmap (LE), additional computation cost is increased and would suffer from information loss problem. The lower the dimension is, the more the information is lost.

In this paper, we propose a novel LNND descriptor to represent video event for detecting abnormal event in crowded scenes. First, the contextual information is incorporated into LNND descriptor, so both anomalous single event and anomalous interaction of multiple events can be detected without additional learning contextual information by spatiotemporal model. Second, LNND is a concise and compact descriptor that its dimensionality is much lower than that of low-level feature descriptor, because it is constructed by considering only a few spatial and temporal neighbor events. Accordingly, the memory requirement for storage and computation time for training detector can be saved. In order to tackle feature's noise and uncertainty which is inevitable in crowded scenes, EMD [24] is adopted as distance measure between two events, which is a well-known robust metric in case of noisy histogram comparison. To deal with the computation expensive problem of original EMD, we introduce WEMD to replace the original EMD as distance measure to significantly degrade the computation complexity.

The main contributions of our work are as follows. (1) We propose a simple yet efficient LNND descriptor to represent video event for anomaly detection in crowded scenes. By using LNND descriptor, contextual information can be accounted for, so both anomalous event and interaction can be efficiently detected with less intermediate process. (2) Due to the fact that dimensionality of LNND descriptor is much lower than typically used low-level feature descriptor, both memory requirement and computation time can be accordingly saved. To our best knowledge, this is the first attempt to represent the video event by local nearest neighbors distance. We use the very concise descriptor and yet achieve the performance that can be comparable with the state-of-the-art methods; therefore, our idea is in accordance with the rule of Occam's razor [25].

The rest of this paper is organized as follows. Section 2 overviews the related works. In Section 3, we introduce the details of the construction of LNND descriptor. In Section 4, we describe the anomaly detection method using LNND descriptor. The experimental results and evaluations are given in Section 5, and the conclusion is given in Section 6.

2. Related Work

Anomaly detection in video surveillance is a hot topic and attracts more and more researchers' interest. Meanwhile, there is a challenge task due to many difficult problems, such as inevitable noise illumination change and deformation in the scenes, diversity of event, and interaction between multiple events. To detect anomaly in crowded scenes, different methods have been proposed to overcome one or more specific problems. These methods can be categorized into two classes according to the used feature descriptor: one is the tracking-based methods and the other is the nontracking-based method.

For tracking-based methods [26–31], the pedestrians or moving objects are firstly detected by frames difference, background subtraction, and so on. Then the trajectories of them are obtained by tracking algorithm. The normalcy model is learned using the obtained normal trajectories, and the trajectories from testing video deviating from the normalcy model are labeled as anomalous behaviors. Although there are many advantages to use trajectory as feature, tracking algorithm tends to fail in crowded scenes due to large number of individuals and frequent occlusions. Hence, the tracking-based method is suitable to be applied in noncrowded scenes. Moreover, tracking-based method cannot be able to deal with the anomalies in temporal.

To avoid the tracking problem in crowded scenes, many nontracking-based methods have been proposed. In these methods, the used feature descriptors, such as optical flow, gradient, and texture-based feature descriptors, were extracted from local 2D region, 3D clip, or local cuboids [2, 5, 9, 12, 32–34]. Those methods are not relying on objects detection and tracking algorithm. Our method belongs to this class, but our method can also be applied in noncrowded scenes. Kratz and Nishino [12] modeled the 3D gradients which were extracted from local spatiotemporal cuboid by using 3D Gaussian model for obtaining the prototypes events, and then a coupled distribution-based hidden Markov model (HMM) was used to detect anomalous events in extremely dense crowded scenes. Mahadevan et al. [4] modeled the normal crowd behavior by mixture of dynamic texture (MDT) models which can capture the dynamic of both motion and appearance. The temporal anomalies are detected using background behavior subtraction, and the spatial anomalies are detected using spatial discriminative saliency detector. The final detection result was obtained by combining two results from both temporally and spatially. In [35], a Neyman-Pearson-based probabilistic framework was proposed to detect rare pattern with respect to their neighbors. Kim and Grauman [3] modeled local optical flow with a mixture of probabilistic PCA models and enforced the consistency by Markov random fields (MRF). Antić and Ommer [36] parsed video frames by establishing a set of hypotheses that jointly explain all the samples that explain the hypotheses. Bertini et al. [13] used a nonparametric model to detect abnormal event in each local region, where the event was characterized by histogram of spatiotemporal gradient. In order to cope with the gradual change in the scenes, some online anomaly detection methods were proposed for preventing concept drifty. Zhao et al. [7] proposed a fully unsupervised method to detect abnormal event in video surveillance. An overcomplete dictionary is learned and updated online. The events with high reconstruction cost under the learned dictionary were classified as abnormal events. Roshtkhari and Levine [23, 37] proposed a probabilistic framework for online learning dominant behaviors and detecting anomalous behaviors in crowded scenes. Crowd behavior was represented as a spatiotemporal composition of video volumes, and anomalous behavior was detected as video volumes arrangement with very low frequency of occurrence. Also some methods were presented for detecting only global anomaly in the scenes, namely, only locating the temporal position of anomalous event. Mehran et al. [2] measured interaction force between individuals using social force model for each video clip and then the normal force flow was represented as bag-of-word and was trained by latent Dirichlet allocation (LDA); the query video clips with low probability under the trained LDA were labeled as anomalous; the anomalous region was localized as the location with maximum force. Cui et al. [38] proposed a method that represented a subject by its action and behavior state, where the action was reflected by its velocity and the behavior state was reflected by its interaction energy potential based on the linear trajectory avoidance (LTA) method. Finally, linear SVM was used to detect abnormal events. Raghavendra et al. [22] proposed a robust method for optimizing the interaction force computed using social force model by particle swarm optimization (PSO). In [39], the directions and displacements of interesting points are calculated for each video clip, and the anomalous behaviors of crowd are detected as the clips with high entropy values. Some methods were proposed for detecting both local and global abnormal events, such as in [5, 9, 11]; three types of descriptors based on MHOF were proposed for detecting both local and global anomalies. Our method can detect both local and global anomalous events.

3. Local Nearest Neighbor Distance Descriptor

In this section, we describe the detail of the construction of LNND descriptor. Given a video sequence V, we divide it into a set of spatiotemporal cuboids $\{\mathbf{V}_{s,t}\}$, where s and t are the locations of the cuboid in spatial and temporal, respectively. Each $\mathbf{V}_{s,t}$ is considered as an event, and all of them have uniform size of $h \times w \times \tau$ and partially overlapped with its neighbor cuboids. Let $X_{s,t}$ denote the low-level feature descriptor extracted from $\mathbf{V}_{s,t}$. We compute the distance between the local event $\mathbf{V}_{s,t}$ and each of its neighbors $\mathbf{V}_{s',t'}$; that is, compute $d(X_{s,t}, X_{s',t'})$. In order to cope with the inevitable noise and uncertainty in the low-level feature descriptor, we adopt EMD as distance measure. Next, we start by introducing the low-level feature extraction.

3.1. Low-Level Visual Feature Descriptor. Generally speaking, most pervious used low-level features can be served for our purpose, such as multiscale histogram of optical flow (MHOF) [5, 9, 11], histogram of spatiotemporal gradient descriptor [13], and LBP-TOP [14]. In this paper, we adopt local motion pattern (LMP) as a feature descriptor [20], due to the fact that it is distinctive, scale invariant, and fast to compute. Different from the LMP descriptor in [20], our LMP descriptor is computed for spatiotemporal gradient magnitude of each pixel rather than for raw pixel value. Thus, the motion and appearance dynamic of crowd can be well characterized by gradient-based LMP descriptor. Given a spatiotemporal cuboid $\mathbf{V} \in \mathbb{R}^{h\times w\times \tau}$ obtained by dividing video sequence, we compute spatiotemporal gradient magnitude for each pixel. Then, the 2nd (variance, M_2), 3rd (skewness, M_3), and 4th (kurtosis, M_4) central moments are computed for each spatial pixel location along the temporal direction, which reflect three important statistical properties, that is, variance, skewness, and kurtosis, of the temporal change of the pixel spatiotemporal gradient magnitude, respectively. We define the moment matrix M_r, $r = \{2,3,4\}$ associated with $\mathbf{V}$ as follows:

$$M_r = \left[m_{i,j}\right], \quad i = 1,2,\dots,h, \;\; j = 1,2,\dots,w, \tag{1}$$

where

$$m_{i,j} = \frac{1}{\tau}\sum_{t=1}^{\tau}\left(v_{ijt}\right)^r. \tag{2}$$

Here, v_{ijt} is the spatiotemporal gradient magnitude value of the pixel at location $\{i, j\}$ of the tth patch. Each moment matrix M_r, $r = \{2,3,4\}$ is transformed to a vector $m_r \in \mathbb{R}^{hw}$. The three moment vectors corresponding to three values of r are concatenated on top of each other to form a single vector $m \in \mathbb{R}^D$, where $D = 3hw$:

$$M = \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix}. \tag{3}$$

The vector M is the computed LMP descriptor. After extracting the low-level feature descriptor for each event, we will present the distance measure between two events in the next subsection.

3.2. Earth Mover's Distance. The Earth Mover's Distance (EMD) [40] is a distance measure between two signatures or histograms, which is robust to geometric deformation, illumination change, and heavy noise. Let $p, q \in \mathbb{R}^D$ be two histograms and are all normalized to unit mass. The EMD is obtained as the solution of the transportation problem:

$$\min_{f_{i,j}\geq 0}\sum_{i=1}^{D}\sum_{j=1}^{D} g_{i,j} f_{i,j}, \quad \text{s.t.} \; \sum_{i=1}^{D} f_{i,j} \leq p_j, \;\; \sum_{j=1}^{D} f_{i,j} \leq q_i, \tag{4}$$

where $f_{i,j}$ denotes the flow between b_i and q_i and $g_{i,j}$ denotes the ground distance between i and j. This problem can be solved by considering it as a linear dynamic programming problem. However, in the case of high-dimensional histograms, solving (4) can be very time consuming due to the number of flow variables involved. For D-dimensional histogram, the computational complexity is $O(D^3 \log D)$. Many efforts had been devoted to reduce the complexity and speed up the calculation of EMD. In [24], a fast EMD-L_1 algorithm was proposed. In EMD-L_1 algorithm, L_1 distance is adopted as ground distance to replace the L_2 distance in original EMD. Consequently, the number of unknown variables in the optimization problem is reduced from $O(D^2)$ to $O(D)$. Accordingly, the time complexity is also reduced from $O(D^3 \log D)$ of original EMD to $O(D^2)$ of EMD-L_1. In [41], a threshold ground distance was adopted in EMD computation. The algorithm transformed the flow network of the EMD so that the number of edges is reduced by an order of magnitude. In our paper, we adopt wavelet EMD (WEMD) to calculate the distance between two events, which is approximation of original EMD proposed in [42]. The wavelet decomposition is applied on the dual program of EMD and the parameters on a small wave are eliminated. The distance between two histograms is well approximated by

$$\text{WEMD}\left(p,q\right) = \sum_{\beta}\alpha_\beta \left|\text{WAV}_\beta\left(p-q\right)\right|, \tag{5}$$

where $\text{WAV}_\beta(b-p)$ are the wavelet transform coefficients of the dimensional difference $p - q$ for all shifts and scales β and $\alpha_\beta = 2^{-2*\beta}$ are the scale characterized by the choice of different scale weighting and different wavelet kernels. The new distance can be efficiently calculated in linear time with respect to the number of bins in the histograms, while the comparison is about as fast as for normal Euclidean distance or χ^2 statistic. In our work, the distance between two events is calculated as follows:

$$d\left(\mathbf{V}_{s,t}, \mathbf{V}_{s',t'}\right) = \sum_{\beta}\alpha_\beta \left|\text{WAV}_\beta\left(\left|M_{s,t}\right| - \left|M_{s',t'}\right|\right)\right|. \tag{6}$$

3.3. The Construction of LNND Descriptor. In video surveillance, most anomalies can be caused by anomalous event

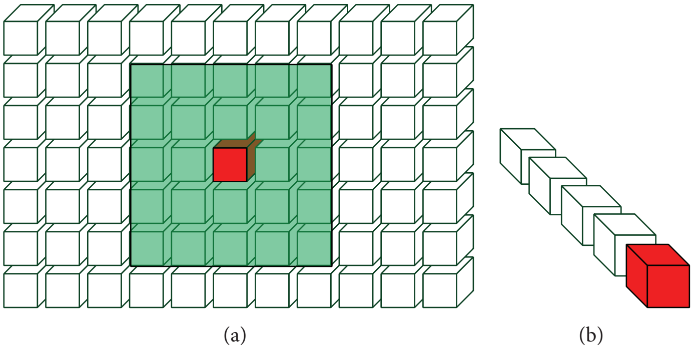


Figure 2: Illustrating the (a) spatial and (b) temporal neighbor sets of given event (red cuboid).

itself or anomalous interactions between multiple events. Anomalous events can be detected by modeling only temporal context using past observations of the event, while anomalous interaction detection needs to account for the spatial context of the event. Given a video event $\mathbf{V}_{s,t}$ and its spatial neighbors set $SN = \{\mathbf{V}_{s',t}\}$ and temporal neighbors set $TN = \{\mathbf{V}_{s,t'}\}$ (see Figure 2), we search K spatial nearest neighbors of the event $\mathbf{V}_{s,t}$ from its spatial neighbors set SN and then define a distance vector X^{sd} using the K distance values to account for the spatial context of the event, given by

$$X^{sd} = [d_1, d_2, \ldots, d_k, \ldots, d_K]^T, \quad (7)$$

where d_k is distance between the event and its kth nearest neighbor, so the K distance values are sorted in an ascending order. The WEMD is adapted as distance measure for reducing the influence of noise and uncertainty in the low-level feature descriptor. The K nearest spatial neighbors are searched in a certain region around the event. We restrict the search range for two reasons: first, anomalous interaction is more relevant to nearby events or located local surroundings of the event; second, the search time can be reduced by restricting the search range. We restrict the spatial search region in a rectangular region which centered at the local event $\mathbf{V}_{s,t}$ with the height and width 5 times larger than local event $\mathbf{V}_{s,t}$ (see Figure 2(a)).

Due to the fact that the interaction between anomalous event and any of its neighbors is abnormal, the anomalous event is basically detected by learning the temporal statistical of spatial distances of training samples. However, in some special cases, such as anomaly occurring in global, anomaly may be missed due to only using spatial distance. Figure 3 illustrates a special example, given three normal training samples and one testing sample containing anomalous event. Compared with the training samples, the spatial distance of testing sample is not changed, so anomaly will not be detected. Although this case is very unusual, we should avoid missing when the case occurred. Consequently, we exploit the temporal context of the event to reflect its temporal variation. We search N temporal nearest neighbors from temporal neighbor set TN and define a distances vector X^{td} as follows:

$$X^{td} = [d_1, d_2, \ldots, d_n, \ldots, d_N]^T. \quad (8)$$

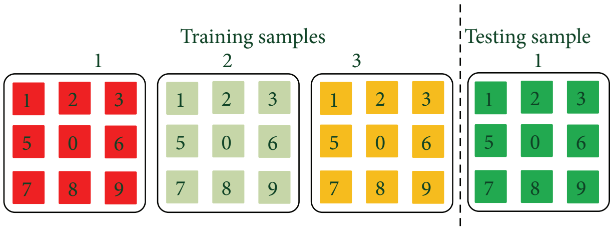


Figure 3: Illustrating the special case, where 1, 2, and 3 are training samples, and testing sample. The colored cuboids refer to the feature of event. The green cuboid refers to anomalous event. We assume that the distance is equivalent between the cuboids with the same color.

Table 1: The dimensions of different descriptors extracted from UCSD Ped1 subset.

Descriptor	Size of cuboids	Dimension
MHOF [9]	10 × 10 × 5	102
HSTG [13]		96
TOP-LBP [14]		768
MPCA [3]		54
LMP [20]		300
LNND		9

The temporal search range is usually restricted in 4 to 8 temporal neighbors. The final LNND descriptor is constructed by concatenating X^{sd} and X^{td} as follows:

$$X = \begin{bmatrix} X^{sd} \\ X^{td} \end{bmatrix}. \quad (9)$$

The dimension of the proposed LNND descriptor is $Q = K + N$. In this work, we select 8 spatial and 1 temporal nearest neighbors, so the dimension of LNND is 9. Table 1 lists the dimension of LNND and other commonly used low-level feature descriptors. We can see from Table 1 that the dimension of LNND descriptor is much lower than that of other low-lever feature descriptors. Consequently, both the storage requirement and the computation time are significantly saved for offline training methods.

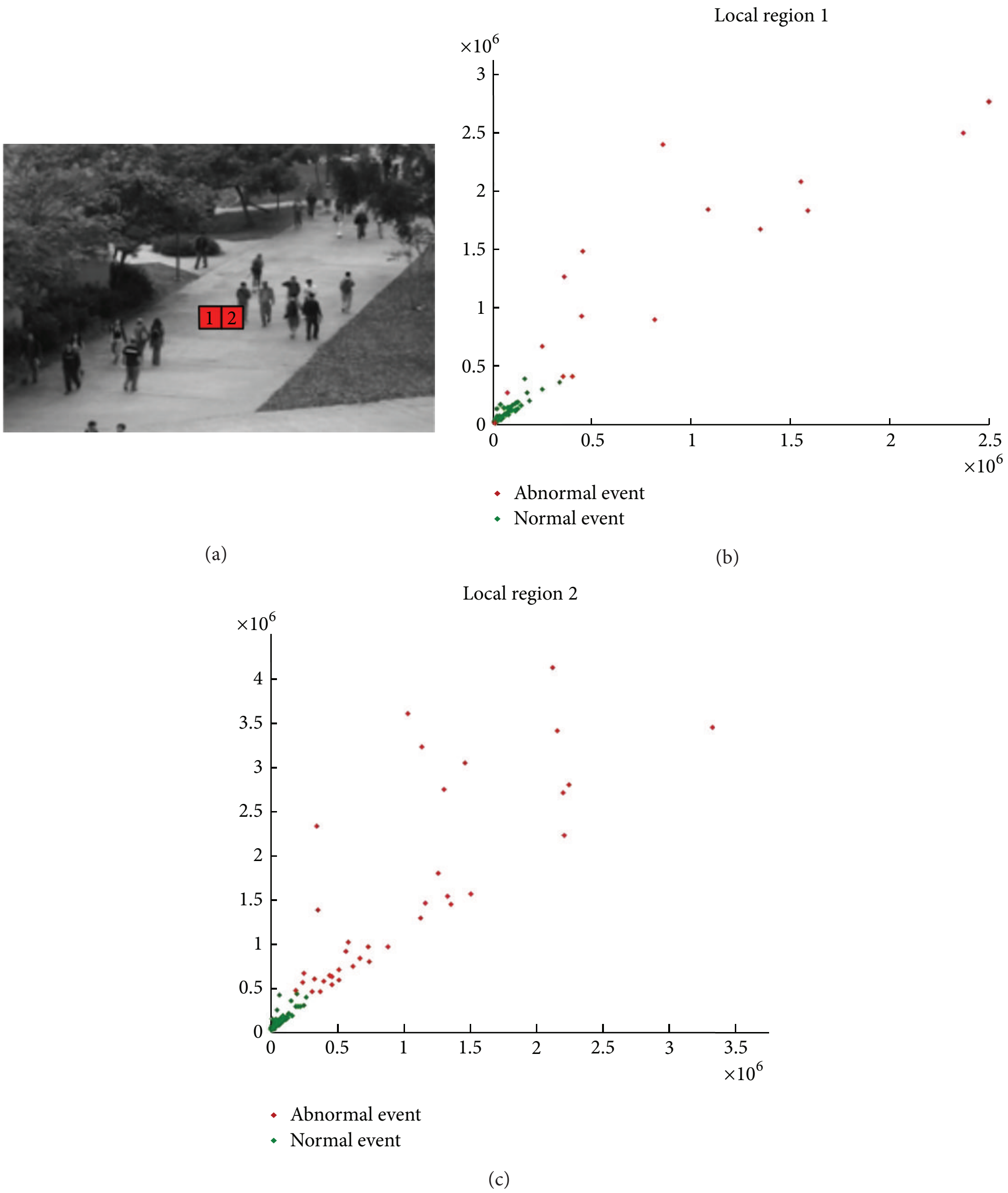


Figure 4: (a) Two local regions selected in the scene; (b) 2D scatter plot of LNND descriptor from local region 1; (c) 2D scatter plot of LNND descriptor from local region 2.

The LNND descriptor provides a good discrimination between normal and anomalous events. We demonstrate the properties by two examples shown in Figure 4; given UCSD (http://www.svcl.ucsd.edu/projects/anomaly/dataset.html) Ped1 sequences, we choose two local regions in the scene, and we plot the LNND descriptors (where $K = 1$ and $N = 1$) of all events within the local regions in a 2D space. We can see from it that most points corresponding to normal events formed a compact cluster, and the points corresponding to anomalous events are far from this cluster and scattered randomly.

4. Anomaly Detection Using LNND Descriptors

In video surveillance application, anomaly detection refers to finding rare or suspicious events from scenes, so it can be formulated as an outlier problem that finds the pattern which deviates from the expected patterns [43]. The expected patterns typically are learned from previous observed normal samples. Many popular modeling techniques have been used for profiling the normal patterns, such as, support vector machine [38], dictionary learning [5, 7, 9], nonnegative

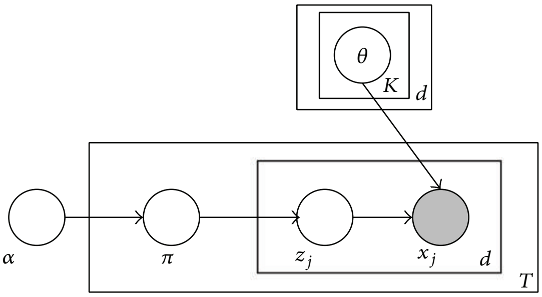


Figure 5: Graphical model representation of MMNB models.

matrix factorization (NMF) [11], graph-model [3, 6, 10], K-NN model [42], and topic model-based methods [2, 8, 19]. In order to account for contextual information, most works modify these techniques or develop additional processes. Due to the fact that our proposed LNND descriptor has incorporated the spatial and temporal context, we can directly detect both anomalous event and interaction by using the temporal model without training spatiotemporal model to account for spatial context, so the intermediate processes are less than most previous works. In our work, we train a fast-mixed membership naive Bayes (Fast MMNB) [44] model in each region. Of course, other popular methods also can be used in our method. We adopt Fast MMNB to profile normal event and detect anomalous event for two reasons: first, the computational cost of Fast MMNB is low and can further improve the real-time performance of anomaly detection system; second, MMNB can deal with large scale dataset with any data type due to the fact that it is designed by taking the advantages of both latent Dirichlet allocation (LDA) and naive Bayes (NB).

The generative process for X following MMNB can be described as follows [44] (see Figure 5).

(1) Choose a mixed-membership vector $\pi \sim$ Dirichlet (α).

(2) For each feature x_j of X,

  (a) choose a component $z_j = c \sim \text{discrete}(\pi)$;

  (b) choose a feature value $x_j \sim p_{\psi_j}(x_j \mid \theta_{jc})$, where ψ_j and θ_{jc} jointly decide an exponential family distribution for feature j and component c. We define $\Theta = \{\theta_{jc}, [j]_1^Q, [c]_1^C\}$.

The density function for X under the generative model is given by

$$p(X \mid \alpha, \Theta) = \int_{\pi} p(\pi \mid \alpha) \left(\prod_{j=1}^{Q} \prod_{c=1}^{C} p(z_j = c \mid \pi) \, p_{\psi_j}(x_j \mid \theta_{jc}) \right) d\pi. \quad (10)$$

And the probability of the whole dataset $\mathbf{X} = [X_1, X_2, \ldots, X_L]$ is given by

$$p(\mathbf{X} \mid \alpha, \Theta) = \prod_{i=1}^{L} \int_{\pi_i} p(\pi \mid \alpha) \times \left(\prod_{j=1}^{Q} \prod_{c=1}^{C} p(z_{ij} = c \mid \pi_i) \, p_{\psi_j}(x_{ij} \mid \theta_{jc}) \right) d\pi_i. \quad (11)$$

For MMNB-Gaussian model, the distributions Θ are defined as a set of Gaussian distributions $\Omega = \{(\mu_{jc}, \sigma_{jc}), [j]_1^Q, [c]_1^C\}$, where μ_{jc} and σ_{jc} are the mean and variance of cth component of jth Gaussian, respectively.

Given a set of training sets $\mathbf{X} = [X_1, X_2, \ldots, X_L]$, the optimal parameters α^* and Ω^* of MMNB-Gaussian model can be learned by maximizing the likelihood of the whole dataset $p(\mathbf{X} \mid \alpha, \Omega)$, given by

$$(\alpha^*, \Omega^*) = \underset{(\alpha,\Omega)}{\arg\max}\ p(\mathbf{X} \mid \alpha, \Omega). \quad (12)$$

A fast variational EM algorithm is proposed for learning the optimal parameters and leads to Fast MMNB; for details of training process we can refer to [44]. At testing stage, we compute log-likelihood $\ell = \log p(X|\alpha, \Omega)$ for each testing event corresponding LNND descriptor X under the trained Fast MMNB. X is classified as an anomaly if the following criterion is satisfied:

$$\ell < \delta, \quad (13)$$

where δ is a user defined threshold that controls sensitivity of the algorithm to anomaly detection. For dealing with the anomalies occurring in different scales, we perform anomaly detection at three scales of pyramid structures (illustrated in Figure 6), and the final log-likelihood map is generated via a product rule, resulting in the spatial intersection of the three detected regions.

5. Experiments

In this section, we validate the advantages of LNND descriptor by conducting extensive experiments on two public datasets including UCSD dataset and UMN dataset (unusual crowd activity dataset of Minnesota University available at http://mha.cs.umn.edu/Movies/Crowd-Activity-All.avi). In this work, we search 8 spatial nearest neighbors and 1 temporal nearest neighbor; that is, set $K = 8$ and $N = 1$, so the dimensionality of LNND descriptor is 9, and the topic number of Fast MMNB is set to 10.

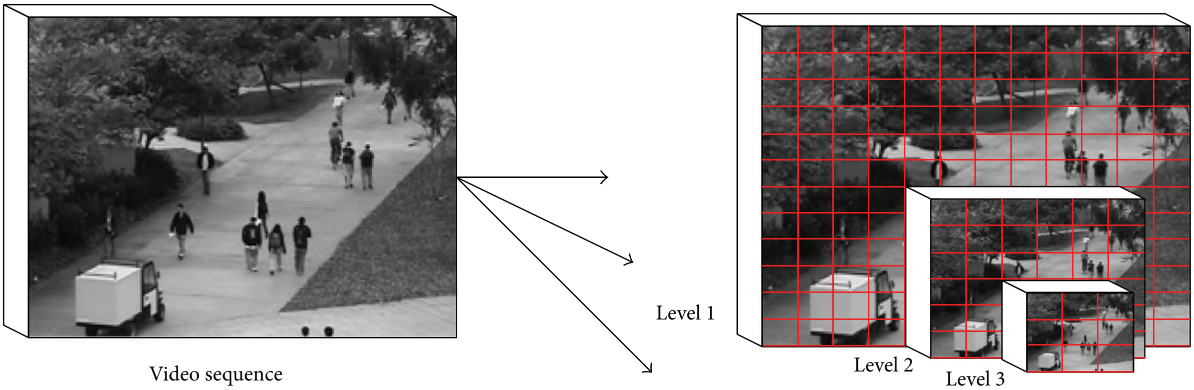


Figure 6: Three level pyramid structures of video sequences.

5.1. UCSD Dataset. The UCSD dataset includes Ped1 and Ped2 subsets and is captured by a fixed camera from two scenes in UCSD campus, respectively. The dataset exhibits crowd moving scenes. The crowd in normal training set only contains the moving pedestrians with normal speed. The anomalies in the testing set are either nonpedestrians moving object or anomalous behavior, such as skaters, bikes, small cars, and pedestrian's irregular motion. The Ped1 contains 34 training sequences and 36 testing sequences. Each sequence contains 200 frames, so the training set has 6800 frames, and the testing set has 7200 frames. Each frame is of size 158 × 238 and is resized into 160×240. The Ped2 contains 16 training and 12 testing sequences, and each sequence contains 120 to 180 sequences. Due to the fact that the ground truths of 3 testing sequences are not provided, we use 9 testing sequences of them for testing our method. For Ped1 subset, we divide video sequence into a set of spatiotemporal cuboids; each cuboid is of size 16×24×5 with 50% spatial pixel overlapping. For Ped2 subset, we divide video sequence into a set of spatiotemporal cuboids; each cuboid is of size 15×15×5 without overlapping. We adopt EER (equal error rate) to evaluate the performance for anomaly detection in UCSD dataset. The lower EER value is, the better performance is achieved. Figure 7 shows some detection results of our method. We can see from it that LNND descriptor-based method can well detect different types of anomalies, such as skaters, bicycles, and small carts. In Figures 8(a) and 8(b), the ROC curves of our method and other state-of-the-art methods for Ped1 and Ped2 are plotted for comparison, respectively. In Table 2, the summary of quantitative results of our method and other state-of-the-art-methods under different criterions is listed.

From Figure 8 and Table 2, we can see that our LNND descriptor-based method has high accuracy of anomaly detection. For Ped1 subset, the EER value of our method is 27.9% that is higher than MPPCA, SFM, and HSTG. Although the EER value of our method is lower than MHOF and MDT, the dimension and computation cost of LNND descriptor is much lower than them. For Ped2 subset, the performance of our LNND-based method outperforms the other comparable state-of-the-art methods. The average EER of our LNND-based method is 25.8% higher than that of MPPCA, SFM, HSTG, and LMH and is comparable to MDT.

Table 2: Summary of the EER values of different descriptor-based methods for comparison.

Descriptors	Ped1	Ped2	Average
MPPCA [3]	35.6%	35.8%	35.7%
SFM [2]	31%	42%	37%
MHOF [9]	19%	—	—
MDT [4]	22.9%	27.9%	25.4%
HSTG [13]	31%	30%	30%
MHOF [11]	15%	—	—
LMH [21]	38.9%	45.8%	42.3%
LNND	**27.9%**	**23.7%**	**25.8%**

5.2. UMN Dataset. The UMN dataset is captured from 3 different scenes, including indoor and outdoor scenes, and the resolution is 320 × 240. The abnormal event in the dataset is crowd panic escaping. They start with the normal event followed by the abnormal events. We portion each scene into two parts. The first part contains 400 frames and is used as training set which only contains normal events, and the rest is used as testing set which contains both normal and abnormal events. In the training stage, video sequences portioned the video into a set of cuboids with size of 10 × 10 × 5. Figure 9 shows some detection results of abnormal event detection from UMN dataset. We can see from it that the anomaly occurred region can be well detected. Figure 10 and Table 3 show the ROC curves and AUC values of different methods, respectively. We can see from it that our method has promising performance that can be comparable to state-of-the art methods.

6. Conclusions

In this paper, we propose a novel LNND descriptor to represent the video event for anomaly detection in crowded scenes. Compared with commonly used low-level feature descriptor in previous works, our LNND descriptor has two advantages. First, both the spatial and temporal contexts are incorporated into LNND descriptor. Using LNND descriptor, both the anomalous event and interactions between multiple

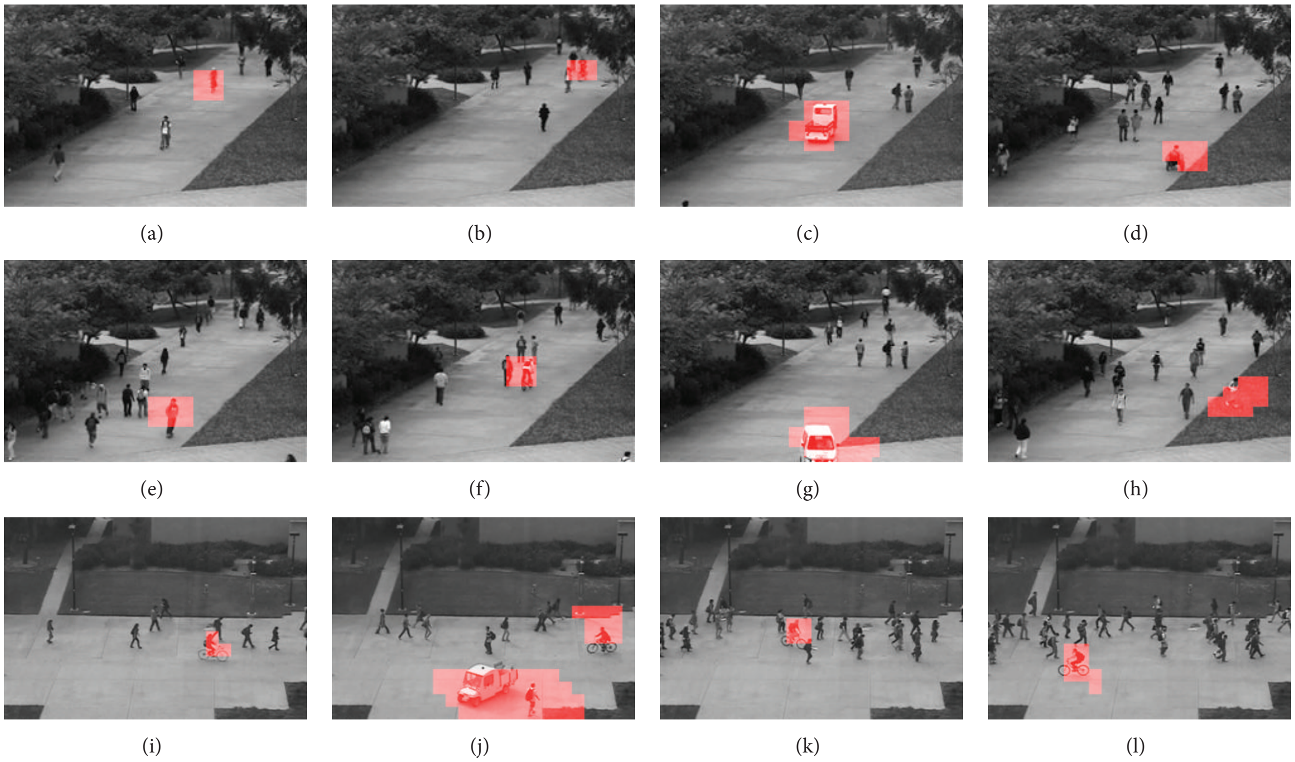


Figure 7: Some detection results on UCSD dataset.

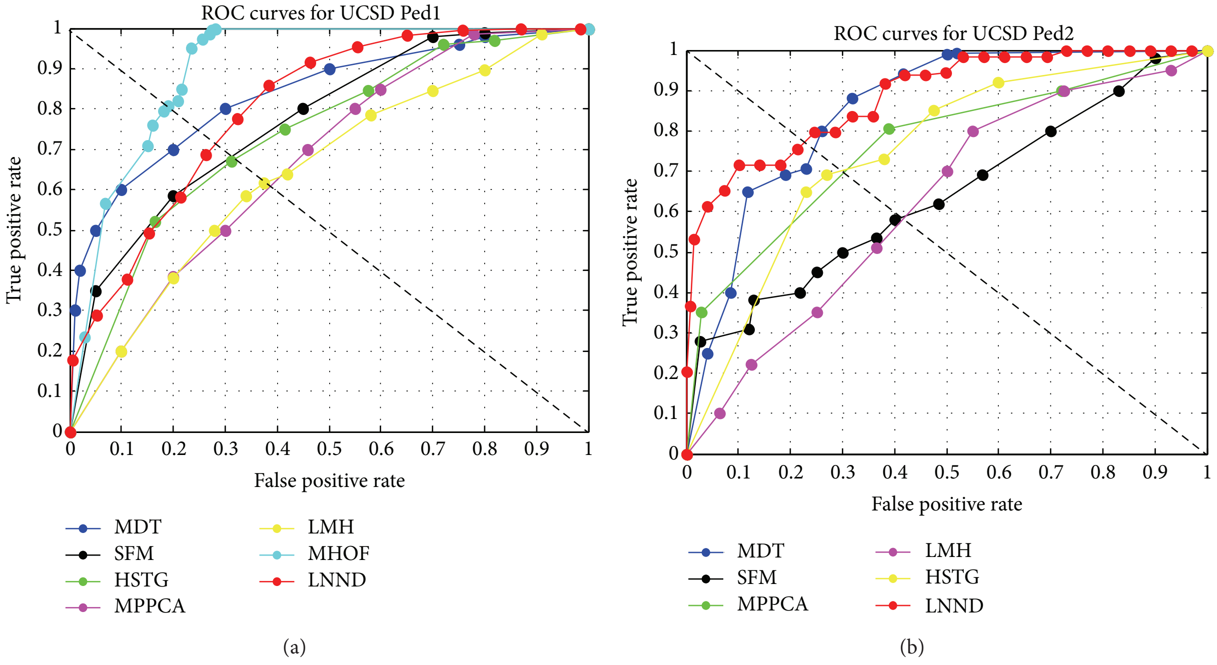


Figure 8: ROC curves of our method and other state-of-the-art methods for (a) UCSD Ped1 and (b) Ped2 datasets.

events can be well detected by training a simple temporal model, unlike previous low-level feature descriptor which needs to rely on a spatiotemporal model to account for spatial context. Second, due to the low dimensionality of LNND descriptor, both the computation time and the memory requirement can be accordingly saved. We perform anomaly detection in UCSD and UMN datasets, and the results are provided for comparing with other state-of-the-art methods. The qualitative and quantitative analyses of experimental results demonstrate that our proposed LNND descriptor is

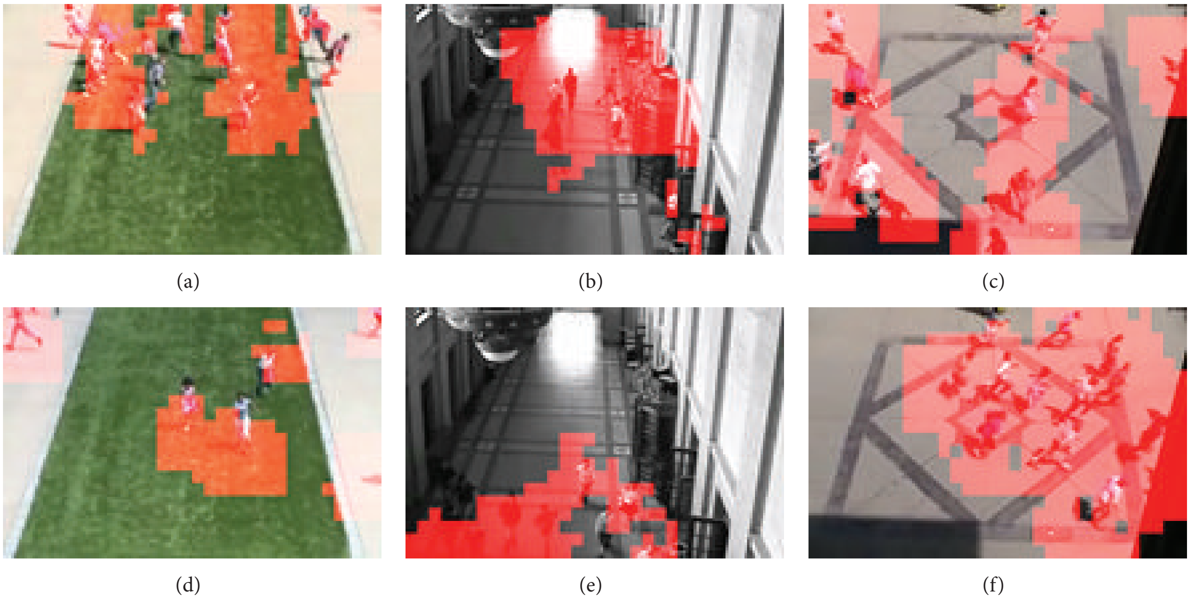


FIGURE 9: Some detection results of our method for UMN dataset.

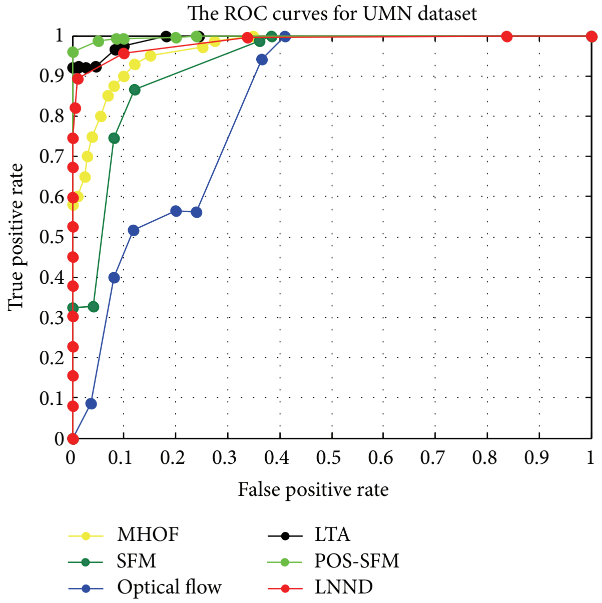


FIGURE 10: ROC curves of our method and other state-of-the-art methods for UMN dataset.

a concise and efficient descriptor. Compared with commonly used low-level feature descriptor on previous anomaly detection works, our LNND-based method is computationally efficient and robust and has promising result. Meanwhile, the intermediate and subsequent process is less than most previous works. As our future work, we will attempt to use our proposed LNND descriptor to some other applications, such as event or action recognition.

TABLE 3: Summary of the EER values of different descriptor-based methods for comparison.

Method	AUC
Chaotic invariants [9]	0.99
SFM [2]	0.96
Optical flow [9]	0.84
MHOF [9]	0.978
PSO SFM [22]	0.996
LTA [22]	0.992
LNND	0.986

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This paper is jointly supported by the National Natural Science Foundation of China "61374161" and "61074106."

References

[1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: a survey," *ACM Computing Surveys*, vol. 41, no. 3, article 15, 2009.

[2] R. Mehran, A. Oyama, and M. Shah, "Abnormal crowd behavior detection using social force model," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '09)*, pp. 935–942, Miami, Fla, USA, 2009.

[3] J. Kim and K. Grauman, "Observe locally, infer globally: a space-time MRF for detecting abnormal activities with incremental

updates," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR '09)*, pp. 2921–2928, Miami, Fla, USA, June 2009.

[4] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos, "Anomaly detection in crowded scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 1975–1981, San Francisco, Calif, USA, June 2010.

[5] Y. Cong, J. Yuan, and J. Liu, "Sparse reconstruction cost for abnormal event detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 3449–3456, 2011.

[6] S. S. Pathan, A. Al-Hamadi, and B. Michaelis, "Using conditional random field for crowd behavior analysis," in *Computer Vision–ACCV 2010 Workshops*, pp. 370–379, Springer, Berlin, Germany, 2010.

[7] B. Zhao, L. Fei-Fei, and E. P. Xing, "Online detection of unusual events in videos via dynamic sparse coding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 3313–3320, IEEE, June 2011.

[8] J. Li, S. Gong, and T. Xiang, "Learning behavioural context," *International Journal of Computer Vision*, vol. 97, no. 3, pp. 276–304, 2012.

[9] Y . Cong, J. Yuan, and J. Liu, "Abnormal event detection in crowded scenes using sparse representation," *Pattern Recognition*, vol. 46, no. 7, pp. 1851–1864, 2013.

[10] M. Thida, H. L. Eng, and P. Remagnino, "Laplacian eigenmap with temporal constraints for local abnormality detection in crowded scenes," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 2147–2156, 2013.

[11] X. Zhu, J. Liu, J. Wang et al., "Sparse representation for robust abnormality detection in crowded scenes," *Pattern Recognition*, vol. 47, no. 5, pp. 1791–1799, 2014.

[12] L. Kratz and K. Nishino, "Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR '2009)*, pp. 1446–1453, Miami, Fla, USA, June 2009.

[13] M. Bertini, A. del Bimbo, and L. Seidenari, "Multi-scale and real-time non-parametric approach for anomaly detection and localization," *Computer Vision and Image Understanding*, vol. 116, no. 3, pp. 320–329, 2012.

[14] Y. Ma and P. Cisar, "Event detection using local binary pattern based dynamic textures," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '09)*, pp. 38–44, June 2009.

[15] A. Zaharescu and R. Wildes, "Anomalous behaviour detection using spatiotemporal oriented energies, subset inclusion histogram comparison and event-driven processing," in *Computer Vision—ECCV 2010*, pp. 563–576, Springer, Berlin, Germany, 2010.

[16] J. Xu, S. Denman, C. Fookes, and S. Sridharan, "Unusual event detection in crowded scenes using bag of LBPs in spatio-temporal patches," in *Proceedings of the International Conference on Digital Image Computing: Techniques and Applications (DICTA '11)*, pp. 549–554, Noosa, Australia, December 2011.

[17] W. Li, V. Mahadevan, and N. Vasconcelos, "Anomaly detection and localization in crowded scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 1, pp. 18–32, 2013.

[18] A. Briassouli and I. Kompatsiaris, "Spatiotemporally localized new event detection in crowds," in *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV '11)*, pp. 928–933, Barcelona, Spain, November 2011.

[19] B. Wang, M. Ye, X. Li, F. Zhao, and J. Ding, "Abnormal crowd behavior detection using high-frequency and spatio-temporal features," *Machine Vision and Applications*, vol. 23, no. 3, pp. 501–511, 2012.

[20] T. Guha and R. K. Ward, "Learning sparse representations for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 8, pp. 1576–1588, 2012.

[21] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz, "Robust real-time unusual event detection using multiple fixed-location monitors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 3, pp. 555–560, 2008.

[22] R. Raghavendra, A. Del Bue, M. Cristani, and V. Murino, "Optimizing interaction force for global anomaly detection in crowded scenes," in *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV Workshops '11)*, pp. 136–143, Barcelona, Spain, November 2011.

[23] M. Javan Roshtkhari and M. D. Levine, "An on-line, real-time learning method for detecting anomalies in videos using spatio-temporal compositions," *Computer Vision and Image Understanding*, vol. 117, no. 10, pp. 1436–1452, 2013.

[24] H. Ling and K. Okada, "An efficient earth mover's distance algorithm for robust histogram comparison," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 5, pp. 840–853, 2007.

[25] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth, "Occam's razor," *Information Processing Letters*, vol. 24, no. 6, pp. 377–380, 1987.

[26] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 747–757, 2000.

[27] C. Piciarelli, C. Micheloni, and G. L. Foresti, "Trajectory-based anomalous event detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 11, pp. 1544–1554, 2008.

[28] T. Chen P, H. Haussecker, A. Bovyrin et al., "Computer vision workload analysis: case study of video surveillance systems," *Intel Technology Journal*, vol. 9, pp. 109–118, 2005.

[29] N. Johnson and D. Hogg, "Learning the distribution of object trajectories for event recognition," *Image and Vision Computing*, vol. 14, no. 8, pp. 609–615, 1996.

[30] F. Jiang, Y. Wu, and A. K. Katsaggelos, "A dynamic hierarchical clustering method for trajectory-based unusual video event detection," *IEEE Transactions on Image Processing*, vol. 18, no. 4, pp. 907–913, 2009.

[31] C. Li, Z. Han, Q. Ye, and J. Jiao, "Visual abnormal behavior detection based on trajectory sparse reconstruction analysis," *Neurocomputing*, vol. 119, pp. 94–100, 2013.

[32] S. Wu, B. E. Moore, and M. Shah, "Chaotic invariants of lagrangian particle trajectories for anomaly detection in crowded scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 2054–2060, San Francisco, Calif, USA, June 2010.

[33] Y. Benezeth, P.-M. Jodoin, V. Saligrama, and C. Rosenberger, "Abnormal events detection based on spatio-temporal co-occurences," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR '09)*, pp. 2458–2465, Miami, Fla, USA, June 2009.

[34] V. Reddy, C. Sanderson, and B. C. Lovell, "Improved anomaly detection in crowded scenes via cell-based analysis of foreground speed, size and texture," in *Proceedings of the Computer*

Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW '11), pp. 55–61, IEEE, June 2011.

[35] V. Saligrama and Z. Chen, "Video anomaly detection based on local statistical aggregates," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 2112–2119, Providence, RI, USA, June 2012.

[36] B. Antić and B. Ommer, "Video parsing for abnormality detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 2415–2422, Barcelona, Spain, November 2011.

[37] M. J. Roshtkhari and M. D. Levine, "Online dominant and anomalous behavior detection in videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 2611–2618, IEEE, Portland, Ore, USA, June 2013.

[38] X. Cui, Q. Liu, M. Gao, and D. N. Metaxas, "Abnormal detection using interaction energy potentials," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 3161–3167, June 2011.

[39] M. H. Sharif and C. Djeraba, "An entropy approach for abnormal activities detection in video streams," *Pattern Recognition*, vol. 45, no. 7, pp. 2543–2561, 2012.

[40] Y. Rubner, C. Tomasi, and L. J. Guibas, "Earth mover's distance as a metric for image retrieval," *International Journal of Computer Vision*, vol. 40, no. 2, pp. 99–121, 2000.

[41] O. Pele and M. Werman, "Fast and robust earth mover's distances," in *Proceedings of the 12th International Conference on Computer Vision (ICCV '09)*, pp. 460–467, October 2009.

[42] S. Shirdhonkar and D. W. Jacobs, "Approximate earth mover's distance in linear time," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, June 2008.

[43] V. J. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial Intelligence Review*, vol. 22, no. 2, pp. 85–126, 2004.

[44] H. Shan and A. Banerjee, "Mixed-membership naive Bayes models," *Data Mining and Knowledge Discovery*, vol. 23, no. 1, pp. 1–62, 2011.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 267872, 9 pages
http://dx.doi.org/10.1155/2014/267872

Research Article

Real-Time Hand Gesture Recognition Using Finger Segmentation

Zhi-hua Chen,[1] Jung-Tae Kim,[1] Jianning Liang,[1] Jing Zhang,[1,2] and Yu-Bo Yuan[1]

[1] *Department of Computer Science and Engineering, East China University of Science and Technology, Shanghai 200237, China*
[2] *State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China*

Correspondence should be addressed to Jianning Liang; j.n.liang@ecust.edu.cn and Yu-Bo Yuan; ybyuan@ecust.edu.cn

Received 24 April 2014; Accepted 29 May 2014; Published 25 June 2014

Academic Editor: Shan Zhao

Hand gesture recognition is very significant for human-computer interaction. In this work, we present a novel real-time method for hand gesture recognition. In our framework, the hand region is extracted from the background with the background subtraction method. Then, the palm and fingers are segmented so as to detect and recognize the fingers. Finally, a rule classifier is applied to predict the labels of hand gestures. The experiments on the data set of 1300 images show that our method performs well and is highly efficient. Moreover, our method shows better performance than a state-of-art method on another data set of hand gestures.

1. Introduction

As we know, the vision-based technology of hand gesture recognition is an important part of human-computer interaction (HCI). In the last decades, keyboard and mouse play a significant role in human-computer interaction. However, owing to the rapid development of hardware and software, new types of HCI methods have been required. In particular, technologies such as speech recognition and gesture recognition receive great attention in the field of HCI.

Gesture is a symbol of physical behavior or emotional expression. It includes body gesture and hand gesture. It falls into two categories: static gesture [1–4] and dynamic gesture [5–8]. For the former, the posture of the body or the gesture of the hand denotes a sign. For the latter, the movement of the body or the hand conveys some messages. Gesture can be used as a tool of communication between computer and human [9–11]. It is greatly different from the traditional hardware based methods and can accomplish human-computer interaction through gesture recognition. Gesture recognition determines the user intent through the recognition of the gesture or movement of the body or body parts. In the past decades, many researchers have strived to improve the hand gesture recognition technology. Hand gesture recognition has great value in many applications such as sign language recognition [12–15], augmented reality (virtual reality) [16–19], sign language interpreters for the disabled [20], and robot control [21, 22].

In [12, 13], the authors detect the hand region from input images and then track and analyze the moving path to recognize America sign language. In [23], Shimada et al. propose a TV control interface using hand gesture recognition. Keskin et al. [24] divide the hand into 21 different regions and train a SVM classifier to model the joint distribution of these regions for various hand gestures so as to classify the gestures. Zeng et al. [20] improve the medical service through the hand gesture recognition. The HCI recognition system of the intelligent wheelchair includes five hand gestures and three compound states. Their system performs reliably in the environment of indoor and outdoor and in the condition of lighting change.

The work flow of hand gesture recognition [25–27] is described as follows. First, the hand region is detected from the original images from the input devices. Then, some kinds of features are extracted to describe hand gestures. Last, the recognition of hand gestures is accomplished by measuring the similarity of the feature data. The input devices providing the original image information includes normal camera, stereo camera, and ToF (time of flight) camera. The stereo camera and ToF camera additionally provide the depth information so it is easy to segment the hand region from the background in terms of the depth map. For the normal camera, the skin color sensitive to the lighting condition and feature points are combined to robustly detect and segment the hand region. When the region of interest (ROI, the hand region in the case) is detected, features are needed to be extracted from the ROI region. Color, brightness, and gradient values are widely used features. Li and Kitani [28] describe various features for hand region detecting including the Gabor filter response, HOG, SIFT, BRIEF, and ORB. For the recognition of hand gestures, various classifiers, for example, SVM (support vector machine), HMM (hidden Markov model), CRF (conditional random field), and adapted boosting classifier are trained to discriminate hand gestures. Although the recognition performance of these sophisticated classifiers is good, the time cost is very high.

In this paper, we present an efficient and effective method for hand gesture recognition. The hand region is detected through the background subtraction method. Then, the palm and fingers are split so as to recognize the fingers. After the fingers are recognized, the hand gesture can be classified through a simple rule classifier.

The novelty of the proposed method is listed as follows.

(i) The first novelty of the proposed method is that the hand gesture recognition is based on the result of finger recognition. Therefore, the recognition is accomplished by a simple and efficient rule classifier instead of the sophisticated but complicated classifiers such as SVM and CRF.

(ii) Some previous works need the users to wear data glove [29] to acquire hand gesture data. However, the special sensors of data glove are expensive and hinder its wide application in real life. In the work [25], the authors use TOF camera, that is, Kinect sensor, to capture the depth of the environment and a special tape worn across the wrist to detect hand region. Our approach only uses a normal camera to capture the vision information of the hand gesture meanwhile does not need the help of the special tape to detect hand regions.

(iii) The third advantage of the proposed method is that it is highly efficient and fit for real-time applications.

The rest of the paper is organized as follows. In Section 2, the proposed method for hand gesture recognition is described in detail. In Section 3, the performance of our approach is evaluated on a data set of hand gestures. Then, our method is compared with a state-of-art method (FEMD) [25] on another data set of hand gestures. Section 4 presents the conclusion and future works.

2. The Proposed Method for Hand Gesture Recognition

2.1. The Overview of the Method. The overview of the hand gesture recognition is described in Figure 1. First, the hand is detected using the background subtraction method and the result of hand detection is transformed to a binary image. Then, the fingers and palm are segmented so as to facilitate the finger recognition. Moreover, the fingers are detected and recognized. Last, hand gestures are recognized using a simple rule classifier.

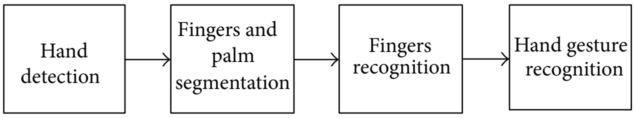
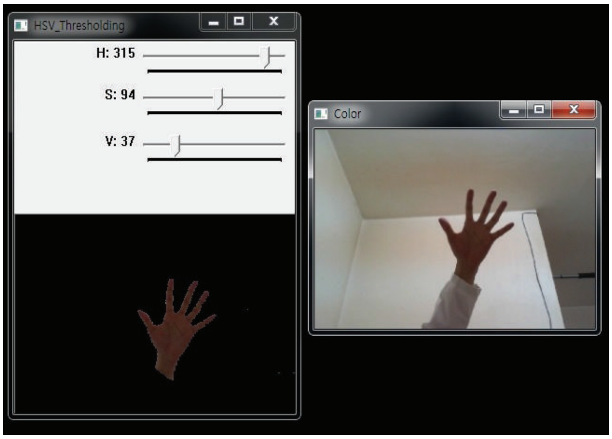

Hand detection → Fingers and palm segmentation → Fingers recognition → Hand gesture recognition

Figure 1: The overview of the proposed method for hand gesture recognition.



H: 315
S: 94
V: 37

Figure 2: The procedure of hand detection.

2.2. Hand Detection. The original images used for hand gesture recognition in the work are demonstrated in Figure 2. These images are captured with a normal camera. These hand images are taken under the same condition. The background of these images is identical. So, it is easy and effective to detect the hand region from the original image using the background subtraction method. However, in some cases, there are other moving objects included in the result of background subtraction. The skin color can be used to discriminate the hand region from the other moving objects. The color of the skin is measured with the HSV model. The HSV (hue, saturation, and value) value of the skin color is 315, 94, and 37, respectively. The image of the detected hand is resized to 200 × 200 to make the gesture recognition invariant to image scale.

2.3. Fingers and Palm Segmentation. The output of the hand detection is a binary image in which the white pixels are the members of the hand region, while the black pixels belong to the background. An example of the hand detection result is shown in Figure 3. Then, the following procedure is implemented on the binary hand image to segment the fingers and palm.

(i) Palm Point. The palm point is defined as the center point of the palm. It is found by the method of distance transform. Distance transform also called distance map is a representation of an image. In the distance transform image,

```
**Input**: A Group of points sampled uniformly from the circle
Find the nearest boundary point of one sampled point $(X, Y)$ (refer to (1))
*Step 1.* Acquire a pixel $(x, y)$ around the sample point $(X, Y)$
        $x = \cos\left(\frac{\text{angle} * \pi}{180}\right) * \text{rad} + X$
        $y = \sin\left(\frac{\text{angle} * \pi}{180}\right) * \text{rad} + Y$
angle $\epsilon$ [0, 360] and rad $\epsilon$ $[1, L]$ ($L$ is the image size)
*Step 2.* If the value of the pixel is 0 i.e. $P(x, y) == 0$,
goto Step 3. Otherwise, increase rad and angle with a step of 1 and then go to Step 1
*Step 3.* Check the values of 8 neighbors of the pixel $(x, y)$, if it holds
    $\forall P(x + dx, y + dy) == 0, \quad (x + dx, y + dy) \in N_{(x,y)}$
$N_{(x,y)}$ is the set of 8 neighbors of the pixel $(x, y)$.
Insert the point $(x + dx, y + dy)$ into the array of palm mask points.
*Step 4.* Increase rad and angle, and then goto Step 1
(i) Continue to search the nearest boundary point of another sampled point until all
    the sampled points are scanned.
(ii) Connect all the points recorded in the array of palm mask points to yield the palm mask.
```

Algorithm 1: The method of producing the palm mask.

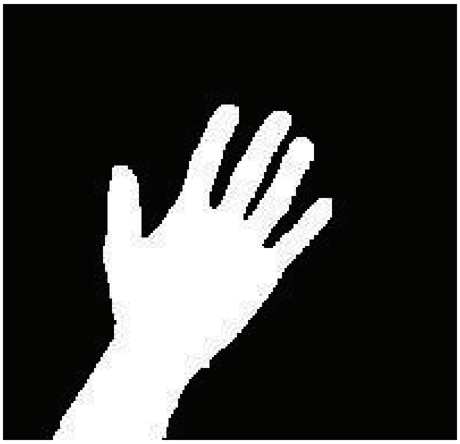


Figure 3: The detected hand region.

each pixel records the distance of it and the nearest boundary pixel. An example of distance transform is demonstrated in Figure 4. In Figure 4(a) is a binary image and in Figure 4(b) is the distance transform image. The block city distance is used to measure the distances between the pixels and the nearest boundary pixels. As is shown in the figure, the center point of the binary image is with the largest distance 4. Thus, in the distance transform image (refer to Figure 5) of the binary hand image, the pixel with largest distance is chosen as the palm point. The found palm point is marked with the point of the green color in Figure 6.

(ii) Inner Circle of the Maximal Radius. When the palm point is found, it can draw a circle with the palm point as the center point inside the palm. The circle is called the inner circle because it is included inside the palm. The radius of the circle gradually increases until it reaches the edge of the palm. That is the radius of the circle stops to increase when the black pixels are included in the circle. The circle is the inner circle of the maximal radius which is drawn as the circle with the red color in Figure 6.

(iii) Wrist Points and Palm Mask. When the radius of the maximal inner circle is acquired, a larger circle the radius of which is 1.2 times of that of the maximal inner circle is produced. The circle is drawn as the blue color circle in Figure 6. Then, some points (X, Y) are sampled uniformly along the circle. That is,

$$X = R\cos\left(\frac{\theta * \pi}{180}\right) + X_0, \qquad Y = R\sin\left(\frac{\theta * \pi}{180}\right) + Y_0, \qquad \theta = 0 : t : 360, \tag{1}$$

where (X_0, Y_0) is the position of the palm point, R is the radius of the circle, and t is the sampling step.

For each sampled point on the circle, its nearest boundary point is found and lined to it. The boundary point is judged in a simple way. If the 8 neighbors of a pixel consist of white and black pixels, it is labeled as a boundary point. All of the nearest boundary points found are linked to yield the palm mask that can be used to segment fingers and the palm. The method for searching the palm mask is described in Algorithm 1. The palm mask of the hand image of Figure 3 is demonstrated in Figure 7. A larger circle instead of the maximal inner circle is used so as to yield a more accurate palm mask for the following segmentation.

Two wrist points are the two ending points of the wrist line across the bottom of the hand. The wrist points are important points for hand gesture recognition. They can be searched in the following manner: if the distance between two successive mask points P_i, P_{i+1} are large, these two mask points are judged as the wrist points. That is,

$$\arg\max_{P_i, P_{i+1}} \text{dist}\left(P_i, P_{i+1}\right), \quad P_i, P_{i+1} \in S, \tag{2}$$

0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0	0

(a)

0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	0
0	1	2	2	2	2	2	1	0
0	1	2	3	3	3	2	1	0
0	1	2	3	4	3	2	1	0
0	1	2	3	3	3	2	1	0
0	1	2	2	2	2	2	1	0
0	1	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0	0

(b)

Figure 4: An example of distance transform: (a) is a binary image; (b) is the distance transform.

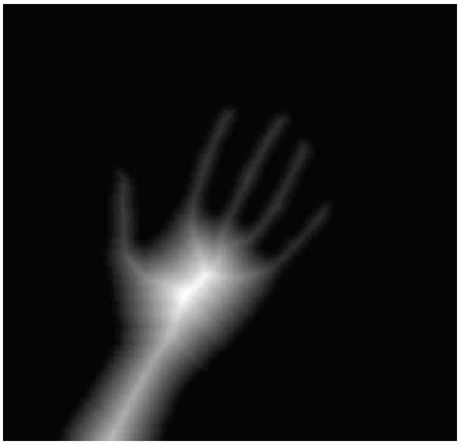


Figure 5: The distance transform of the hand image in Figure 3.

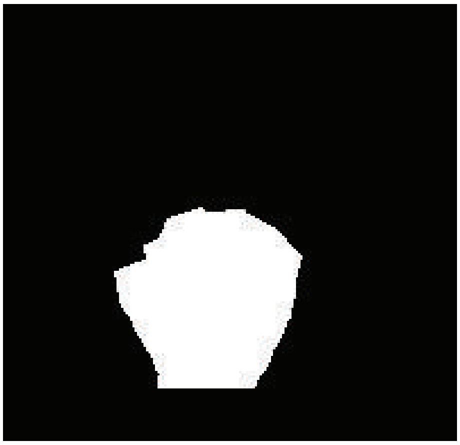


Figure 7: The palm mask.

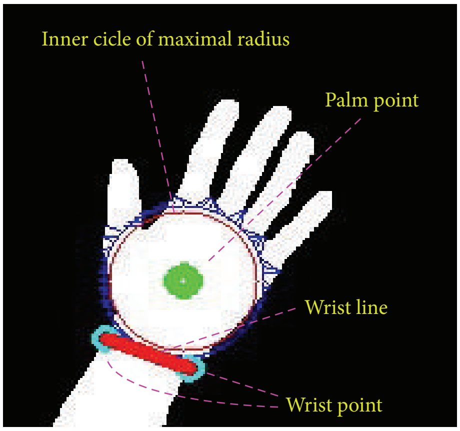


Figure 6: The palm point, wrist points, the wrist line, and the inner circle of the maximal radius.

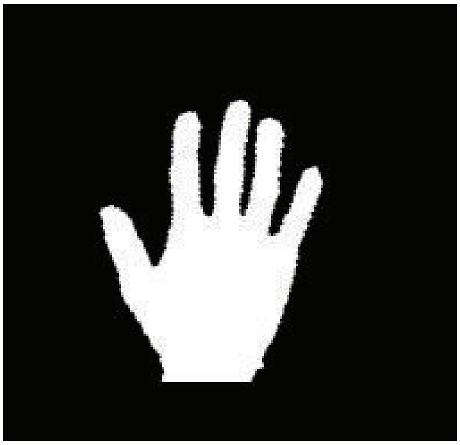


Figure 8: The rotated and cut hand image.

where S is the set of palm mask points and dist$(*, *)$ is the distance between two points. Please refer to Figure 6 for the wrist points and wrist line.

(iv) Hand Rotation. When the palm point and wrist point are obtained, it can yield an arrow pointing from the palm point to the middle point of the wrist line at the bottom of the hand. Then, the arrow is adjusted to the direction of the north. The hand image rotates synchronously so as to make the hand gesture invariant to the rotation. Meanwhile, the parts below the wrist line in the rotated image are cut to produce an accurate hand image that does not enclose the part of the arm. Figure 8 is the rotated and cut hand image.

(v) Fingers and Palm Segmentation. With the help of the palm mask, fingers and the palm can be segmented easily. The part of the hand that is covered by the palm mask is the palm, while the other parts of the hand are fingers. A segmentation result of fingers and the palm is shown in Figure 9.

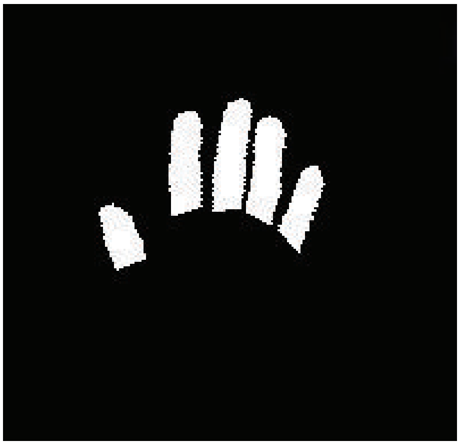


Figure 9: The segmented fingers.

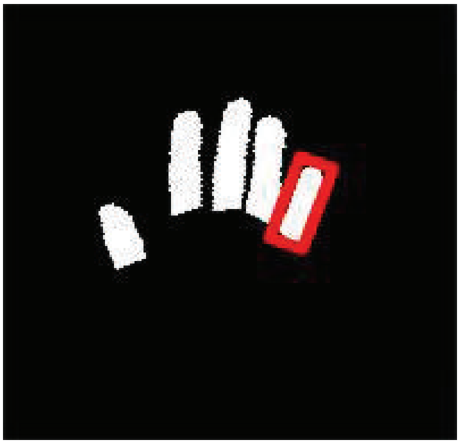


Figure 10: The minimal bounding box.

2.4. *Fingers Recognition.*

In the segmentation image of fingers, the labeling algorithm is applied to mark the regions of the fingers. In the result of the labeling method, the detected regions in which the number of pixels is too small is regarded as noisy regions and discarded. Only the regions of enough sizes are regarded as fingers and remain. For each remained region, that is, a finger, the minimal bounding box is found to enclose the finger. A minimal bounding box is denoted as a red rectangle in Figure 10. Then, the center of the minimal bounding box is used to represent the center point of the finger.

(i) Thumb Detection and Recognition. The centers of the fingers are lined to the palm point. Then, the degrees between these lines and the wrist line are computed. If there is a degree smaller than 50°, it means that the thumb appears in the hand image. The corresponding center is the center point of the thumb. The detected thumb is marked with the number 1. If all the degrees are larger than 50°, the thumb does not exist in the image.

(ii) Detection and Recognition of Other Fingers. In order to detect and recognize the other fingers, the palm line is first searched. The palm line parallels to the wrist line. The palm line is searched in the way: start from the row of the wrist line. For each row, a line paralleling to the wrist line crosses the hand. If there is only one connected set of white pixels in the intersection of the line and the hand, the line shifts upward. Once there are more than one connected sets of white pixels in the intersection of the line and the hand, the line is regarded as a candidate of the palm line. In the case of the thumb not detected, the line crossing the hand with more than one connected sets of white pixels in their intersection is chosen as the palm line. In the case of the thumb existing, the line continues to move upward with the edge points of the palm instead of the thumb as the starting point of the line. Now, since the thumb is taken away, there is only one connected set of pixels in the intersection of the line and the hand. Once the connected set of white pixels turns to 2 again, the palm line is found. The search of the palm line is shown in Figure 11.

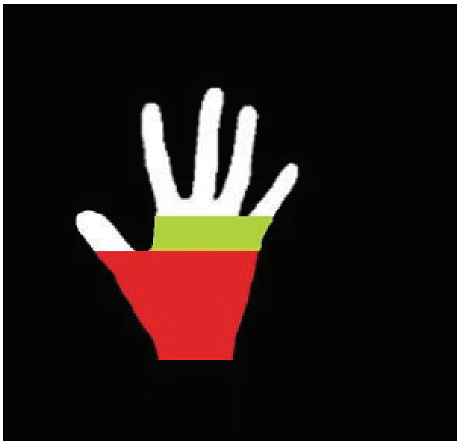


Figure 11: The palm line.

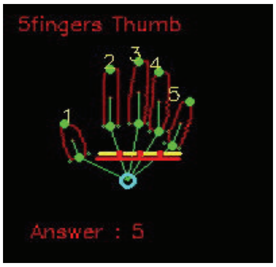


Figure 12: The recognition of the fingers.

After the palm line is obtained, it is divided into 4 parts. According to the horizontal coordinate of the center point of a finger, it falls into certain parts. If the finger falls into the first part, it is the forefinger. If the finger belongs to the second part, it is the middle finger. The third part corresponds to the ring finger. The fourth part is the little finger. The result of finger recognition of Figure 3 is demonstrated in Figure 12. In the figure, the yellow line is the palm line and the red line parallels to the wrist line.

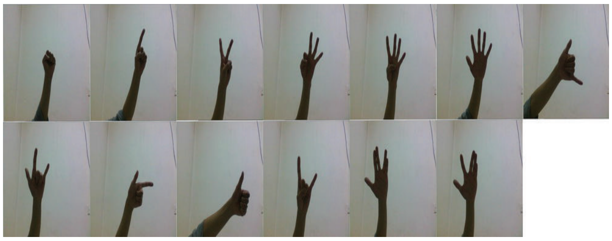


Figure 13: The image set of hand gestures used in the experiments. From left to right and then from top to bottom; these gestures are labeled as 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, S1, S2, and S3.

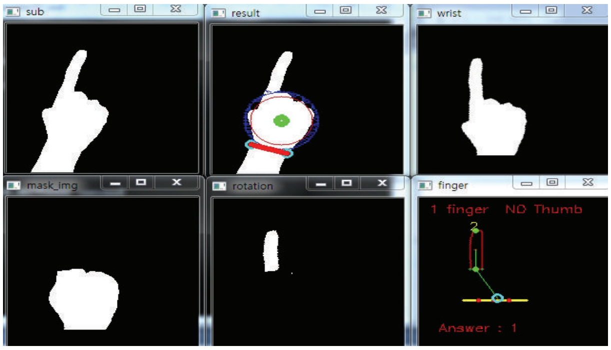
In some case, two or more fingers stay closely and there is no interval among the fingers. An example of the case is referred to Figure 18. In order to discriminate the case from that of a single finger, the width of the minimal bounding box is used as a discrimination index. If the width of the minimal bounding box is equal to a usual value, the detected region is a single finger. If the width of the minimal bounding box is several times of the usual value, the detected region corresponds to several fingers that stay together closely. For the robustness of finger recognition, the distances and angles between fingers are also taken into account to discriminate different gestures.

2.5. Recognition of Hand Gestures. When the fingers are detected and recognized, the hand gesture can be recognized using a simple rule classifier. In the rule classifier, the hand gesture is predicted according to the number and content of fingers detected. The content of the fingers means what fingers are detected. The rule classifier is very effective and efficient. For example, if three fingers, that is, the middle finger, the ring finger, and the little finger, are detected, the hand gesture is classified as the label 3 (refer to Figure 13 for the labels of the hand gestures).

3. Experimental Results

3.1. Data Sets. In the experiments, two data sets of hand gestures are used to evaluate the performance of the proposed method. The data set 1 is an image collection of thirteen gestures. For each gesture, 100 images are captured. So, there are total 1300 images for hand gesture recognition. All the gesture images belong to 3 females and 4 males. The size of one gesture image is 640 × 480. The thirteen gestures are shown in Figure 13. From left to right and then from top to bottom, these gestures are labeled as 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, S1, S2, and S3.

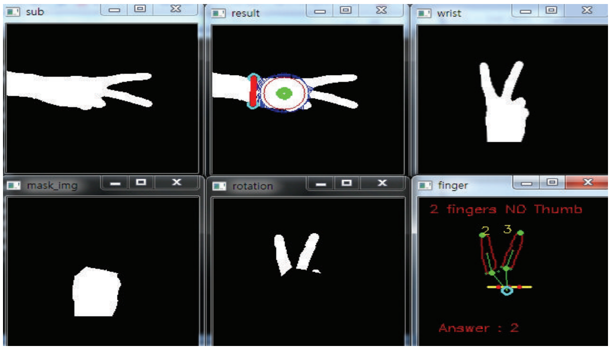
Another data set [25] is collected from 10 subjects, and it contains 10 gestures for number 0 to 9. So, there are total 10 × 10 × 10 cases. The data set captured in cluttered backgrounds is a great challenge for hand gesture recognition. Besides, for each gesture, the subject poses with variations in hand orientation, scale, articulation, and so forth. We compare our method with FEMD [25] on the data set.



1 finger NO Thumb

Answer : 1

Figure 14: The recognition of the hand gesture 1.



2 fingers NO Thumb

Answer : 2

Figure 15: The recognition of the hand gesture 2.

3.2. Performance Evaluation on Data Set 1

(i) Classification Accuracy. In order to measure the performance of the proposed hand gesture recognition method, the classification accuracy is evaluated in the experiments. In the training stage, the rules discriminating the thirteen gestures are produced. Then, the rule classifier uses the rules to predict the identification of the testing image. In Figures 14, 15, 16, 17, and 18, the recognition of five gestures are demonstrated. In each figure, there are six subfigures which are the images showing the binary hand image, the palm point and wrist line, the calibrated hand image, the palm mask, the detected fingers, and finger and gesture recognition, respectively. In the subfigure of finger and gesture recognition, the label of the gesture is predicted. The predicted label is shown behind the word "Answer."

The classification result of the total 1300 images is summarized with a confusion matrix in the Table 1. In the confusion matrix, the first column and the last row are the labels of the gestures. The other entries of the matrix record the numbers of the gesture images predicted as the corresponding labels. For example, for the first row, the numbers 99 and 1 are in the columns corresponding to the labels 1 and 3, respectively. It means that there are 99 and 1 images predicted as the labels 1 and 3 in the 100 testing images of the gesture 1. So, for the testing images of the gesture 1, the classification accuracy is 99%. As is shown in the confusion matrix, the proposed method performs well and obtains very high classification accuracies. The total

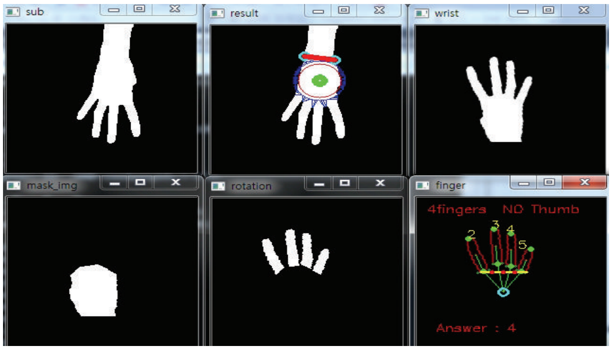


Figure 16: The recognition of the hand gesture 4.

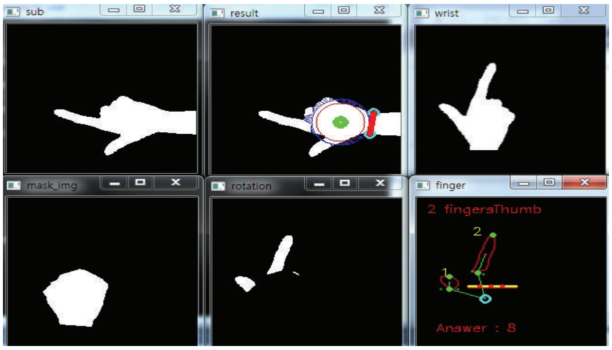


Figure 17: The recognition of the hand gesture 8.

classification accuracy of 1300 testing image is 96.69%. In the confusion matrix, the gestures of S2 and S3 are misclassified as 5. The reason is described as follows: for some gestures of S2 and S3, the fingers do not stay closely. That is, there is a hole between two fingers. So, in these cases, the gestures of S2 and S3 are misclassified as 5.

(ii) Time Cost. The time cost for recognizing the gestures is reported in Table 2. In the table, the unit of the time cost is second. A value in the second row is the averaging runtime of 100 images of one gesture. For the total 1300 images, the averaging time cost to recognize hand gestures is 0.024 seconds. The experiments are run on the laptop computer of Intel i7-2630 2.00 GHz CPU and 4 GB RAM. It is obvious that the proposed method is very highly efficient and can meet the requirement of the real-time applications.

3.3. Performance Comparison on Data Set 2. The comparison of the proposed method and a state-of-art method FEMD is performed on data set 2. The classification results are also summarized with the confusion matrixes. The description of the confusion matrixes is similar to that in Table 1. The confusion matrix of our method is shown in Table 3. The confusion matrix of FEMD is demonstrated in Table 4. The averaging accuracy of the proposed method is 96.6%. The averaging accuracy of FEMD is 93.2%. The comparison results on data set 2 show that our method outperforms FEMD. The averaging time of our method spent on recognizing a hand gesture is 0.0202 seconds.

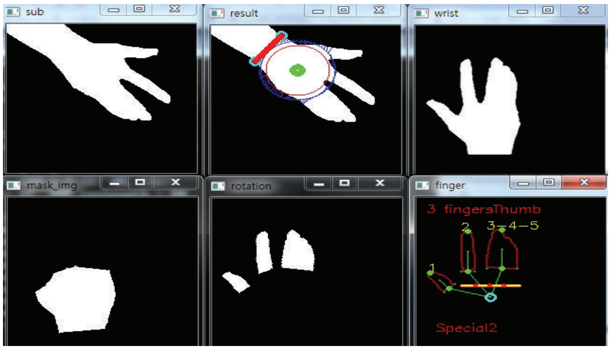


Figure 18: The recognition of the hand gesture S2.

Table 1: The confusion matrix of hand gesture recognition on data set 1.

1	99		1										
2		94	6										
3		2	95	3									
4			4	95	1								
5				3	93	4							
6						100							
7	4						96						
8	2		5					92	1				
9									100				
0	1									99			
S1			1								99		
S2					2							98	
S3					3								97
	1	2	3	4	5	6	7	8	9	0	S1	S2	S3

4. Conclusion and Future Works

A new method for hand gesture recognition is introduced in this paper. The hand region is detected from the background by the background subtraction method. Then, the palm and fingers are segmented. On the basis of the segmentation, the fingers in the hand image are discovered and recognized. The recognition of hand gestures is accomplished by a simple rule classifier. The performance of our method is evaluated on a data set of 1300 hand images. The experimental results show that our approach performs well and is fit for the real-time applications. Moreover, the proposed method outperforms the state-of-art FEMD on an image collection of hand gestures.

The performance of the proposed method highly depends on the result of hand detection. If there are moving objects with the color similar to that of the skin, the objects exist in the result of the hand detection and then degrade the performance of the hand gesture recognition. However, the machine learning algorithms can discriminate the hand from the background. ToF cameras provide the depth information that can improve the performance of hand detection. So, in

Table 2: The runtime of hand gesture recognition.

1	2	3	4	5	6	7	8	9	0	S1	S2	S3
0.024	0.021	0.022	0.024	0.027	0.023	0.026	0.022	0.025	0.022	0.022	0.026	0.021

Table 3: The confusion matrix of our method on data set 2.

0	99	1								
1		100								
2		7	91					1		1
3				100						
4					99		1			
5					3	97				
6							99	1		
7			2	1			9	88		
8		7							93	
9										100
	0	1	2	3	4	5	6	7	8	9

Table 4: The confusion matrix of FEMD on data set 2 (from [25]).

0	95	1						3	1	
1	3	86	4	2			1	4		
2		2	94	2			2			
3			4	87	6		3			
4				7	89	3	1			
5	1	2				95			2	
6			1			1	96	2		
7	6	2						92		
8	1					1			98	
9										100
	0	1	2	3	4	5	6	7	8	9

future works, machine learning methods and ToF cameras may be used to address the complex background problem and improve the robustness of hand detection.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by the National Nature Science Foundation of China (Grant no. 61370174 and Grant no. 61300133) and the Fundamental Research Funds for the Central Universities.

References

[1] A. D. Bagdanov, A. Del Bimbo, L. Seidenari, and L. Usai, "Real-time hand status recognition from RGB-D imagery," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR '12)*, pp. 2456–2459, November 2012.

[2] M. Elmezain, A. Al-Hamadi, and B. Michaelis, "A robust method for hand gesture segmentation and recognition using forward spotting scheme in conditional random fields," in *Proceedings of the 20th International Conference on Pattern Recognition (ICPR '10)*, pp. 3850–3853, August 2010.

[3] C.-S. Lee, S. Y. Chun, and S. W. Park, "Articulated hand configuration and rotation estimation using extended torus manifold embedding," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR '12)*, pp. 441–444, November 2012.

[4] M. R. Malgireddy, J. J. Corso, S. Setlur, V. Govindaraju, and D. Mandalapu, "A framework for hand gesture recognition and spotting using sub-gesture modeling," in *Proceedings of the 20th International Conference on Pattern Recognition (ICPR '10)*, pp. 3780–3783, August 2010.

[5] P. Suryanarayan, A. Subramanian, and D. Mandalapu, "Dynamic hand pose recognition using depth data," in *Proceedings of the 20th International Conference on Pattern Recognition (ICPR '10)*, pp. 3105–3108, August 2010.

[6] S. Park, S. Yu, J. Kim, S. Kim, and S. Lee, "3D hand tracking using Kalman filter in depth space," *Eurasip Journal on Advances in Signal Processing*, vol. 2012, no. 1, article 36, 2012.

[7] J. L. Raheja, A. Chaudhary, and K. Singal, "Tracking of fingertips and centers of palm using KINECT," in *Proceedings of the 2nd International Conference on Computational Intelligence, Modelling and Simulation (CIMSim '11)*, pp. 248–252, September 2011.

[8] Y. Wang, C. Yang, X. Wu, S. Xu, and H. Li, "Kinect based dynamic hand gesture recognition algorithm research," in *Proceedings of the 4th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC '12)*, pp. 274–279, August 2012.

[9] M. Panwar, "Hand gesture recognition based on shape parameters," in *Proceedings of the International Conference on Computing, Communication and Applications (ICCCA '12)*, pp. 1–6, February 2012.

[10] Z. Y. Meng, J.-S. Pan, K.-K. Tseng, and W. Zheng, "Dominant points based hand finger counting for recognition under skin color extraction in hand gesture control system," in *Proceedings of the 6th International Conference on Genetic and Evolutionary Computing (ICGEC '12)*, pp. 364–367, August 2012.

[11] R. Harshitha, I. A. Syed, and S. Srivasthava, "Hci using hand gesture recognition for digital sand model," in *Proceedings of the 2nd IEEE International Conference on Image Information Processing (ICIIP '13)*, pp. 453–457, 2013.

[12] R. Yang, S. Sarkar, and B. Loeding, "Handling movement epenthesis and hand segmentation ambiguities in continuous sign language recognition using nested dynamic programming," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 3, pp. 462–477, 2010.

[13] Z. Zafrulla, H. Brashear, T. Starner, H. Hamilton, and P. Presti, "American sign language recognition with the kinect," in *Proceedings of the 13th ACM International Conference on Multimodal Interfaces (ICMI '11)*, pp. 279–286, November 2011.

[14] D. Uebersax, J. Gall, M. Van den Bergh, and L. Van Gool, "Real-time sign language letter and word recognition from depth

data," in *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV '11)*, pp. 383–390, November 2011.

[15] N. Pugeault and R. Bowden, "Spelling it out: real-time ASL fingerspelling recognition," in *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV '11)*, pp. 1114–1119, November 2011.

[16] D. Wickeroth, P. Benölken, and U. Lang, "Markerless gesture based interaction for design review scenarios," in *Proceedings of the 2nd International Conference on the Applications of Digital Information and Web Technologies (ICADIWT '09)*, pp. 682–687, August 2009.

[17] V. Frati and D. Prattichizzo, "Using Kinect for hand tracking and rendering in wearable haptics," in *Proceedings of the IEEE World Haptics Conference (WHC '11)*, pp. 317–321, June 2011.

[18] J. Choi, H. Park, and J.-I. Park, "Hand shape recognition using distance transform and shape decomposition," in *Proceedings of the 18th IEEE International Conference on Image Processing (ICIP '11)*, pp. 3605–3608, September 2011.

[19] T.-D. Tan and Z.-M. Guo, "Research of hand positioning and gesture recognition based on binocular vision," in *Proceedings of the IEEE International Symposium on Virtual Reality Innovations (ISVRI '11)*, pp. 311–315, March 2011.

[20] J. Zeng, Y. Sun, and F. Wang, "A natural hand gesture system for intelligent human-computer interaction and medical assistance," in *Proceedings of the 3rd Global Congress on Intelligent Systems (GCIS '12)*, pp. 382–385, November 2012.

[21] D. Droeschel, J. Stückler, and S. Behnke, "Learning to interpret pointing gestures with a time-of-flight camera," in *Proceedings of the 6th ACM/IEEE International Conference on Human-Robot Interaction (HRI '11)*, pp. 481–488, March 2011.

[22] K. Hu, S. Canavan, and L. Yin, "Hand pointing estimation for human computer interaction based on two orthogonal-views," in *Proceedings of the 20th International Conference on Pattern Recognition (ICPR '10)*, pp. 3760–3763, August 2010.

[23] A. Shimada, T. Yamashita, and R.-I. Taniguchi, "Hand gesture based TV control system—towards both user—& machine-friendly gesture applications," in *Proceedings of the 19th Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV '13)*, pp. 121–126, February 2013.

[24] C. Keskin, F. Kiraç, Y. E. Kara, and L. Akarun, "Real time hand pose estimation using depth sensors," in *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV '11)*, pp. 1228–1234, November 2011.

[25] Z. Ren, J. Yuan, J. Meng, and Z. Zhang, "Robust part-based hand gesture recognition using kinect sensor," *IEEE Transactions on Multimedia*, vol. 15, no. 5, pp. 1110–1120, 2013.

[26] Z. Mo and U. Neumann, "Real-time hand pose recognition using low-resolution depth images," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, pp. 1499–1505, June 2006.

[27] S. Miyamoto, T. Matsuo, N. Shimada, and Y. Shirai, "Real-time and precise 3-D hand posture estimation based on classification tree trained with variations of appearances," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR '12)*, pp. 453–456, November 2012.

[28] C. Li and K. M. Kitani, "Pixel-level hand detection in egocentric videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 3570–3577, 2013.

[29] G. Dewaele, F. Devernay, and R. Horaud, "Hand motion from 3d point trajectories and a smooth surface model," in *Computer Vision—ECCV 2004*, vol. 3021 of *Lecture Notes in Computer Science*, pp. 495–507, Springer, 2004.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 404375, 23 pages
http://dx.doi.org/10.1155/2014/404375

Research Article

A DAG Scheduling Scheme on Heterogeneous Computing Systems Using Tuple-Based Chemical Reaction Optimization

Yuyi Jiang, Zhiqing Shao, and Yi Guo

College of Information Science and Engineering, East China University of Science and Technology, Shanghai 200237, China

Correspondence should be addressed to Zhiqing Shao; zshao@ecust.edu.cn

Received 1 April 2014; Revised 5 May 2014; Accepted 8 May 2014; Published 24 June 2014

Academic Editor: Yu-Bo Yuan

A complex computing problem can be solved efficiently on a system with multiple computing nodes by dividing its implementation code into several parallel processing modules or tasks that can be formulated as directed acyclic graph (DAG) problems. The DAG jobs may be mapped to and scheduled on the computing nodes to minimize the total execution time. Searching an optimal DAG scheduling solution is considered to be NP-complete. This paper proposed a tuple molecular structure-based chemical reaction optimization (TMSCRO) method for DAG scheduling on heterogeneous computing systems, based on a very recently proposed metaheuristic method, chemical reaction optimization (CRO). Comparing with other CRO-based algorithms for DAG scheduling, the design of tuple reaction molecular structure and four elementary reaction operators of TMSCRO is more reasonable. TMSCRO also applies the concept of constrained critical paths (CCPs), constrained-critical-path directed acyclic graph (CCPDAG) and super molecule for accelerating convergence. In this paper, we have also conducted simulation experiments to verify the effectiveness and efficiency of TMSCRO upon a large set of randomly generated graphs and the graphs for real world problems.

1. Introduction

Modern computer systems with multiple processors working in parallel may enhance the processing capacity for an application. The effective scheduling of parallel modules of the application may fully exploit the parallelism. The application modules may communicate and synchronize several times during the processing. The limitation of the overall application performance may be incurred by a large communication cost on heterogeneous systems with a combination of GPUs, multicore processors and CELL processors, or distributed memory systems. And an effective scheduling may greatly improve the performance of the application.

Scheduling generally defines not only the processing order of application modules but also the processor assignment of these modules. The concept of makespan (i.e., the schedule length) is used to evaluate the scheduling solution quality including the entire execution and communication cost of all the modules. On the heterogeneous systems [1–4], searching optimal schedules minimizing the makespan is considered as a NP-complete problem. Therefore, two classes of scheduling strategies have been proposed to solve this problem by finding the suboptimal solution with lower time overhead, such as heuristic scheduling and metaheuristic scheduling.

Heuristic scheduling strategies try to identify a good solution by exploiting the heuristics. An important subclass of heuristic scheduling is list scheduling with an ordered task list for a DAG job on the basis of some greedy heuristics. Moreover, the ordered tasks are selected to be allocated to the processors which minimize the start times in list scheduling algorithms. In heuristic scheduling, the attempted solutions are narrowed down by greedy heuristics to a very small portion of the entire solution space. And this limitation of the solution searching leads to the low time complexity. However, the higher complexity DAG scheduling problems have, the harder greedy heuristics produce consistent results on a wide range of problems, because the quality of the found solutions relies on the effectiveness of the heuristics, heavily.

Metaheuristic scheduling strategies such as ant colony optimization (ACO), genetic algorithms (GA), Tabu search (TS), simulated annealing (SA), and so forth take more

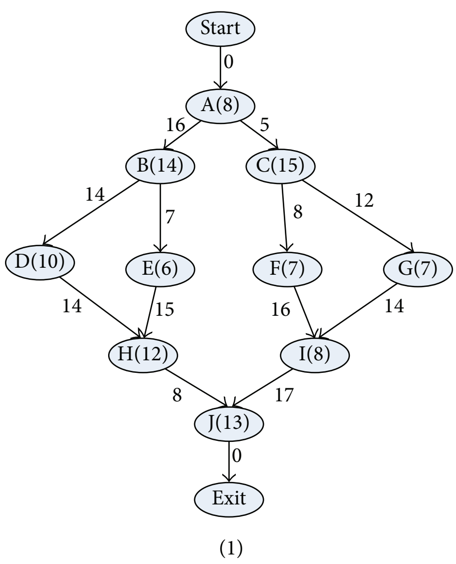


FIGURE 1: Two simple DAG models with 7 and 10 tasks.

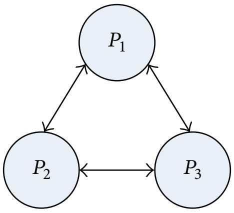


FIGURE 2: A fully connected parallel system with 3 heterogeneous processors.

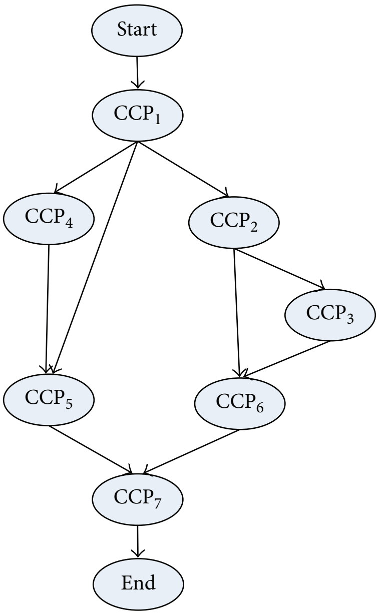


FIGURE 3: CCPDAG corresponding to the DAG as shown in Figure 1 and the CCP as indicated in Table 1.

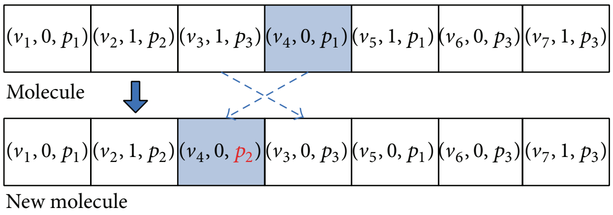


FIGURE 4: Illustration of molecular structure change for on-wall ineffective collision.

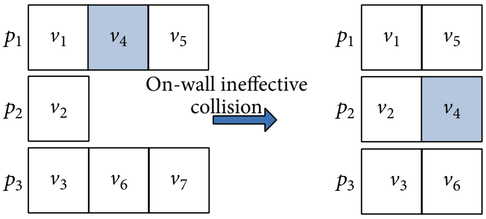


FIGURE 5: Illustration of the task-to-computing-node mapping for on-wall ineffective collision.

time cost than heuristic scheduling strategies, but they can produce consistent results with high quality on the problems with a wide range by directed searching solution spaces.

Chemical reaction optimization (CRO) is a new metaheuristic method proposed very recently and has shown its power to deal with NP-complete problem. There is only one CRO-based algorithm called double molecular structure-based CRO (DMSCRO) for DAG scheduling on heterogeneous system as far as we know. DMSCRO has a better performance on makespan and convergence rate than genetic algorithm (GA) for DAG scheduling on heterogeneous systems. However, the rate of convergence of DMSCRO as a metaheuristic method is still defective. This paper proposes a

```
//PHASE 1: Find the constrained critical paths (CCPs)
Find set of critical paths CP according to the description in the second paragraph of Section 3.1.
$j = 1$
for $i = 1$ to $|\mathrm{CP}|$ do
    while there exist ready nodes in $\mathrm{CP}_i$ do
        Insert ready node $v_k$ into constrained critical path Queue ($Q_j$).
    end while
    $j \leftarrow j + 1$
    $i \leftarrow i\% |\mathrm{CP}|$
end for
//PHASE 2: Assign and schedule tasks
for $j = \{1, 2, \ldots, |Q|\}$ do
    for each processor $P_r \in P$ do
        for each node $w \in Q_j$ do
            Find the start time of node $k$, which is the predecessor of $w$
              $\mathrm{ST}_{P_r}(w, k) = \max\left(\left(\left(\mathrm{AEFT}_k\right) + \mathrm{CM}\left(w, P_r, k, P_x\right)\right), \mathrm{AT}_{P_r}\right)$
            Find the finish time of the node
              $\mathrm{EFT}_{P_r}(w) = \max\left(\mathrm{ST}_{P_r}(w, k)\right)_{\forall k\in\mathrm{Pred}(w)} + \mathrm{EC}_{P_r}(w)$
        end for
        Find the finish time of the CCP $Q_j$
          $\mathrm{CEFT}_{P_r}(Q_j) = \max\left(\left(\mathrm{EFT}_{P_r}(w)\right)_{\forall w\in Q_j}\right)$
    end for
    Assign the processor to CCP $Q_j$ which minimizes $\mathrm{CEFT}_{P_r}(Q_j)$.
    Let $P_x$ be assigned, update $\mathrm{AEFT}_w$ of each task $w$ in $Q_j$
    $(\mathrm{AEFT}_w)_{\forall w\in Q_j} = (\mathrm{EFT}_{P_x}(w))_{\forall w\in Q_j}$
end for
```

Algorithm 1: CEFT.

```
for each $E_i = (ev_s, ev_e, w_{s,e})$ in $E$
    $\mathrm{CCP}_s$ = BelongCCP ($ev_s$);
    $\mathrm{CCP}_e$ = BelongCCP ($ev_e$);
    if $(\mathrm{CCP}_s \neq \mathrm{CCP}_e)$ & $(\mathrm{CCPE}(\mathrm{CCP}_s, \mathrm{CCP}_e))$ does not exist
        create CCPE $(\mathrm{CCP}_s, \mathrm{CCP}_e)$
    end if
    add Start and End
    add edges among Start and CCP nodes
    add edges among End and CCP nodes
end for
```

Algorithm 2: Gen_CCPDAG(DAG, CCP) generating CCPDAG.

new CRO-based algorithm, tuple molecular structure-based CRO (TMSCRO), for the mentioned problem, encoding the two basic components of DAG scheduling, module execution order and module-to-processor mapping, into an array of tuples. Combining this kind of molecular structure with the elementary reaction operator designed in TMSCRO has a better capability of intensification and diversification than DMSCRO. Moreover, in TMSCRO, the concept of constrained critical paths (CCPs) [5] and constrained-critical-path directed acyclic graph (CCPDAG) are applied to creating initial population in order to speed up the convergence of TMSCRO. In addition, the first initial molecule, InitS, is also considered to be a super molecule [6] for accelerating convergence, which is converted from the scheduling result of the algorithm constrained earliest finish time (CEFT).

In theory, a metaheuristic method will gradually approach the optimal result if it runs for long enough, based on No-Free-Lunch Theorem, which means the performances of the search for optimal solution of each metaheuristic algorithm are alike when averaged over all possible fitness functions. We have conducted the simulation experiments over the graphs abstracted from two well-known real applications: Gaussian elimination and molecular dynamics application and also a large set of randomly generated graphs. The experiment results show that the proposed TMSCRO can achieve similar performance as DMSCRO

```
InitS = ConvertMole(InitCCPS);
update each $f_i$ in molecule InitS as defined in the last paragraph of Section 5.1.1
MoleN = 1;
while MoleN ≤ PopSize − 1 do
    for each $\text{CCP}_i$ in CCP molecule CCPS
        find the first successor Succ($i$) in CCPDAG from $i$ to the end;
        for each $\text{CCP}_j$, $j \in (i, \text{Succ}(i))$
            find the first predecessor Pred($j$) from Succ($i$) to the begin in CCP molecule CCPS;
            if Pred($j$) < $i$
                interchanged position of $(\text{CCP}_i, \text{sp}_i)$ and $(\text{CCP}_j, \text{sp}_j)$ in CCP molecule CCPS;
            end if
        end for
    end for
    Generate a new CCP molecule CCPS′;
    $S$ = ConvertMole(CCPS′)
    update each $f_i$ in reaction molecule $S$ as defined in the last paragraph of Section 5.1.1
    MoleN ← MoleN + 1;
end while
```

ALGORITHM 3: InitTMolecule(InitCCPS) generating the initial population.

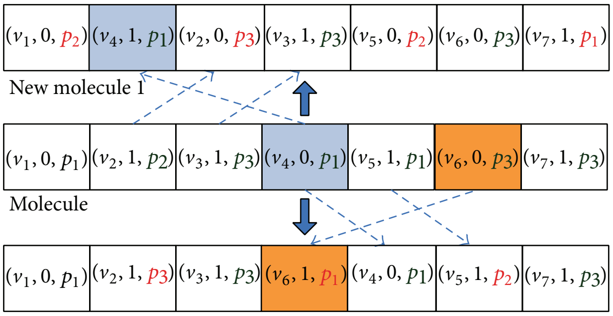


FIGURE 6: Illustration of molecular structure change for decomposition.

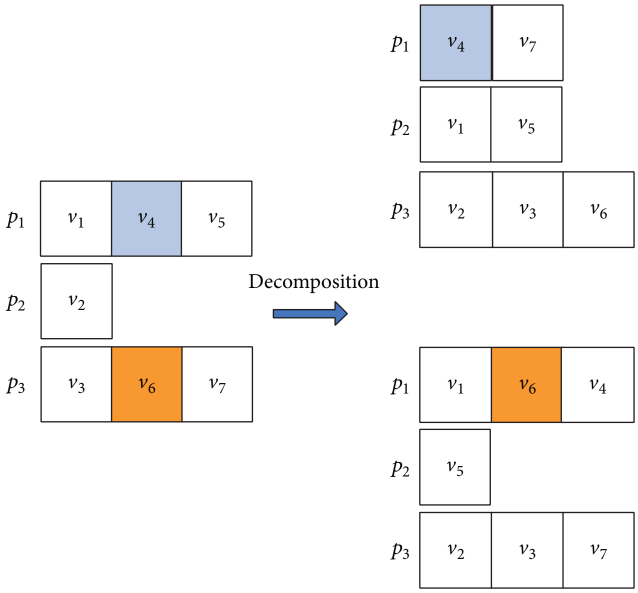


FIGURE 7: Illustration of the task-to-computing-node mapping for decomposition.

in the literature in terms of makespan and outperforms the heuristic algorithms.

There are three major contributions of this work.

(1) Developing TMSCRO based on CRO framework by designing a more reasonable molecule encoding method and elementary chemical reaction operators on intensification and diversification search than DMSCRO.

(2) For accelerating convergence, applying CEFT and CCPDAG to the data pretreatment, utilizing the concept of CCPs in the initialization, and using the first initial molecule, InitS, to be a super molecule in TMSCRO.

(3) Verifying the effectiveness and efficiency of the proposed TMSCRO by simulation experiments. The simulation results of this paper show that TMSCRO is able to approach similar makespan as DMSCRO, but it finds good solutions faster than DMSCRO by 12.89% on average (by 26.29% in the best case).

2. Related Work

Most of the scheduling algorithms can be categorized into heuristic scheduling (including list scheduling, duplication-based scheduling, and cluster scheduling) and metaheuristic (i.e., guided-random-search-based) scheduling. These strategies are to generate the scheduling solution before the execution of the application. The approaches adopted by these different scheduling strategies are summarized in this section.

2.1. Heuristic Scheduling. Heuristic methods usually provide near-optimal solutions for a task scheduling problem in less

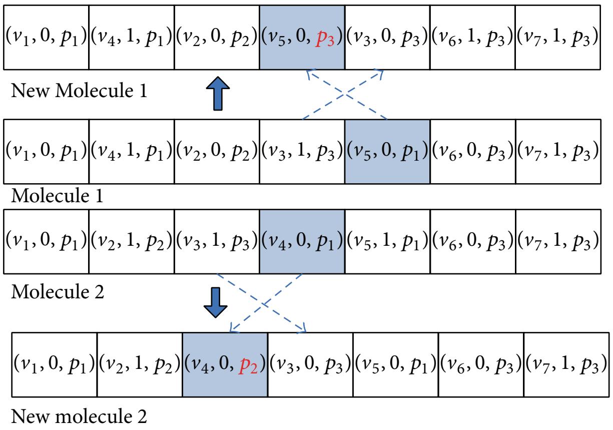

$(v_1,0,p_1)(v_4,1,p_1)(v_2,0,p_2)(v_5,0,p_3)(v_3,0,p_3)(v_6,1,p_3)(v_7,1,p_3)$
New Molecule 1
$(v_1,0,p_1)(v_4,1,p_1)(v_2,0,p_2)(v_3,1,p_3)(v_5,0,p_1)(v_6,0,p_3)(v_7,1,p_3)$
Molecule 1
$(v_1,0,p_1)(v_2,1,p_2)(v_3,1,p_3)(v_4,0,p_1)(v_5,1,p_1)(v_6,0,p_3)(v_7,1,p_3)$
Molecule 2
$(v_1,0,p_1)(v_2,1,p_2)(v_4,0,p_2)(v_3,0,p_3)(v_5,0,p_1)(v_6,0,p_3)(v_7,1,p_3)$
New molecule 2

Figure 8: Illustration of molecular structure change for intermolecular ineffective collision.

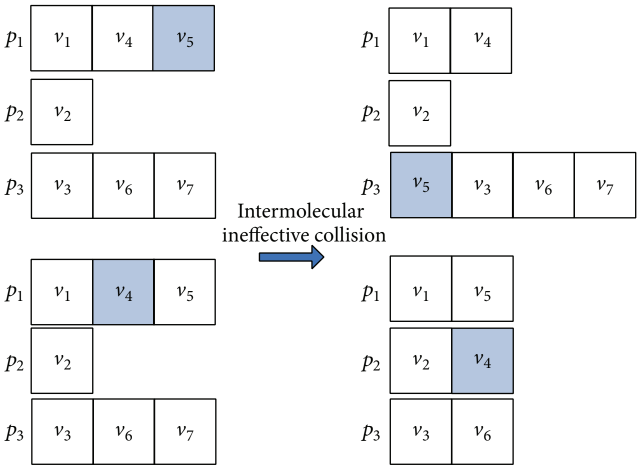


Figure 9: Illustration of the task-to-computing-node mapping for intermolecular ineffective collision.

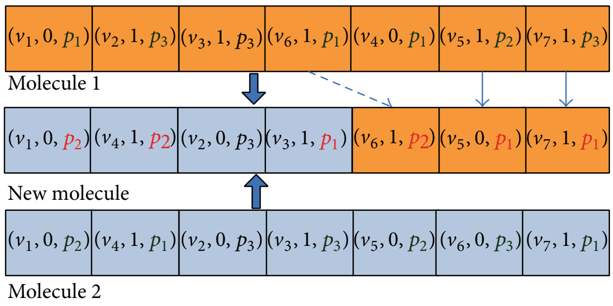

$(v_1,0,p_1)(v_2,1,p_3)(v_3,1,p_3)(v_6,1,p_1)(v_4,0,p_1)(v_5,1,p_2)(v_7,1,p_3)$
Molecule 1
$(v_1,0,p_2)(v_4,1,p_2)(v_2,0,p_3)(v_3,1,p_1)(v_6,1,p_2)(v_5,0,p_1)(v_7,1,p_1)$
New molecule
$(v_1,0,p_2)(v_4,1,p_1)(v_2,0,p_3)(v_3,1,p_3)(v_5,0,p_2)(v_6,0,p_3)(v_7,1,p_1)$
Molecule 2

Figure 10: Illustration of molecular structure change for synthesis.

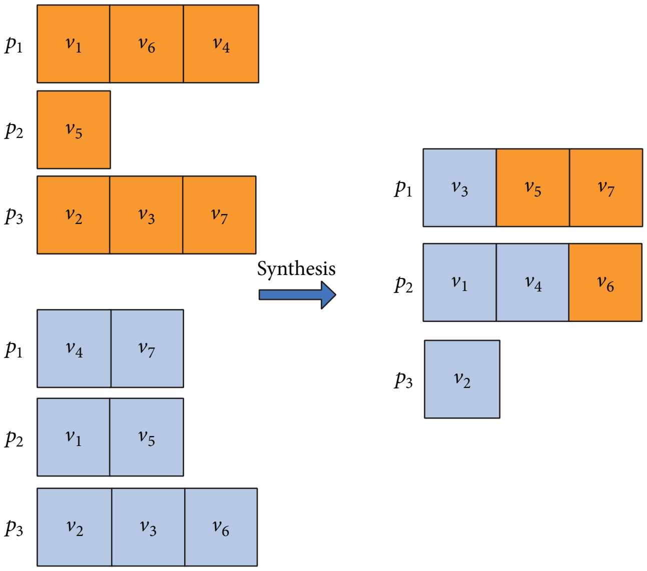


Figure 11: Illustration of the task-to-computing-node mapping for synthesis.

than polynomial time. The approaches adopted by heuristic method search only one path in the solution space, ignoring other possible ones [7]. Three typical kinds of algorithms based on heuristic scheduling for the DAG scheduling problem are discussed as below, such as list scheduling [7, 8], cluster scheduling [9, 10], and duplication-based scheduling [11, 12].

The list scheduling [7, 13–21] generates a schedule solution in two primary phases. In phase 1, all the tasks are processed in a sequence order by their assigned priorities, which are normally based on the task execution and communication costs. There are two attributes used in most list scheduling algorithms, such as b-level and t-level, to assign task priorities. In a DAG, b-level of a node (task) is the length of the longest path from the end node to the node; however, t-level of a node is the length of the longest path from the entry node to the node. In phase 2, the processors are assigned to each task in the sequence.

The heterogeneous earliest finish time (HEFT) scheduling algorithm [16] assigns the scheduling task priorities based on the earliest start time of each task. HEFT allocates a task to the processor which minimizes the task's start time.

The modified critical path (MCP) scheduling [22] considers only one CP (critical path) of the DAG and assigns the scheduling priority to tasks based on their latest start time. The latest start times of the CP tasks are equal to their t-levels. MCP allocates a task to the processor which minimizes the task's start time.

Dynamic-level scheduling (DLS) [23] uses the concept of the dynamic level, which is the difference between the b-level and earliest start time of a task on a processor. Each time the (task, processor) pair with the largest dynamic-level value is chosen by DLS during the task scheduling.

Mapping heuristic (MH) [24] assigns the task scheduling priorities based on the static b-level of each task, which is the b-level without the communication costs between tasks. Then, a task is allocated to the processor which gives the earliest start time.

Levelized-min time (LMT) [17] assigns the task scheduling priority in two steps. Firstly, it groups the tasks into different levels based on the topology of the DAG, and then in each level, the task with the highest priority is the one with

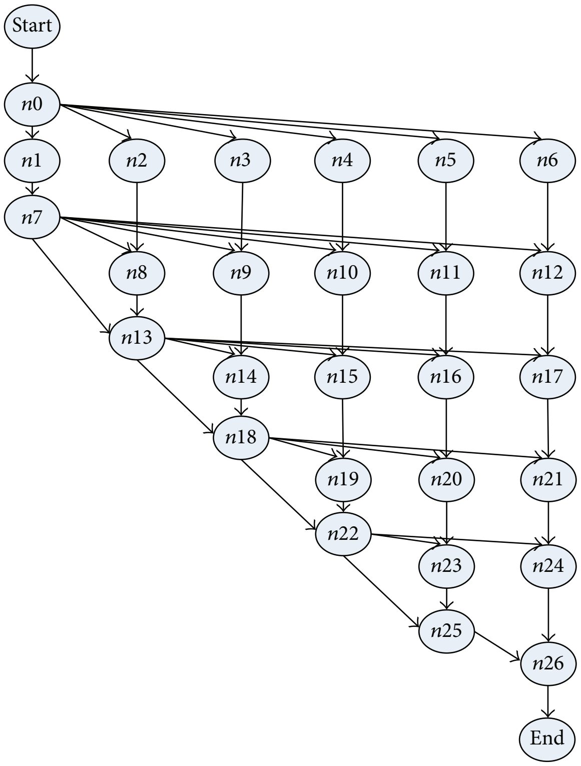


Figure 12: Gaussian elimination for a matrix of size 7.

the largest execution cost. A task is allocated to the processor which minimizes the sum of the total communication costs with the tasks in the previous level and the task's execution cost.

There are two heuristic algorithms for DAG scheduling on heterogeneous systems proposed in [8]. One algorithm named HEFT_T uses the sum of *t*-level and *b*-level to assign the priority to each task. In HEFT_T, the critical tasks are attempted to be on the same processor, and the other tasks are allocated to the processor that gives earliest start time. The other algorithm named HEFT_B applies the concept of *b*-level to assign the priority (i.e., scheduling order) to each task. After the priority assignment, a task is allocated to the processor that minimizes the start time. The extensive experiment results in [8] demonstrate that HEFT_B and HEFT_T outperform (in terms of makespan) other representative heuristic algorithms in heterogeneous systems, such as DLS, MH, and LMT.

Comparing with the list scheduling algorithms, the duplication-based algorithms [23, 25–29] attempt to duplicate the tasks to the same processor on heterogeneous systems, because the duplication may eliminate the communication cost of these tasks and it may effectively reduce the total schedule length.

The clustering algorithms [8, 11, 30–32] regard task collections as clusters to be mapped to appropriate processors. These algorithms are mostly used in the homogeneous systems with unbounded number of processors and they will use as many processors as possible to reduce the schedule length. Then, if the number of the processors used for scheduling is

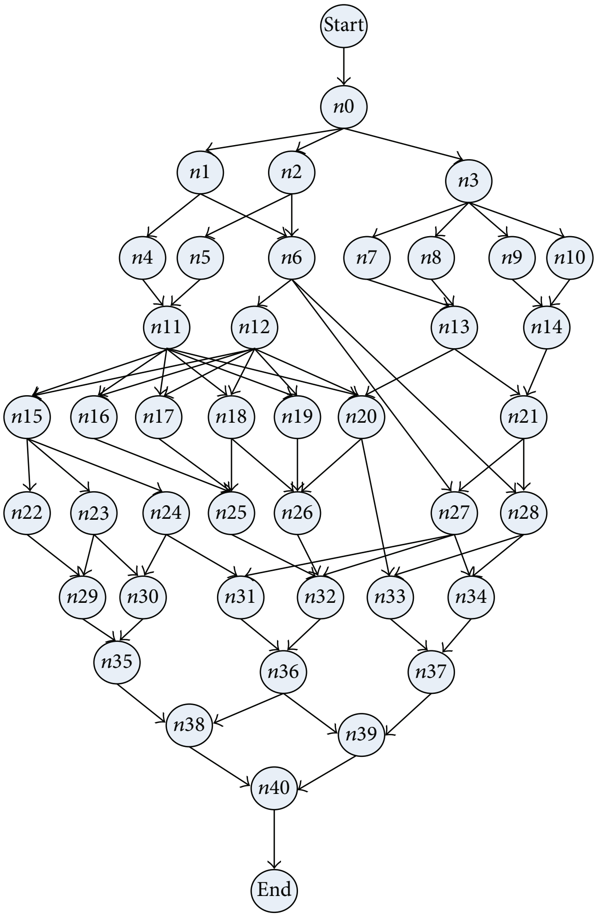


Figure 13: A molecular dynamics code.

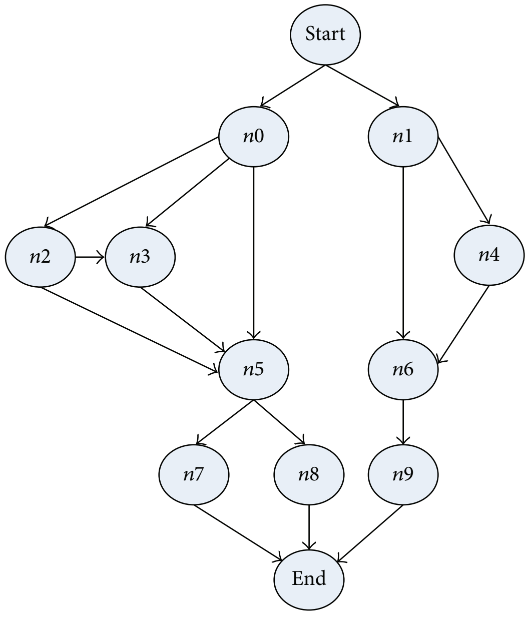


Figure 14: A random graph with 10 nodes.

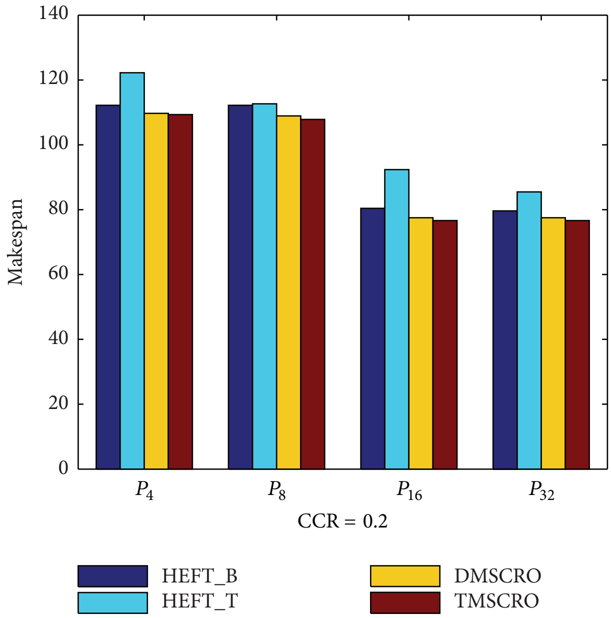


FIGURE 15: Average makespan for Gaussian elimination.

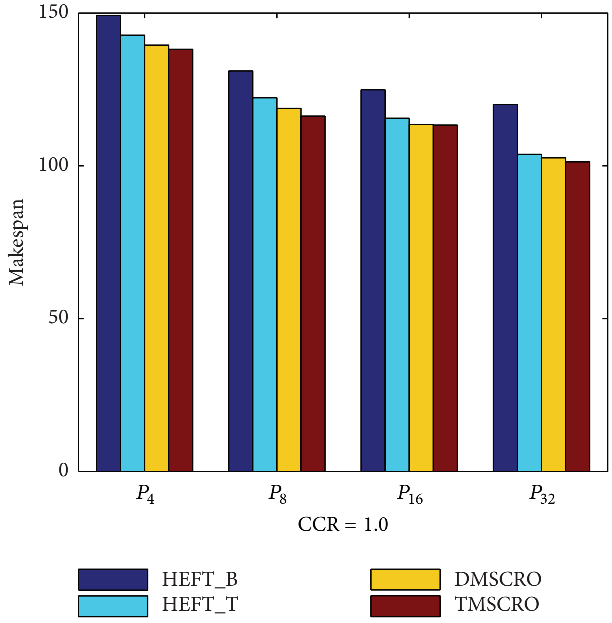


FIGURE 17: Average makespan for the molecular dynamics code.

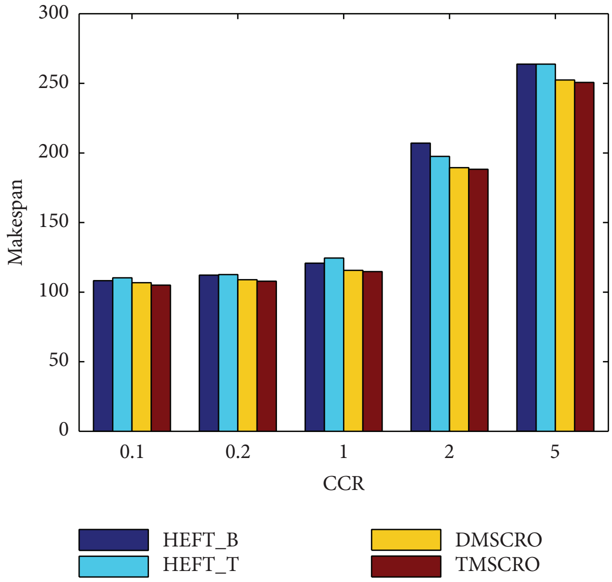


FIGURE 16: Average makespan for Gaussian elimination; the number of processors is 8.

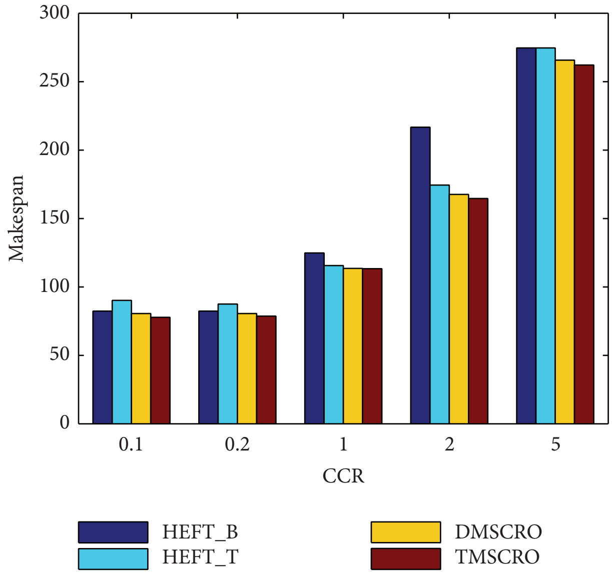


FIGURE 18: Average makespan for the molecular dynamics code; the number of processors is 16.

more than that of the available processors, the task collections (clusters) are processed further to fit in with a limited number of processors.

2.2. Metaheuristic Scheduling. In comparison with the algorithms based on heuristic scheduling, the metaheuristic (guided-random-search-based) algorithms use a combinatorial process for solution searching. In general, with robust performance on many kinds of scheduling problems, the metaheuristic algorithms need sampling candidate solutions in the search space, sufficiently. Many metaheuristic algorithms have been applied to solve the task scheduling problem successfully, such as GA, chemical reaction optimization (CRO), energy-efficient stochastic [33], and so forth.

GA [15, 31, 34–36] is the mostly used metaheuristic method for DAG scheduling. In [15], a solution for scheduling is encoded as one-dimensional string representing an ordered list of tasks to be allocated to a processor. In each string of two parent solutions, the crossover operator selects a crossover point randomly and then merges the head portion of one parent with the tail portion of the other. Mutation operator

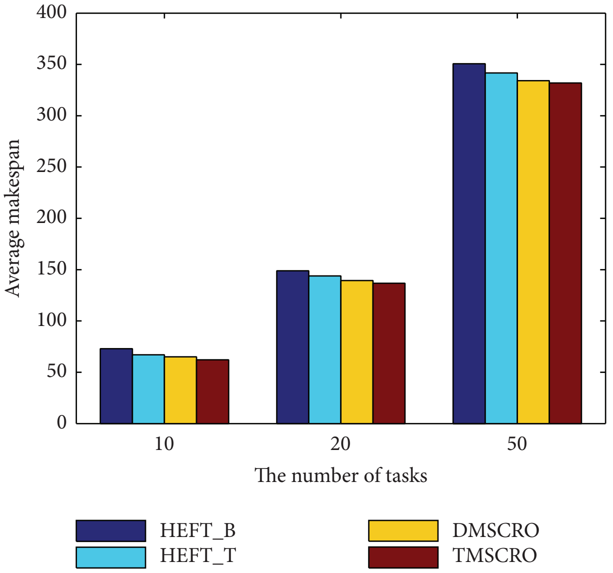


Figure 19: Average makespan of different task numbers, CCR = 10; the number of processors is 32.

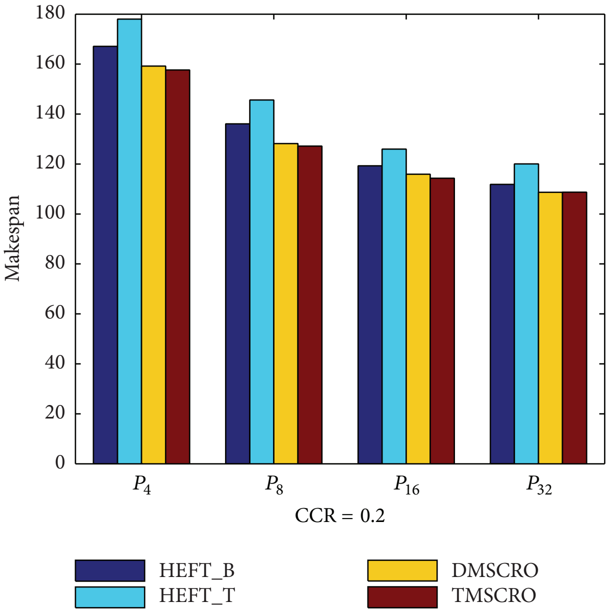


Figure 20: Average makespan of four algorithms under different processor numbers and the low communication costs; the number of tasks is 50.

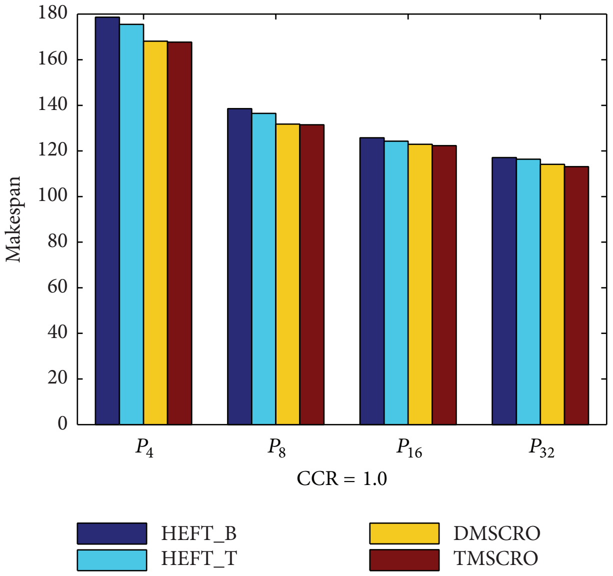


Figure 21: Average makespan of four algorithms under different processor numbers and the low communication costs; the number of tasks is 50.

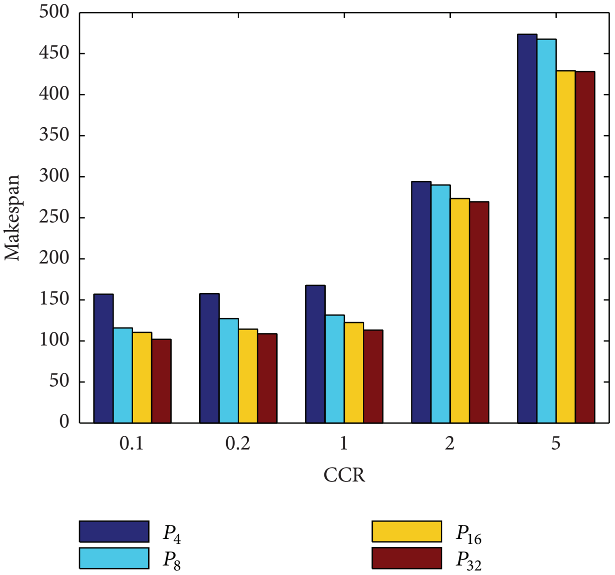


Figure 22: Average makespan of TMSCRO under different values of CCR; the number of tasks is 50.

exchanges two tasks in two solutions, randomly. The concept of makespan is used to evaluate the scheduling solution quality by fitness function.

Chemical reaction optimization (CRO) was proposed very recently [20, 30, 37–39]. It mimics the interactions of molecules in chemical reactions. CRO has good performance already in solving many problems, such as quadratic assignment problem (QAP), resource-constrained project scheduling problem (RCPSP), channel assignment problem (CAP) [39], task scheduling in grid computing (TSGC) [40], and 0-1 knapsack problem (KP01) [41]. So far as we know, double molecular structure-based chemical reaction optimization (DMSCRO) recently proposed in [37] is the only one CRO-based algorithm with two molecular structures for DAG scheduling on heterogeneous systems. CRO-based algorithm (just DMSCRO) mimics the chemical reaction process in a closed container and accords with energy conservation. In DMSCRO, one solution for DAG scheduling including two essential components, task execution order and task-to-processor mapping, corresponds to a double-structured

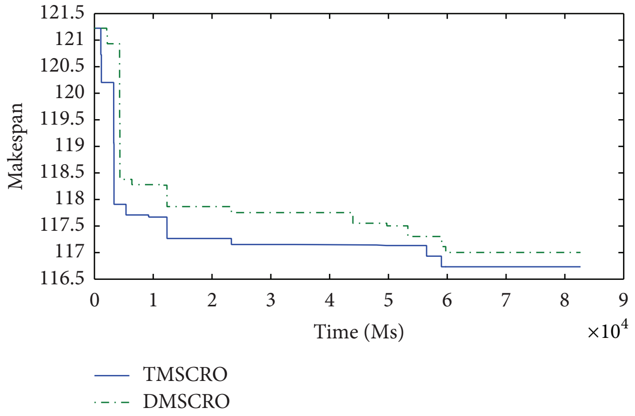


FIGURE 23: The convergence trace for Gaussian elimination; ccr = 0.2; the number of processors is 8.

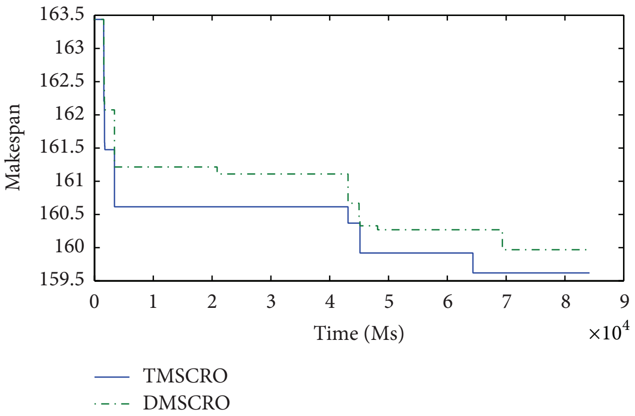


FIGURE 24: The convergence trace for the molecular dynamics code; ccr = 1; the number of processors is 16.

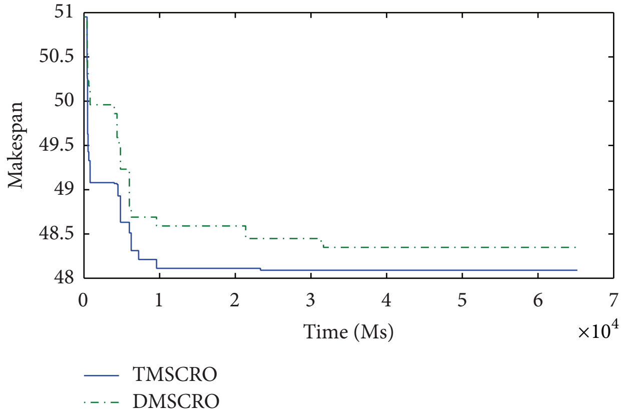


FIGURE 25: The convergence trace for the randomly generated DAGs with each containing 10 tasks.

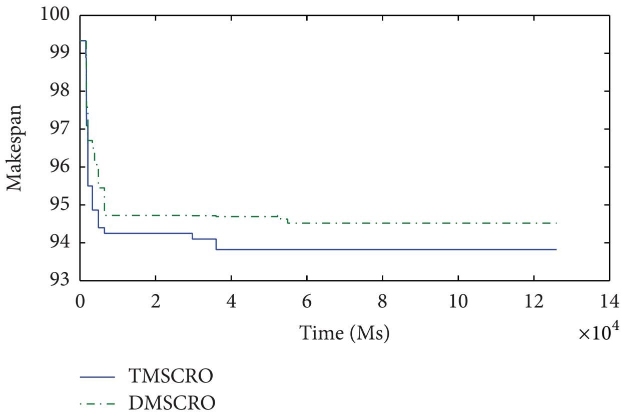


FIGURE 26: The convergence trace for the randomly generated DAGs with each containing 20 tasks.

molecule with two kinds of energy, potential energy (PE) and kinetic energy (KE). The value of PE of a molecule is just the fitness value (objective value), makespan, of the corresponding solution, which can be calculated by the fitness function designed in DMSCRO, and KE with a nonnegative value is to help the molecule escape from local optimums. There are four kinds of elementary reactions used to do the intensification and diversification search in the solution space to find the solution with the minimal makespan, and the principle of the reaction selection is in detail presented in Section 3.2. Moreover, a central buffer is also applied in DMSCRO for energy interchange and conservation during the searching progress. However, as a metaheuristic method for DAG scheduling, DMSCRO still has very large time expenditure and the rate of convergence of this algorithm needs to be improved. Comparing with GA, DMSCRO is similar in model and workload to TMSCRO proposed in this paper.

Our work is concerned with the DAG scheduling problems and the flaw of CRO-based method for DAG scheduling, proposing a tuple molecular structure-based chemical reaction optimization (TMSCRO). Comparing with DMSCRO, TMSCRO applies CEFT [5] to data pretreatment to take the advantage of CCPs as heuristic information for accelerating convergence. Moreover, the molecule structure and elementary reaction operators design in TMSCRO are more reasonable than those in DMSCRO on intensification and diversification of searching the solution space.

3. Background

3.1. CEFT. Constrained earliest finish time (CEFT) based on the constrained critical paths (CCPs) was proposed for heterogeneous system scheduling in [5]. In contrast to other approaches, the CEFT strategy takes account of a broader view of the input DAG. Moreover, the CCPs can be scheduled efficiently because of their static generation.

The constrained critical path (CCP) is a collection with the tasks ready for scheduling only. A task is ready when all its predecessors were processed. In CEFT, a critical path (CP) is generally the longest path from the start node to the end node for scheduling in the DAG. The DAG is initially traversed and critical paths are found. Then it is pruned off the nodes that constitute a critical path. The subsequent traversals

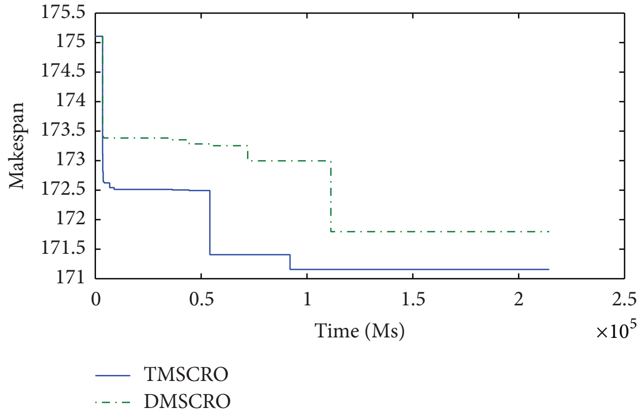


FIGURE 27: The convergence trace for the randomly generated DAGs with each containing 50 tasks.

of the pruned graph produce the remaining critical paths. While the nodes are being removed from the task graph, a pseudo-edge to the start or end node is added if a node has no predecessors or no successors, respectively. The CCPs are subsequently formed by selecting ready nodes in the critical paths in a round-robin fashion. Each CCP may be assigned a single processor which has the minimum finish time of processing all the tasks in the CCP. All the tasks in a CCP not only reduce the communication cost, but also benefit from a broader view of the task graph.

Consider the CEFT algorithm generates schedules for n tasks with $|P|$ heterogeneous processors. Some specific terms and their usage are indicated in Table 1.

The CEFT scheduling approach (Algorithm 1) works in two phases. (1) The critical paths are generated according to the description in the second paragraph of Section 3.1. The critical paths are traversed and the ready nodes are inserted into the constrained critical paths (CCPs) $\text{CCP}_j, \forall j = 1, 2, \ldots, |Q|$. If no more ready nodes are in a critical path, the constrained critical path takes nodes from the next critical path following round-robin traversal of the critical paths. (2) All the CCPs are traversed in order (line 12). Then, $\text{ST}_{P_r}(w, k)$, the maximum of AT_{P_r} and the start time of the predecessors of each node w, is calculated (1). $\text{EFT}_{P_r}(w)$ is computed as the sum of $\text{ST}_{P_r}(w, k)$ and $\text{EC}_{P_r}(w)$ (2). $E_{P_r}(Q_j)$ is the maximum of the finish times of all the CCP nodes on the same processor P_r (3). The processor is then assigned to constrained-critical-path CCP_j which minimizes the $\text{CEFT}_{P_r}(\text{CCP}_j)$ value (line 20). After the actual finish time AEFT_w of each task w in CCP_j is updated, the processor assignment continues iteratively.

3.2. CRO. Chemical reaction optimization (CRO) mimics the process of a chemical reaction where molecules undergo a series of reactions between each other or with the environment in a closed container. The molecules are manipulated agents with a profile of three necessary properties of the molecule, including the following. (1) The molecular structure S: S actually structure represents the positions of atoms in a molecule. Molecular structure can be in the form of a number, a vector, a matrix, or even a graph which is independent of the problem, (2) (Current) potential energy (PE): PE is the objective function value of the current molecular structure ω, that is, $\text{PE}_\omega = f(\omega)$. (3) (Current) kinetic energy (KE): KE is a nonnegative number and it helps the molecule escape from local optimums. There is a central energy buffer implemented in CRO. The energy in CRO may accord with energy conservation and can be exchanged between molecules and the buffer.

Four kinds of elementary reactions may happen in CRO, which are defined as below.

(1) On-wall ineffective collision: on-wall ineffective collision is a unimolecule reaction with only one molecule. In this reaction, a molecule ω is allowed to change to another one ω', if their energy values accord with the following inequality:

$$\text{PE}_\omega + \text{KE}_\omega \geq \text{PE}_{\omega'}; \tag{1}$$

after this reaction, KE will be redistributed in CRO. The redundant energy with the value $\text{KE}_{\omega'} = (\text{PE}_\omega + \text{KE}_\omega - \text{PE}_{\omega'}) \times t$ will be stored in the central energy buffer. Parameter t is a random number from KELossRate to 1 and KELossRate, a system parameter set during the CRO initialization, is the KE loss rate less than 1.

(2) Decomposition: decomposition is the other unimolecule reaction in CRO. A molecule ω may decompose into two new molecules, ω_1' and ω_2', if their energy values accord with inequality (2), in which buf denotes the energy in the buffer, representing the energy interactions between molecules and the central energy buffer:

$$\text{PE}_\omega + \text{KE}_\omega + \text{buf} \geq \text{PE}_{\omega_1'} + \text{PE}_{\omega_2'}; \tag{2}$$

after this reaction, buf is updated by (3) and the KEs of ω_1' and ω_2' are, respectively, computed as (4) and (5), where $\text{Edecomp} = (\text{PE}_\omega + \text{KE}_\omega) - (\text{PE}_{\omega_1'} + \text{PE}_{\omega_2'})$ and $\mu 1$, $\mu 2$, $\mu 3$, $\mu 4$ is a number randomly selected from the range of $[0, 1]$. Consider

$$\text{buf} = \text{Edecomp} + \text{buf} - \left(\text{PE}_{\omega_1'} + \text{PE}_{\omega_2'}\right), \tag{3}$$

$$\text{KE}_{\omega_1'} = \left(\text{Edecomp} + \text{buf}\right) \times \mu 1 \times \mu 2, \tag{4}$$

$$\text{KE}_{\omega_2'} = \left(\text{Edecomp} + \text{buf} - \text{KE}_{\omega_1'}\right) \times \mu 3 \times \mu 4. \tag{5}$$

(3) Intermolecular ineffective collision: intermolecular ineffective collision is an intermolecule reaction with two molecules. Two molecules, ω_1 and ω_2, may change to two new molecules, ω_1' and ω_2', if their energy values accord with the following inequality:

$$\text{PE}_{\omega_1} + \text{PE}_{\omega_2} + \text{KE}_{\omega_1} + \text{KE}_{\omega_2} \geq \text{PE}_{\omega_1'} + \text{PE}_{\omega_2'}; \tag{6}$$

after this reaction, the KEs of ω_1' and ω_2', $\text{KE}_{\omega_1'}$ and $\text{KE}_{\omega_2'}$, will share the spare energy Eintermole calculated by (7). $\text{KE}_{\omega_1'}$ and $\text{KE}_{\omega_2'}$ are computed as (8) and (9), respectively, where $\mu 1$ is a number randomly selected from the range of $[0, 1]$. Consider

$$\text{Eintermole} = \left(\text{PE}_{\omega_1} + \text{PE}_{\omega_2} + \text{KE}_{\omega_1} + \text{KE}_{\omega_2}\right) - \left(\text{PE}_{\omega_1'} + \text{PE}_{\omega_2'}\right), \tag{7}$$

$$\text{KE}_{\omega_1'} = \text{Eintermole} \times \mu 1, \tag{8}$$

$$\text{KE}_{\omega_2'} = \text{Eintermoler} \times \left(1 - \mu 1\right). \tag{9}$$

(4) Synthesis: synthesis is also an intermolecule reaction. Two molecules, ω_1 and ω_2, may be combined to a new molecule, ω', if their energy values accord with inequality (10). The KE of ω' is computed as (11):

$$\text{PE}_{\omega_1} + \text{PE}_{\omega_2} + \text{KE}_{\omega_1} + \text{KE}_{\omega_2} \geq \text{PE}_{\omega'}, \tag{10}$$

$$\text{KE}_{\omega'} = \text{PE}_{\omega_1} + \text{PE}_{\omega_2} + \text{KE}_{\omega_1} + \text{KE}_{\omega_2} - \text{PE}_{\omega'}. \tag{11}$$

The canonical CRO works as follows. Firstly, the initialization of CRO is to set system parameters, such as PopSize (the size of the molecules), KELossRate, InitialKE (the initial energy of molecules), buf (initial energy in the buffer), and MoleColl (MoleColl is a threshold value to determine whether to perform a unimolecule reaction or an intermolecule reaction). Then the CRO processes a loop. In each iteration, whether to perform a unimolecule reaction or an intermolecule reaction is first decided in the following way. A number ε is randomly selected from the range of $[0, 1]$. If ε is bigger than MoleColl, a unimolecule reaction will be chosen, or an intermolecular reaction is to occur. If it is a unimolecular reaction, a parameter θ as a threshold value is used to guide the further choice of on-wall collision or decomposition. NumHit is the parameter used to record the total collision number of a molecule. It will be updated after a molecule undergoes a collision. If the NumHit of a molecule is larger than θ, a decomposition will then be selected. Similarly, a parameter ϑ is used to further decide selection of an intermolecule collision reaction or a synthesis reaction. ϑ specifies the least KE of a molecule. Synthesis reaction will be chosen when both KEs of the molecules ω_1 and ω_2 are less than ϑ, or intermolecular ineffective collision reaction will take place. When the stopping criterion satisfies (e.g., a better solution cannot be found after a certain number of consecutive iterations), the loop will be stopped and the best solution is just the molecule that possesses the lowest PE.

4. Models

This section discusses the system, application, and task scheduling model assumed in this work. The definition of the notations can be found in the Notations section.

4.1. System Model. In this paper, there are multiple heterogeneous processors in the target system, which are presented by $P = \{p_i \mid i = 1, 2, 3, \ldots, |P|\}$. They are fully interconnected with high speed network. Each task in a DAG can only be executed on one processor on heterogeneous system. The edges of the graph are labeled with communication cost that should be taken into account if its start and end tasks are executed on different processors. The communication cost is zero when the same processor is assigned to two communicating modules.

We assume a static computing system model in which the constrained relations and the execution costs of tasks are known a priori and the execution and communication can be performed simultaneously by the processors. In this paper, the heterogeneity is represented by $\text{EC}_{P_r}(w)$, which means the execution cost of a node w using processor P_r. As the assumption of the MHM model, the heterogeneity in the simulations is set as follows to make a processor have different speed for different tasks. The value of each $\text{EC}_{P_r}(w)$ is randomly chosen within the scope of $[1 - g\%, 1 + g\%]$ by using a parameter g $(g \in (0, 1))$. Therefore, the heterogeneity level can be formulated as $(1 + g\%)/(1 - g\%)$. g is set as the value that makes the heterogeneity level 2 in this paper unless otherwise specified.

4.2. Application Model. In DAG scheduling, finding optimal schedules is to find the scheduling solution with the minimum schedule length. The schedule length encompasses the entire execution and communication cost of all the modules and is also termed as makespan. In this paper, the task scheduling problem is to map a set of tasks to a set of processors, aiming at minimizing the makespan. It takes as input a directed acyclic graph DAG $= (V, E)$, with $|V|$ nodes representing tasks, and $|E|$ edges representing constrained relations among the tasks. $V = (v_1, v_2, \ldots, v_i, \ldots, v_{|V|})$ is a node sequence in which the hypothetical entry node (with no predecessors) v_1 and end node (with no successors) $v_{|V|}$, respectively, represent the beginning and the end of execution. The execution cost value of v_i on processor p_k is denoted as $\text{EC}_{p_k}(v_i)$, and the average computation cost of v_i, denoted as $\overline{W(v_i)}$, can be calculated by (12). The parameter for the amounts of computing power available at each node in a heterogeneous system and its heterogeneous level value is given in the 5th paragraph of Section 6 and Table 1.

$E = \{E_i \mid i = 1, 2, 3, \ldots, |E|\}$ is an edge set in which $E_i = (\text{ev}_s, \text{ev}_e, \text{ew}_{s,e})$, with ev_s & $\text{ev}_e \in \{v_1, v_2, \ldots, v_{|V|}\}$ representing its start and end nodes, and the value of communication cost between ev_s and ev_e is denoted as $\text{ew}_{s,e}$. The DAG topology of an exemplar application model and system model is shown in Figures 1 and 2, respectively.

Consider

$$\overline{W(v_i)} = \sum_{k=1}^{|P|} \frac{\text{EC}_{p_k}(v_i)}{|P|}. \tag{12}$$

The constrained-critical-path sequence of DAG $= (V, E)$ is denoted as CCP $= (\text{CCP}_1, \text{CCP}_2, \ldots, \text{CCP}_{|\text{CCP}|})$ with $\text{CCP}_i = (\text{cv}_{i,1}, \text{cv}_{i,2}, \ldots, \text{cv}_{i,|\text{CCP}_i|})$ in which the set $\{\text{cv}_{i,1}, \text{cv}_{i,2}, \ldots, \text{cv}_{i,|\text{CCP}_i|}\} \subseteq \{v_1, v_2, \ldots, v_{|V|}\}$.

The start time of the task v_i on processor p_k is denoted as $\text{ST}_{p_k}(v_i)$, which can be calculated using (13), where $\text{Pred}(v_i)$ is the set of the predecessors of the task v_i. And the earliest finish

```
for $i = 1; i \le |V|; i{+}{+}$
        for each $\mathrm{CCP}_j$ in molecule CCPS
            for each $\mathrm{cv}_k$ in $\mathrm{CCP}_j$
                $v_i = \mathrm{cv}_k$;
                $f_i = 0$;
                $p_i = \mathrm{sp}_j$;
                Generate a new tuple $(v_i, f_i, p_i)$
            end for
        end for
end for
Generate a new reaction molecule $S = ((v_1, f_1, p_1), (v_2, f_2, p_2), \ldots, (v_{|V|}, f_{|V|}, p_{|V|}))$;
for each $(v_i, f_i, p_i)$ in reaction molecule $S$
        find the first successor $\mathrm{Succ}(v_i)$ in DAG from $i$ to the end;
        for each $v_j \in (v_i, \mathrm{Succ}(v_i))$
            find the first predecessor $v_k = \mathrm{Pred}(v_j)$ from $\mathrm{Succ}(v_i)$ to the begin in reaction molecule $S$;
                if $k < i$
                    interchanged position of $(v_i, f_i, p_i)$ and $(v_j, f_j, p_j)$ in reaction molecule $S$;
                end if
        end for
end for
for each $p_i$ in reaction molecule $S$ to randomly change;
        change $p_i$ randomly
end for
return $S$;
```

Algorithm 4: ConvertMole(CCPS) converting a CCPS to an S.

```
slength = 0;
for each node $v$ in $S = ((v_1, f_1, p_1), (v_2, f_2, p_2), \ldots, (v_{|V|}, f_{|V|}, p_{|V|}))$ do
    Calculate the start time of predecessor node $pv$ of $v$
    $\mathrm{ST}_{p_v}(v, pv) = \max\left(\left(\mathrm{EFT}_{pv} + \mathrm{CM}\left(v, p_v, pv, p_{pv}\right)\right), \mathrm{AT}_{P_r}\right)$;
    Find the finish time of $v$
    $\mathrm{EFT}_{p_v}(v) = \max\left(\left(\mathrm{ST}_{p_v}(v, pv)_{\forall pv \in \mathrm{Pred}(v)}\right) + \mathrm{EC}_{p_v}(v)\right)$;
    if slength < $\mathrm{EFT}_{p_v}(v)$
        update scheduling length
        slength = $\mathrm{EFT}_{p_v}(v)$;
    end if
end for
return slength;
```

Algorithm 5: Fit(S) calculating the fitness value of a molecule and the processor allocation optimization.

time of the task v_i on processor p_k is denoted as $\mathrm{EFT}_{p_k}(v_i)$, which can be calculated using (14):

$$\mathrm{ST}_{p_k}(v_i) = \begin{cases} 0, & v_i = v_1 \\ \max\limits_{v_j \in \mathrm{Pred}(v_i)} \mathrm{EFT}_{p_k}(v_i), & p_k = p_m \\ \max\limits_{v_j \in \mathrm{Pred}(v_i)} \left(\mathrm{EFT}_{p_m}(v_j) + \mathrm{ew}_{j,i}\right), & p_k \neq p_m \end{cases} \tag{13}$$

$$\mathrm{EFT}_{p_k}(v_i) = \mathrm{ST}_{p_k}(v_i) + \mathrm{EC}_{p_k}(v_i). \tag{14}$$

The communication to computation ratio (CCR) can be used to indicate whether a DAG is communication intensive or computation intensive. For a given DAG, it is computed by the average communication cost divided by the average computation cost on a target computing system. The computation can be formulated as follows:

$$\mathrm{CCR} = \frac{\sum_{(v_i, v_j, \mathrm{ew}_{i,j}) \in E} \mathrm{ew}_{i,j}}{\overline{W(v_i)}}. \tag{15}$$

5. Design of TMSCRO

TMSCRO mimics the interactions of molecules in chemical reactions with the concepts of molecule, atoms, molecular structure, and energy of a molecule. The structure of a molecule is unique, which represents the atom positions in a molecule. The interactions of molecules in four kinds of basic

```
Initialize PopSize, KELossRate, MoleColl and InitialKE, $\theta$ and $\vartheta$;
Call Algorithm 2 to generate the initial population of TMSCRO, CROPop;
Call Algorithm 3 to calculate PE of each molecule in CROPop;
while the stopping criteria is not met do
    Generate $\varepsilon \in [0, 1]$;
    if $\varepsilon >$ MoleColl
        Select a reaction molecule $S$ from CROPop randomly;
        if $((\mathrm{NumHit}_S - \mathrm{MinHit}_S) > \theta)$ & $(S \neq \mathrm{InitS})$
            Call DecompT to generate new molecules $S_1'$ and $S_2'$;
            Call Algorithm 3 to calculate $\mathrm{PE}_{S_1'}$ and $\mathrm{PE}_{S_2'}$;
            if Inequality (2) holds
                Remove $S$ from CROPop;
                Add $S_1'$ and $S_2'$ to CROPop;
            end if
        else
            Call OnWallT to generate a new molecules $S'$;
            Call Algorithm 3 to calculate $\mathrm{PE}_{S'}$;
            If $(S = \mathrm{InitS})$
                InitS = $S'$;
            end if
            Remove $S$ from CROPop;
            Add $S'$ to CROPop;
        end if
    else
        Select two molecules $S_1$ and $S_2$ from CROPop randomly;
        if $(\mathrm{KE}_{S_1} < \vartheta)$ & $(\mathrm{KE}_{S_2} < \vartheta)$ & $(S_1 \neq \mathrm{InitS})$ & $(S_2 \neq \mathrm{InitS})$
            Call SynthT to generate a new molecule $S'$;
            Call Algorithm 3 to calculate $\mathrm{PE}_{S'}$;
            if Inequality (10) holds
                Remove $S_1$ and $S_2$ from CROPop;
                Add $S'$ to CROPop;
            end if
        else
            Call IntermoleT to generate two new molecules $S_1'$ and $S_2'$;
            Call Algorithm 3 to calculate $\mathrm{PE}_{S_1'}$ and $\mathrm{PE}_{S_2'}$;
            if $(S_1 = \mathrm{InitS})$
                InitS = $S_1'$;
            else if $(S_2 = \mathrm{InitS})$
                InitS = $S_2'$;
            end if
            Remove $S_1$ and $S_2$ from CROPop;
            Add $S_1'$ and $S_2'$ to CROPop;
        end if
    end if
end while
return the molecule with the lowest PE in CROPop;
```

Algorithm 6: TMSCRO(DAG) The TMSCRO outline(framework).

chemical reactions, on-wall ineffective collision, decomposition, intermolecular ineffective collision, and synthesis, aim to transform to the molecule with more stable states which has lower energy. In DAG scheduling, a scheduling solution including a task and processor allocation corresponds to a molecule in TMSCRO. This paper also designs the operators on the encoded scheduling solutions (tuple arrays). These designed operators correspond to the chemical reactions and change the molecular structures. The arrays with different tuples represent different scheduling solutions, and we can calculate the corresponding makespan of the scheduling solution. A scheduling solution makespan corresponds to the energy of a molecule.

In this section, we first present the data pretreatment of the TMSCRO. After the presentation of the encoding of scheduling solutions and the fitness function used in the TMSCRO, we present the design of four elementary chemical reaction operators in each part of the TMSCRO. Finally, we outline the framework of the TMSCRO scheme and discuss a few important properties in TMSCRO.

Table 1: Specific terms and their usage for the CEFT algorithm.

$\mathrm{EC}_{P_r}(w)$	Execution cost of a node w using processor P_r
$\mathrm{CM}(w, P_r, v, P_x)$	Communication cost from node v to w, if P_x has been assigned to node v and P_r is assigned to node w
$\mathrm{ST}_{P_r}(w, v)$	Possible start time of node w which is assigned the processor P_r with the v node being any predecessor of w which has already been scheduled
$\mathrm{EFT}_{P_r}(w)$	Finish time of node w using processor P_r
AEFT_w	Actual finish time of node w
$\mathrm{CEFT}_{P_r}(\mathrm{CCP}_j)$	Finish time of the constrained critical path Q_j when processor P_r is assigned to it
AT_{P_r}	Availability time of P_r
$\mathrm{Pred}(w)$	Set of predecessors of node w
$\mathrm{Succ}(w)$	Set of successors of node w
$\mathrm{AEC}(w)$	Average execution cost of node w

Table 2: CCP corresponding to the DAG as shown in Figure 1(1).

i	CCP_i
1	A-B-D
2	C-G
3	F
4	E
5	H
6	I
7	J

Table 3: Configuration parameters for the simulation of TMSCRO.

Parameter	Value
InitialKE	1000
θ	500
ϑ	10
Buffer	200
KELossRate	0.2
MoleColl	0.2
PopSize	10
g	0.33
Number of runs	50

Table 4: Configuration parameters for the Gaussian elimination graphs.

Parameter	Possible values
CCR	{0.1, 0.2, 1, 2, 5}
Number of processors	{4, 8, 16, 32}
Number of tasks	27

5.1. Molecular Structure, Data Pretreatment, and Fitness Function. This subsection first presents the encoding of scheduling solutions (i.e., the molecular structure) and data pretreatment, respectively. Then we give the statement of the fitness function for optimization designed in TMSCRO.

5.1.1. Molecular Structure and Data Pretreatment. A reasonable initial population in CRO-based methods may increase the scope of searching over the fitness function [20] to support faster convergence and to result in a better solution. Constrained critical paths (CCPs) can be seen as the classification of task sequences constructed by constrained earliest finish time (CEFT) algorithm, which takes into account all factors in DAG (i.e., the average of each task execution cost, the communication costs, and the graph topology). Therefore, TMSCRO utilizes the CCPs to create a reasonable initial population based on a broad view of DAG.

The data pretreatment is to generate the CCPDAG from DAG and to construct CCPS for the initialization of TMSCRO. The CCPDAG is a directed acyclic graph with |CCP| nodes representing constrained critical paths (CCP_s), two virtual nodes (i.e., start and end) representing the beginning and exit of execution, respectively, and |CE| edges representing dependencies among the nodes. The edges of CCPDAG are not labeled with communication overhead which is different from DAG. The data pretreatment includes two steps.

(1) The CCP and the processor allocation of each element of CCP in DAG can be obtained by executing CEFT and the first initial CCP solution, InitCCPS $= ((\mathrm{CCP}_1, \mathrm{sp}_1), (\mathrm{CCP}_2, \mathrm{sp}_2), \ldots, (\mathrm{CCP}_{|\mathrm{CCP}|}, \mathrm{sp}_{|\mathrm{CCP}|}))$, can also be got, in which $((\mathrm{CCP}_i, \mathrm{sp}_i))$ is sorted as the generated order of CCP_i and sp_i is processor assignment of CCP_i after executing CEFT. Consider the graph as shown in Figure 1; the resulting CCPs are indicated in Table 2.

(2) After the execution of CEFT for DAG, the CCPDAG is generated with the input of CCP and DAG. A detailed description is given in Algorithm 2.

As shown in Algorithm 1, the edge E_i of DAG with the start node CCP_s and the end node CCP_e is obtained in each loop (line 1). $\mathrm{BelongCCP}(v_i)$ represents which CCP_j in $\mathrm{CCP}v_i$ belongs to (line 2 and line 3). If CCP_s and CCP_e are different CCPs and there is no edge between them (line 4), then the edge between CCP_s and CCP_e is generated (line 5). Finally, the nodes, start and end, and the edges among them and CCP nodes are added (line 7, line 8, and line 9). Consider the DAG as shown in Figure 1 and the CCP as indicated in Table 1. The resulting CCPDAG is shown in Figure 3.

In this paper, there are two kinds of molecular structures of TMSCRO, CCPS, and S. CCP molecular structure CCPS is just used in the initialization of TMSCRO, which can be formulated as in (16). Whereas the reaction molecular structure S converted from CCPS is used to participate in the elementary reaction of TMSCRO. In CCPS, $((\mathrm{CCP}_i, \mathrm{sp}_i))$s are sorted as the topology of CCPDAG in which CCP_i is constrained critical path (CCP), and sp_i is the processor assigned to CCP_i. $|\mathrm{CCP}| \leq |V|$ because the number of elements in each SCCP_i is greater than or equal to one. A reaction molecule S can be formulated as in (17), which consists of an array of atoms (i.e., tuples) representing a solution of DAG scheduling problem. A tuple includes three integers v_i, f_i, and p_i. The reaction molecular structure S is

Table 5: The experiment results for the Gaussian elimination graph under different processors, CCR = 0.2.

The number of processors	HEFT_B (the average makespan)	HEFT_T (the average makespan)	DMSCRO (the average makespan)	TMSCRO (the average makespan)	TMSCRO (the best makespan)	TMSCRO (the worst makespan)	TMSCRO (the variance of resultant makespans)
4	112.2	122.227	109.9	109.31	109.2	109.9	0.2473
8	112.2	112.648	108.9	107.83	107.1	108.9	0.9613
16	80.4	92.354	77.5	76.62	76.3	78.9	1.6696
32	79.64	85.454	77.5	76.62	76.1	78.9	1.7201

Table 6: The experiment results for the Gaussian elimination graph under different CCRs; the number of processors is 8.

CCR	HEFT_B (the average makespan)	HEFT_T (the average makespan)	DMSCRO (the average makespan)	TMSCRO (the average makespan)	TMSCRO (the best makespan)	TMSCRO (the worst makespan)	TMSCRO (the variance of resultant makespans)
0.1	108.2	110.312	106.78	105.04	104.76	106.6	1.7271
0.2	112.2	112.648	108.9	107.83	107.1	108.9	0.9613
1	120.752	124.536	115.63	114.717	114.3	115.4	0.3787
2	207.055	197.504	189.4	188.303	188.1	188.75	0.1522
5	263.8	263.8	252.39	250.671	250.3	251.79	0.9178

Table 7: Configuration parameters for the molecular dynamics code graphs.

Parameter	Possible values
CCR	{0.1, 0.2, 1, 2, 5}
Number of processors	{4, 8, 16, 32}
Number of tasks	41

encoded with each integer in the permutation representing a task in DAG, the constraint relationship between a tuple and the one before it, and the processor p_i. In each reaction molecular structure S, v_i represents a task in DAG and $(v_1, v_2, \ldots, v_{|V|})$ is a topological sequence of DAG. In S, if v_A of the tuple A, which is before tuple B, is the predecessor of v_B of tuple B in DAG, the second integer of tuple B, f_B, will be 1, or it will be 0. p_i represents the processor allocation of each v_i in the tuple. The sequence of the tuples in a reaction molecular structure S represents the scheduling order of each task in DAG:

$$\mathrm{CCPS} = \left(\left(\mathrm{CCP}_1, \mathrm{sp}_1\right), \left(\mathrm{CCP}_2, \mathrm{sp}_2\right), \ldots, \left(\mathrm{CCP}_{|\mathrm{CCP}|}, \mathrm{sp}_{|\mathrm{CCP}|}\right)\right), \tag{16}$$

$$S = \left(\left(v_1, f_1, p_1\right), \left(v_2, f_2, p_2\right), \ldots, \left(v_{|V|}, f_{|V|}, p_{|V|}\right)\right). \tag{17}$$

5.1.2. Fitness Function. The initial molecule generator is used to generate the initial solutions for TMSCRO to manipulate. The first molecule InitS is converted from InitCCPS. Part three sp_i of each tuple is generated by a random perturbation in the first InitCCPS. A detailed description is given in Algorithms 3 and 4 and presents how to convert a CCPS to an S.

Potential energy (PE) is defined as the objective function (fitness function) value of the corresponding solution represented by S. The overall schedule length of the entire DAG, namely, makespan, is the largest finish time among all tasks, which is equivalent to the actual finish time of the end node in DAG. For the DAG scheduling problem by TMSCRO, the goal is to obtain the scheduling that minimizes makespan and ensure that the precedence of the tasks is not violated. Hence, each fitness function value is defined as

$$\mathrm{PE}_S = \text{makespan} = \mathrm{Fit}(S). \tag{18}$$

Algorithm 5 presents how to calculate the value of the optimization fitness function Fit(S).

5.2. Elementary Chemical Reaction Operators. This subsection presents four elementary chemical reaction operators for sequence optimization and processor allocation optimization designed in TMSCRO, including on-wall collision, decomposition, intermolecular collision, and synthesis.

5.2.1. On-Wall Ineffective Collision. In this paper, the operator, OnWallT, is used to generate a new molecule S' from a given reaction molecule S for optimization. OnWallT works as follows. (1) The operator randomly chooses a tuple (v_i, f_i, p_i) with $f_i = 0$ in S and then exchanges the positions of (v_i, f_i, p_i) and $(v_{i-1}, f_{i-1}, p_{i-1})$. (2) f_{i-1}, f_i and f_{i+1} in S are modified as defined in the last paragraph of Section 5.1.1. (3) The operator changes p_i randomly. In the end, the operator generates a new molecule S' from S as an intensification search. Figures 4 and 5 show the example which is the molecule corresponding to the DAG as shown in Figure 1(2).

Table 8: The experiment results for the molecular dynamics code graph under different processors, CCR = 1.0.

The number of processors	HEFT_B (the average makespan)	HEFT_T (the average makespan)	DMSCRO (the average makespan)	TMSCRO (the average makespan)	TMSCRO (the best makespan)	TMSCRO (the worst makespan)	TMSCRO (the variance of resultant makespans)
4	149.205	142.763	139.51	138.13	137.87	138.6	0.1749
8	131.031	122.265	118.8	116.9	116.2	117.33	0.2764
16	124.868	115.584	113.52	113.36	113.1	113.43	0.0237
32	120.047	103.784	102.617	101.29	101.023	101.47	0.0442

Table 9: The experiment results for the molecular dynamics code graph under different CCRs; the number of processors is 16.

CCR	HEFT_B (the average makespan)	HEFT_T (the average makespan)	DMSCRO (the average makespan)	TMSCRO (the average makespan)	TMSCRO (the best makespan)	TMSCRO (the worst makespan)	TMSCRO (the variance of resultant makespans)
0.1	82.336	90.136	80.53	77.781	77.3	78.9	0.9459
0.2	82.356	87.504	80.53	78.704	78.21	79.13	0.2002
1	124.868	115.584	113.52	113.36	113.1	113.43	0.0237
2	216.735	174.501	167.612	164.7	164.32	164.91	0.0742
5	274.7	274.7	265.8	262.173	262.022	262.6	0.1344

Table 10: Configuration parameters for random graphs.

Parameter	Possible values
CCR	{0.1, 0.2, 1, 2, 5, 10}
Number of processors	{4, 8, 16, 32}
Number of tasks	{10, 20, 50}

5.2.2. Decomposition. In this paper, the operator, DecompT, is used to generate new molecules S_1' and S_2' from a given reaction molecule S. DecompT works as follows. (1) The operator randomly chooses two tuples (tuples) (v_i, f_i, p_i) with $f_i = 0$ and (v_t, f_t, p_t) with $f_t = 0$ in S and then finds the tuple with the first predecessor of (v_i, f_i, p_i), such as (v_j, f_j, p_j), from the selection position to the beginning of reaction molecule S. (2) A random number $k \in [j+1, i-1]$ is generated, and the tuple (v_i, f_i, p_i) is stored in a temporary variable temp, and then from the position $i-1$, the operator shifts each tuple by one place to the right position until a position k. (3) The operator moves the tuple temp to the position k. The rest of the tuples in S_1' are the same as those in S. (4) f_i, f_{i+1} and f_k in S are modified as defined in the last paragraph of Section 5.1.1. (5) The operator generates the other new molecule S_2' as the former steps. The only difference is that, in step 2, we use (v_t, f_t, p_t) instead of (v_i, f_i, p_i). (6) The operator keeps the tuples in S_1', which is at the odd position in S, and retains the tuples in S_2', which is at the even position in S, and then changes the remaining $p_x s$ of tuples in S_1'’ and S_2', randomly. In the end, the operator generates two new molecules S_1' and S_2' from S as a diversification search. Figures 6 and 7 show the example which is the molecule corresponding to the DAG as shown in Figure 1(2).

5.2.3. Intermolecular Ineffective Collision. In this paper, the operator, IntermoleT, is used to generate new molecules S_1' and S_2' from given molecules S_1 and S_2. This operator first uses the steps in OnWallT to generate S_1' from S_1, and then the operator generates the other new molecule S_2' from S_2 in similar fashion. In the end, the operator generates two new molecules S_1' and S_2' from S_1 and S_2 as an intensification search. Figures 8 and 9 show the example which is the molecule corresponding to the DAG as shown in Figure 1(2).

5.2.4. Synthesis. In this paper, the operator, SynthT, is used to generate a new molecule S' from given molecules S_1 and S_2 for optimization. SynthT works as follows. (1) If $|V|$ is plural, then the integer $i = |V|/2$; else $i = (|V| + 1)/2$. (2) S_1 and S_2 are cut off at the position i to become the left and right segments. (3) The left segments of S' are inherited from the left segments of S_1, randomly. (4) Each tuple in the right segments of S' comes from the tuples in S_2 that do not appear in the left segment of S', with their f_x modified as defined in the last paragraph of Section 5.1.1 as well. (5) The operator keeps the tuples in S', which are at the same position in S_1 and S_2 with the same $p_x s$, and then changes the remaining $p_y s$ in S', randomly. As a result, the operator generates S' from S_1 and S_2 as a diversification search. Figures 10 and 11 show the example which is the molecule corresponding to the DAG as shown in Figure 1(2).

5.3. The Framework and Analysis of TMSCRO. The framework of TMSCRO is shown as an outline to schedule a DAG job in Algorithm 6 and the output of Algorithm 6 is just the resultant near-optimal solution for the corresponding DAG scheduling problem. In this framework, TMSCRO first

Table 11: The experiment results for the random graph under different task numbers, CCR = 10; the number of processors is 32.

The number of tasks	TMSCRO (the average makespan)	TMSCRO (the best makespan)	TMSCRO (the worst makespan)	TMSCRO (the variance of resultant makespans)
10	73	67	65.1	62.2
20	148.9	143.9	139.421	136.8
50	350.7	341.7	334.17	331.9

Table 12: The experiment results for the random graph under different processors, CCR = 0.2; the number of tasks is 50.

The number of processors	HEFT_B (the average makespan)	HEFT_T (the average makespan)	DMSCRO (the average makespan)	TMSCRO (the average makespan)	TMSCRO (the best makespan)	TMSCRO (the worst makespan)	TMSCRO (the variance of resultant makespans)
4	167.12	178.023	159.234	157.63	157.12	158.3	0.3923
8	136.088	145.649	128.17	127.178	127.06	127.7	0.1949
16	119.292	125.986	115.9	114.33	114.1	115.2	0.4753
32	111.866	120.065	108.7	108.71	108.31	108.9	0.0733

initializes the process. Then, the process enters a loop. In each iteration, one of the elementary chemical reaction operators for optimization is performed to generate new molecules and PE of newly generated molecules will be calculated. The whole working of TMSCRO for DAG scheduling on heterogeneous problem is as presented in the last paragraph in Section 3.2. However, InitS is considered to be a super molecule [6], so it will be tracked and only participates in on-wall ineffective collision and intermolecular ineffective collision to explore as much as possible the solution space in its neighborhoods and the main purpose is to prevent InitS from changing dramatically. The iteration repeats until the stopping criteria are met. The stopping criteria may be set based on different parameters, such as the maximum amount of CPU time used, the maximum number of iterations performed, an objective function value less than a predefined threshold obtained, and the maximum number of iterations performed without further performance improvement. The stopping criterion of TMSCRO in the experiments of this paper is that the makespan is not changed after 5000 consecutive iterations in each loop. The time complexity of TMSCRO is $O(\text{iters} \times [2 \times (|V|^2 + |E| \times |P|)]$, where iters is the number of iterations in TMSCRO, respectively.

It is very difficult to theoretically prove the optimality of the CRO (as well as DMSCRO and TMSCRO) scheme [37]. However, by analyzing the molecular structure, chemical reaction operators, and the operational environment in TMSCRO, it can be shown to some extent that TMSCRO scheme has the advantage of three points in comparison with GA, SA, and DMSCRO.

First, just like DMSCRO, TMSCRO enjoys the advantages of GA and SA to some extent by analyzing the chemical reaction operators designed in TMSCRO and the operator environment of TMSCRO: (1) the OnWallT and IntermoleT in TMSCRO exchange the partial structure of two different molecules like the crossover operator in GA. (2) The energy conservation requirement in TMSCRO is able to guide the searching of the optimal solution in a similar way as the Metropolis Algorithm of SA guides the evolution of the solutions in SA. Second, constrained earliest finish time (CEFT) algorithm constructs constrained critical paths (CCPs) by taking into account a broader view of the input DAG [5]. TMSCRO applies CEFT and CCPDAG to the data pretreatment and utilizes CCPs in the initialization of TMSCRO to create a more reasonable initial population than DMSCRO for accelerating convergence, because a wide distributed initial population in CRO-based methods may increase the scope of searching over the fitness function [20] to support faster convergence and to result in a better solution. Moreover, to some degree, InitS is also similar to the super molecule in super molecule-based CRO or the "elite" in GA [6]. However, the "elite" in GA is usually generated from two chromosomes, while InitS is based on the whole input DAG by executing CEFT. Third, the operators with the molecular structure in TMSCRO are designed more reasonably than DMSCRO. In CRO-based algorithm, the operators of on-wall collision and intermolecular collision are used for intensifications, while the operators of decomposition and synthesis are for diversifications. The better the operator can get the better the search results of intensification and diversification are. This feature of CRO is very important, which gives CRO more opportunities to jump out of the local optimum and explore the wider areas in the solution space. In TMSCRO, the operators of OnWallT and IntermoleT every time only exchange the positions of one tuple and its former neighbor in the molecule with better capability of intensification on sequence optimization than DMSCRO, of which the reaction operators, OnWall (ω_1) and Intermole (ω_1, ω_2) [37] (ω_1 and ω_2 are big molecules in DMSCRO), may change the task sequence(s) dramatically. Moreover, under the consideration that the optimization includes not only sequence but also processor assignment optimization,

Table 13: The experiment results for the random graph under different processors, CCR = 1.0; the number of tasks is 50.

The number of processors	HEFT_B (the average makespan)	HEFT_T (the average makespan)	DMSCRO (the average makespan)	TMSCRO (the average makespan)	TMSCRO (the best makespan)	TMSCRO (the worst makespan)	TMSCRO (the variance of resultant makespans)
4	178.662	175.52	168.12	167.703	167.42	168	0.0857
8	138.572	136.47	131.8	131.451	131.1	131.9	0.178
16	125.772	124.31	122.91	122.32	122.1	122.432	0.0233
32	117.11	116.4	114.124	113.127	112.9	113.54	0.1348

Table 14: The experiment results for the random graph under different task CCRs, the number of tasks is 50.

CCR	The number of processors is 4	The number of processors is 8	The number of processors is 16	The number of processors is 32
0.1	156.97	115.724	110.3	101.87
0.2	157.63	127.178	114.33	108.71
1	167.703	131.451	122.32	113.127
2	294.042	289.878	273.375	269.514
5	473.5	467.61	429.13	428.13

all reaction operators in TMSCRO can change the processor assignment, but DMSCRO has only two reactions, on-wall and synthesis [37], for processor assignment optimization. On the one hand, TMSCRO has 100% probability of searching the processor assignment solution space by four elementary reactions, with better capability of diversification and intensification on processor assignment optimization than DMSCRO, of which the chance to search this kind of solution space is only 50%. On the other hand, the division of diversification and intensification of four reactions in TMSCRO is very clear; however, this is not in DMSCRO. In each iteration, the diversification and intensification search in TMSCRO have the same probability to be conducted, whereas the possibility of diversification or intensification search in DMSCRO is uncertainty. This design enhances the ability to get better rapidity of convergence and search result in the whole solution space, which is demonstrated by the experimental results in Section 6.3.

6. Simulation and Results

The simulations have been performed to test TMSCRO scheduling algorithm in comparison with heuristic (HEFT_B and HEFT_T) [8] for DAG scheduling and with two metaheuristic algorithms, double molecular structure-based chemical reaction optimization (DMSCRO) [37], by using two sets of graph topology such as the real world application (Gaussian elimination and molecular dynamics code) and randomly generated application. The task graph for Gaussian elimination for input matrix of size 7 is shown in Figure 12, whereas a molecular dynamics code graph is shown in Figure 13. Figure 14 shows a random graph with 10 nodes. The baseline performance is the makespan obtained by DMSCRO.

Considering that HEFT_B and HEFT_T have better performance than other heuristics algorithms for DAG scheduling on heterogeneous computing systems, as proposed in the 8th paragraph in Section 2.1, these two algorithms are used to be the representatives of heuristics in the simulation. There are three reasons why we regard the makespan performance of DMSCRO [37] scheduling as the baseline performance. (1) So far as we know, DMSCRO is the only one CRO-based algorithm for DAG scheduling which takes into account the searching of the task order and processor assignment. (2) As discussed in the 3rd paragraph of Section 2.2, DMSCRO [37] has the closest system model and workload to that of TMSCRO. (3) In [37], CRO-based scheduling algorithm is considered as absorbing the strengths of SA and GA. However, the underlying principles and philosophies of SA are very different from DMSCRO, and because the DMSCRO is also proved to be more effective than genetic algorithm (GA) [15] as presented in [37], we just use DMSCRO to represent the metaheuristic algorithms. We propose to make a comparison between TMSCRO and DMSCRO to validate the advantages of TMSCRO over DMSCRO.

The performance has been evaluated by the parameter makespan. The makespan values plotted in the bar graph of makespan and the chart of converge trace are, respectively, the average result of 50 and 25 independent runs to validate the robustness of TMSCRO. The communication cost is calculated by using computation costs and the computation cost ratio (CCR) values. The computation can be formulated as in (17):

$$\text{Communication Cost} = \text{CCR} * \text{Computation Cost}. \quad (19)$$

All the suggested values for the other parameters of the simulation of TMSCRO and their values are listed in Table 3. These values are proposed in [20].

6.1. Real World Application Graphs. The real world application set is used to evaluate the performance of TMSCRO, which consists of two real world problem graph topologies, Gaussian elimination [22] and molecular dynamics code [19].

6.1.1. Gaussian Elimination. Gaussian elimination is a well-known method to solve a system of linear equations. Gaussian elimination converts a set of linear equations to the upper triangular form by applying elementary row operators on them systematically. As shown in Figure 12, the matrix size of the task graph of Gaussian elimination algorithm is 7, with 27 tasks in total. In [37], this DAG has been used for the simulation of DMSCRO, and we also apply it to the evaluation of TMSCRO in this paper. Under the consideration that graph structure is fixed, the variable parameters are only 22 the communication to computation ratio (CCR) value and the heterogeneous processor number. In the simulation, CCR values were set as 0.1, 0.2, 1, 2, and 5, respectively. Considering the identical operator is executed on each processor and the information communicated between heterogeneous processors is the same in Gaussian elimination, the execution cost of each task is supposed to be the same and all communication links have the same communication cost.

The parameters and their values of the Gaussian elimination graphs performed in the simulation are given in Table 4.

The makespan of TMSCRO, DMSCRO, HEFT_B, and HEFT_T under the increasing processor number is shown in Figure 15. As shown in Figure 15, it can also been seen that as the processor number increases, the average makespan declines, and the advantage of TMSCRO and DMSCRO over HEFT_B and HEFT_T also decreases, because when more computing nodes are contributed to run the same scale of tasks, less intelligent scheduling algorithms are needed in order to achieve good performance.

As the intelligent random search algorithms, TMSCRO and DMSCRO search a wider area of the solution space than HEFT_B, HEFT_T, or other heuristic algorithms, which narrow the search down to a very small portion of the solution space. This is the reason why TMSCRO and DMSCRO are more likely to obtain better solutions and outperform HEFT_B and HEFT_T.

The simulation results show that the performance of TMSCRO and DMSCRO is very similar to the fundamental reason that these algorithms are metaheuristic algorithms. Based on No-Free-Lunch Theorem in the field of metaheuristics, the performances of all well-designed metaheuristic search algorithms for optimal solution are the same, when averaged over all possible objective functions. The optimal solution will be gradually approached by a well-designed metaheuristic algorithm in theory, if it runs for long enough. The DMSCRO developed in [37] is well-designed, and we use it in the simulations of this paper. Therefore similar simulation results of the performances of TMSCRO and DMSCRO indicate that TMSCRO we developed is also well-designed. The detailed experiment result is shown in Table 5.

In Figure 15, the figure shows that TMSCRO is superior to DMSCRO slightly. There will be only one reason for it: the stopping criteria set in this simulation are that the makespan stays unchanged for 5000 consecutive iterations in the search loop. As discussed in the last paragraph of Section 5, all metaheuristic methods that search for optimal solutions are the same in performance when averaged over all possible objective functions. And these experimental stopping criteria make TMSCRO and DMSCRO run for long enough to gradually approach the optimal solution. Moreover, better convergence of TMSCRO makes it more efficient in searching good solutions than DMSCRO by running much less iteration times. More detailed experiment results in this regard will be presented in Section 6.3.

Figure 16 shows that the average makespan of these four algorithms increases rapidly under the CCR increasing. The reason for it is because as CCR increases, the application becomes more communication intensive, making the heterogeneous processors in the idle state for longer. As shown in Figure 16, TMSCRO and DMSCRO outperform HEFT_B and HEFT_T with the advantage being more obvious as CCR becomes larger. These experimental results suggest that, for communication-intensive applications, TMSCRO and DMSCRO can deliver more consistent performance and perform more effectively than heuristic algorithms, HEFT_B and HEFT_T, in a wide range of scenarios for DAG scheduling. The detailed experiment result is shown in Table 6.

6.1.2. Molecular Dynamics Code. Figure 13 shows the DAG of a molecular dynamics code as presented in [19]. As the experiment of Gaussian elimination, the structure of graph and the number of processors are fixed. The varied parameters are the number of heterogeneous processors and the CCR values which are used in our simulation are 0.1, 0.2, 1, 2, and 5.

The parameters and their values of the molecular dynamics code graphs performed in the simulation are given in Table 7.

As shown in Figures 18 and 19, under different heterogeneous processor number and different CCR values, the average makespans of TMSCRO and DMSCRO are over HEFT_B and HEFT_T, respectively. In Figure 17, it can be observed that, with the number of heterogeneous processors increasing, the average makespan decreases. The average makespan with respect to different CCR values is shown in Figure 18. The average makespan increases with the value of CCR increasing. The detailed experiment results are shown in Tables 8 and 9, respectively.

6.2. Random Generated Application Graphs. An effective mechanism to generate random graph for various applications is proposed in [42]. By using the probability for an edge between any two nodes, it can generate a random graph without incline towards a specific topology.

In the random graph generation of this mechanism, the topological order is used to guarantee the precedence constraints; that is, an edge exists between two nodes v_1 and v_2 only if $v_1 < v_2$. For probability pb, $\lfloor |V| * \text{pb} \rfloor$ edges are created from every node m to another node $(N_1 + (1/\text{pb}) * i) \bmod |V|$, where $1 \le i \le \lfloor |V| * \text{pb} \rfloor$, and $\lfloor V \rfloor$ is the total account of task nodes in DAG.

Table 15: Configuration parameters of convergence experiment for the Gaussian elimination graph.

Parameter	Value
CCR	0.2
Number of processors	8
Number of tasks	27

Table 16: Configuration parameters of convergence experiment for the molecular dynamics graph.

Parameter	Value
CCR	1
Number of processors	16
Number of tasks	41

The parameters and their values of the random graphs performed in the simulation are given in Table 10.

Figure 19 shows that TMSCRO always outperforms HEFT_B, HEFT_T, and DMSCRO with the number of tasks in a DAG increasing. The comparison of the average makespan of four algorithms under the increase of heterogeneous processor number is shown in Figures 20 and 21. As can be seen from these figures, the performance of TMSCRO is better than the other three algorithms in all cases. The reasons for these two figures are the same as those explained in Figure 15. The detailed experiment results are shown in Tables 11, 12, and 13, respectively.

As shown in Figure 22, it can be observed that the average makespan approached by TMSCRO increases rapidly with CCR values increasing. This may be because as CCR increases, the application becomes more communication intensive, making the heterogeneous processors in the idle state for longer. The detailed experiment results are shown in Table 14.

6.3. Convergence Trace of TMSCRO. The result of the experiments in the previous subsections is the final makespan obtained by TMSCRO and DMSCRO, showing that TMSCRO can obtain similar makespan performance as DMSCRO. Moreover, in some cases the final makespan achieved by TMSCRO is even better than that by DMSCRO after the stop criteria are satisfied. In this section, the change of makespan in the experiments as TMSCRO and DMSCRO progress during the search is demonstrated by comparing the convergence trace of these two algorithms. These experiments help further reveal the better performance of TMSCRO on convergence and can also help explain why the TMSCRO sometimes outperforms DMSCRO in some cases.

The parameters and their values of the Gaussian elimination, molecular dynamics code, and random graphs performed in the simulation are given in Tables 15, 16, and 17, respectively.

Figures 23 and 24, respectively, plot the convergence traces for processing Gaussian elimination and the molecular dynamics code. Figures 25, 26, and 27 show the convergence traces when processing the sets of randomly generated DAGs and each set contains the DAGs of 10, 20, and 50 tasks, respectively. These figures demonstrated that the makespan performance decreases quickly as both TMSCRO and DMSCRO progress and that the decreasing trends tail off when the algorithms run for long enough. These figures also show that, in most cases, the convergence traces of both algorithms are rather different even though the final makespans obtained by them are almost the same.

Table 17: Configuration parameters of convergence experiment for the random graphs.

Parameter	Values
CCR	{0.2, 1}
Number of processors	{8, 16}
Number of tasks	{10, 20, 50}

Table 18: The results of the statistical analysis over the average coverage rate at different sampling times of all the experiments (the threshold of **P** is set as **0.05**).

DAG	The value of P after Friedman test	Average convergence acceleration ratio
Gaussian elimination	7.10×10^{-8}	4.23%
Molecular dynamics code	2.54×10^{-8}	7.21%
Random graph with 10 tasks	4.26×10^{-8}	23.27%
Random graph with 20 tasks	3.48×10^{-8}	16.41%
Random graph with 50 tasks	2.58×10^{-8}	13.32%

The statistical analysis results over the average coverage rate at 5000 ascending sampling points from start time to end time of all the experiments are shown in Table 18 (the threshold of P is set as 0.05), which are obtained by Friedman test, and each experiment is carried out 25 times. We can find that the differences between two algorithms in performance are significant from a statistical point of view. The reason of it is because the super molecule makes TMSOCRO have a stronger convergence capability, especially early in each run. Moreover, the performance of TMSCRO on convergence is better than DMSCRO. Quantitatively, our records show that TMSCRO converges faster than DMSCRO by 12.89% on average in all the cases (by 23.27% on average in the best case).

In these experiments, the stopping criteria of the algorithms are that the algorithm stops when the makespan performance remains unchanged for a preset number of consecutive iterations in the search loop (in the experiments, it is 5000 iterations). In reality, the algorithms can also stop when the total processing time of it reaches a preset value (e.g., 180s). Moreover, both of TMSCRO and DMSCRO have the same initial population. In this case, the fact that TMSCRO outperforms DMSCRO on convergence means that the makespan achieved by TMSCRO could be much better than that by DMSCRO when the stopping criteria of the algorithm are satisfied. The reason for this can be explained by the analysis presented in the last paragraph of Section 5.3.

7. Conclusion

In this paper, we developed a TMSCRO for DAG scheduling on heterogeneous systems based on chemical reaction optimization (CRO) method. With a more reasonable reaction molecular structure and four designed elementary chemical reaction operators, TMSCRO has a better ability on intensification and diversification search than DMSCRO, which is the only one CRO-based algorithm for DAG scheduling on heterogeneous systems as far as we know. Moreover, in TMSCRO, the algorithm constrained earliest finish time (CEFT) and constrained-critical-path directed acyclic graph (CCPDAG) are applied to the data pretreatment, and the concept of constrained paths (CCPs) is also utilized in the initialization. We also use the first initial molecule, InitS, to be a super molecule for accelerating convergence. As a metaheuristic method, the TMSCRO algorithm can cover a much larger search space than heuristic scheduling approaches. The experiments show that TMSCRO outperforms HEFT_B and HEFT_T and can achieve a higher speedup of task executions than DMSCRO.

In future work, we plan to extend TMSCRO by applying synchronous communication strategy to parallelize the processing of TMSCRO. This kind of design will divide the molecules into groups and each group of molecules is handled by a CPU or GPU. So, multiple groups can be manipulated simultaneously in parallel and molecules can also be exchanged among the CPUs or GPUs from time to time in order to reduce the time cost.

Notations

DAG = (V, E):	Input directed acyclic graph with $\lvert V\rvert$ nodes representing tasks, and $\lvert E\rvert$ edges representing constrained relations among the tasks
$V = (v_1, v_2, \ldots, v_{\lvert V\rvert})$:	Node sequence in which the hypothetical entry node (with no predecessors) v_1 and end node (with no successors) $v_{\lvert V\rvert}$, respectively, represent the beginning and end of execution
$E = \{E_i \mid i = 1, 2, 3, \ldots, \lvert E\rvert\}$:	Edge set in which $E_i = (\mathrm{ev}_s, \mathrm{ev}_e, \mathrm{ew}_{s,e})$, with ev_s & $\mathrm{ev}_e \in \{v_1, v_2, \ldots, v_{\lvert V\rvert}\}$ representing its start and end nodes, and the value of communication cost between ev_s and ev_e denoted as $\mathrm{ew}_{s,e}$
$P = \{p_i \mid i = 1, 2, 3, \ldots, \lvert P\rvert\}$:	Set of multiple heterogeneous processors in target system
$\mathrm{CCP} = (\mathrm{CCP}_1, \mathrm{CCP}_2, \ldots, \mathrm{CCP}_{\lvert\mathrm{CCP}\rvert})$:	Constrained-critical-path sequence of DAG = (V, E)
$\mathrm{CCP}_i = (\mathrm{cv}_{i,1}, \mathrm{cv}_{i,2}, \ldots, \mathrm{cv}_{i,\lvert\mathrm{CCP}_i\rvert})$:	Constrained critical path in which the set $\{\mathrm{cv}_{i,1}, \mathrm{cv}_{i,2}, \ldots, \mathrm{cv}_{i,\lvert\mathrm{CCP}_i\rvert}\} \subseteq \{v_1, v_2, \ldots, v_{\lvert V\rvert}\}$
CCPDAG:	Directed acyclic graph with $\lvert\mathrm{CCP}\rvert$ nodes representing CCPs, two virtual nodes (i.e., start and end) representing the beginning and exit of execution, respectively, and $\lvert\mathrm{CE}\rvert$ edges representing dependencies among all nodes
$\mathrm{CCPS} = ((\mathrm{CCP}_1, \mathrm{sp}_1), (\mathrm{CCP}_2, \mathrm{sp}_2), \ldots, (\mathrm{CCP}_{\lvert\mathrm{CCP}\rvert}, \mathrm{sp}_{\lvert\mathrm{CCP}\rvert}))$:	A CCP molecule used in the initialization of TMSCRO, in which sp_i is the processor assigned to the constrained-critical-path CCP_i
$S = ((v_1, f_1, p_1), (v_2, f_2, p_2), \ldots, (v_{\lvert V\rvert}, f_{\lvert V\rvert}, p_{\lvert V\rvert}))$:	A reaction molecule (i.e., solution) in TMSCRO
(v_i, f_i, p_i):	Atom (i.e., tuple) in S
InitCCPS:	The first CCP molecule for the initialization of TMSCRO
InitS:	The first molecule in TMSCRO
BelongCCP(w):	CCP_i that node w belongs to
CCPE(CCPs, CCP_e):	Edge between CCPs and CCPe
$\overline{W(v)}$:	Average computation cost of node v
$\mathrm{EC}_{P_r}(w)$:	Execution cost of a node w using processor P_r
$\mathrm{CM}(w, P_r, v, P_x)$:	Communication cost from node v to w, if P_x has been assigned to node v and P_r is assigned to node w
$\mathrm{ST}_{P_r}(w, v)$:	Possible start time of node w which is assigned the processor P_r with the v node being any predecessor of w which has already been scheduled
$\mathrm{EFT}_{P_r}(w)$:	Finish time of node w using processor P_r
AT_{P_r}:	Availability time of P_r
Pred(w) :	Set of predecessors of node w
Succ(w) :	Set of successors of node w
CCR :	Communication to computation ratio
g :	The parameter to adjust the heterogeneity level in a heterogeneous system
PE:	Current potential energy of a molecule
KE:	Current kinetic energy of a molecule
InitialKE:	Initial kinetic energy of a molecule
θ:	Threshold value guiding the choice of on-wall collision or decomposition
ϑ:	Threshold value guiding the choice of intermolecule collision or synthesis
Buffer:	Initial energy in the central energy buffer

KELossRate:	Loss rate of kinetic energy
MoleColl:	Threshold value to determine whether to perform a unimolecule reaction or an intermolecule reaction
PopSize:	Size of the molecules
NumHit:	Total collision number of a molecule.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] J. L. R. L. Graham, E. L. Lawler, and A. R. Kan, "Optimization and approximation in deterministic sequencing and scheduling: a survey," *Annals of Discrete Mathematics*, vol. 5, pp. 287–326, 1979.

[2] C. Papadimitriou and M. Yannakakis, "Towards an architecture-independent analysis of parallel algorithms," in *Proceedings of the 20th Annual ACM Symposium on Theory of Computing (STOC '88)*, pp. 510–513, 1988.

[3] V. Sarkar, *Partitioning and Scheduling Parallel Programs for Multiprocessors*, The MIT Press, Cambridge, Mass, USA, 1989.

[4] P. Chrétienne, "Task scheduling with interprocessor communication delays," *European Journal of Operational Research*, vol. 57, no. 3, pp. 348–354, 1992.

[5] M. A. Khan, "Scheduling for heterogeneous systems using constrained critical paths," *Parallel Computing*, vol. 38, no. 4-5, pp. 175–193, 2012.

[6] J. Xu, Y. S. Albert Lam, and O. K. Victor Li, "Stock portfolio selection using chemical reaction optimization," in *Proceedings of the International Conference on Operations Research and Financial Engineering (ICORFE '11)*, pp. 458–463, 2011.

[7] Y.-K. Kwok and I. Ahmad, "Static scheduling algorithms for allocating directed task graphs to multiprocessors," *ACM Computing Surveys*, vol. 31, no. 4, pp. 406–471, 1999.

[8] H. Topcuoglu, S. Hariri, and M.-Y. Wu, "Performance-effective and low-complexity task scheduling for heterogeneous computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, no. 3, pp. 260–274, 2002.

[9] A. Amini, T. Y. Wah, M. R. Saybani, and S. R. A. S. Yazdi, "A study of density-grid based clustering algorithms on data streams," in *Proceedings of the 8th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD '11)*, pp. 1652–1656, Shanghai, China, July 2011.

[10] H. Cheng, "A high efficient task scheduling algorithm based on heterogeneous multi-core processor," in *Proceedings of the 2nd International Workshop on Database Technology and Applications (DBTA '10)*, pp. 1–14, Wuhan, China, November 2010.

[11] T. Tsuchiya, T. Osada, and T. Kikuno, "A new heuristic algorithm based on gas for multiprocessor scheduling with task duplication," in *Proceedings of the 3rd International Conference on Algorithms and Architectures for Parallel Processing (ICAPP '97)*, pp. 295–308, Melbourne, Australia, December 1997.

[12] R. Bajaj and D. P. Agrawal, "Improving scheduling of tasks in a heterogeneous environment," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 2, pp. 107–118, 2004.

[13] H.-W. Ge, L. Sun, Y.-C. Liang, and F. Qian, "An effective PSO and AIS-based hybrid intelligent algorithm for job-shop scheduling," *IEEE Transactions on Systems, Man, and Cybernetics A: Systems and Humans*, vol. 38, no. 2, pp. 358–368, 2008.

[14] N. B. Ho and J. C. Tay, "Solving multiple-objective flexible job shop problems by evolution and local search," *IEEE Transactions on Systems, Man and Cybernetics C: Applications and Reviews*, vol. 38, no. 5, pp. 674–685, 2008.

[15] E. S. H. Hou, N. Ansari, and H. Ren, "Genetic algorithm for multiprocessor scheduling," *IEEE Transactions on Parallel and Distributed Systems*, vol. 5, no. 2, pp. 113–120, 1994.

[16] J.-J. Hwang, Y.-C. Chow, F. D. Anger, and C.-Y. Lee, "Scheduling precedence graphs in systems with interprocessor communication times," *SIAM Journal on Computing*, vol. 18, no. 2, pp. 244–257, 1989.

[17] M. Iverson, F. Özgüner, and G. Follen, "Parallelizing existing applications in a distributed heterogeneous environment," in *Proceedings of the IEEE International Conference on Heterogeneous Computing Workshop (HCW '95)*, pp. 93–100, 1995.

[18] M. H. Kashani and M. Jahanshahi, "Using simulated annealing for task scheduling in distributed systems," in *Proceedings of the International Conference on Computational Intelligence, Modelling, and Simulation (CSSim '09)*, pp. 265–269, Brno, Czech Republic, September 2009.

[19] S. Kim and J. Browne, "A general approach to mapping of parallel computation upon multiprocessor architectures," in *Proceedings of the International Conference on Parallel Processing*, vol. 3, pp. 1–8, 1988.

[20] A. Y. S. Lam and V. O. K. Li, "Chemical-reaction-inspired metaheuristic for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 3, pp. 381–399, 2010.

[21] H. Li, L. Wang, and J. Liu, "Task scheduling of computational grid based on particle swarm algorithm," in *Proceedings of the 3rd International Joint Conference on Computational Sciences and Optimization (CSO '10)*, vol. 2, pp. 332–336, Huangshan, China, May 2010.

[22] M.-Y. Wu and D. D. Gajski, "Hypertool: a programming aid for message-passing systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 1, no. 3, pp. 330–343, 1990.

[23] G. C. Sih and E. A. Lee, "Compile-time scheduling heuristic for interconnection-constrained heterogeneous processor architectures," *IEEE Transactions on Parallel and Distributed Systems*, vol. 4, no. 2, pp. 175–187, 1993.

[24] H. El-Rewini and T. G. Lewis, "Scheduling parallel program tasks onto arbitrary target machines," *Journal of Parallel and Distributed Computing*, vol. 9, no. 2, pp. 138–153, 1990.

[25] F.-T. Lin, "Fuzzy job-shop scheduling based on ranking level (lambda, 1) interval-valued fuzzy numbers," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 4, pp. 510–522, 2002.

[26] B. Liu, L. Wang, and Y.-H. Jin, "An effective PSO-based memetic algorithm for flow shop scheduling," *IEEE Transactions on Systems, Man, and Cybernetics B: Cybernetics*, vol. 37, no. 1, pp. 18–27, 2007.

[27] F. Pop, C. Dobre, and V. Cristea, "Genetic algorithm for DAG scheduling in Grid environments," in *Proceedings of the IEEE 5th International Conference on Intelligent Computer Communication and Processing (ICCP '09)*, pp. 299–305, Cluj-Napoca, Romania, August 2009.

[28] R. Shanmugapriya, S. Padmavathi, and S. M. Shalinie, "Contention awareness in task scheduling using tabu search," in *Proceedings of the IEEE International Advance Computing Conference (IACC '09)*, pp. 272–277, Patiala, India, March 2009.

[29] L. Shi and Y. Pan, "An efficient search method for job-shop scheduling problems," *IEEE Transactions on Automation Science and Engineering*, vol. 2, no. 1, pp. 73–77, 2005.

[30] P. Choudhury, R. Kumar, and P. P. Chakrabarti, "Hybrid scheduling of dynamic task graphs with selective duplication for multiprocessors under memory and time constraints," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 7, pp. 967–980, 2008.

[31] S. Song, K. Hwang, and Y.-K. Kwok, "Risk-resilient heuristics and genetic algorithms for security-assured grid job scheduling," *IEEE Transactions on Computers*, vol. 55, no. 6, pp. 703–719, 2006.

[32] D. P. Spooner, J. Cao, S. A. Jarvis, L. He, and G. R. Nudd, "Performance-aware workflow management for grid computing," *The Computer Journal*, vol. 48, no. 3, pp. 347–357, 2005.

[33] K. Li, X. Tang, and K. Li, "Energy-efficient stochastic task scheduling on heterogeneous computing systems," *IEEE Transactions on Parallel and Distributed Systems*, 2014.

[34] J. Wang, Q. Duan, Y. Jiang, and X. Zhu, "A new algorithm for grid independent task schedule: genetic simulated annealing," in *Proceedings of the World Automation Congress (WAC '10)*, pp. 165–171, Kobe, Japan, September 2010.

[35] L. He, D. Zou, Z. Zhang, C. Chen, H. Jin, and S. Jarvis, "Developing resource consolidation frameworks for moldable virtual machines in clouds," *Future Generation Computer Systems*, vol. 32, pp. 69–81, 2012.

[36] Y. Xu, K. Li, J. Hu, and K. Li, "A genetic algorithm for task scheduling on heterogeneous computing systems using multiple priority queues," *Information Sciences*, vol. 270, pp. 255–287, 2014.

[37] Y. Xu, K. Li, L. He, and T. K. Truonga, "A DAG scheduling scheme on heterogeneous computing systems using double molecular structure-based chemical reaction optimization," *Journal of Parallel and Distributed Computing*, vol. 73, no. 9, pp. 1306–1322, 2013.

[38] J. Xu, A. Lam, and V. Li, "Chemical reaction optimization for the grid scheduling problem," in *Proceedings of the IEEE International Conference on Communications (ICC '10)*, pp. 1–5, Cape Town, South Africa, May 2010.

[39] B. Varghese, G. Mckee, and V. Alexandrov, "Can agent intelligence be used to achieve fault tolerant parallel computing systems?" *Parallel Processing Letters*, vol. 21, no. 4, pp. 379–396, 2011.

[40] J. Xu, A. Lam, and V. Li, "Chemical reaction optimization for task scheduling in grid computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 10, pp. 1624–1631, 2011.

[41] T. K. Truong, K. Li, and Y. Xu, "Chemical reaction optimization with greedy strategy for the 0-1 knapsack problem," *Applied Soft Computing Journal*, vol. 13, no. 4, pp. 1774–1780, 2013.

[42] V. A. F. Almeida, I. M. M. Vasconcelos, J. N. C. Arabe, and D. A. Menasce, "Using random task graphs to investigate the potential benefits of heterogeneity in parallel systems," in *Proceedings of the ACM/IEEE Conference on Supercomputing (Supercomputing '92)*, pp. 683–691, IEEE Computer Society Press, Los Alamitos, Calif, USA, 1992.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 753860, 6 pages
http://dx.doi.org/10.1155/2014/753860

Research Article

Obscenity Detection Using Haar-Like Features and Gentle Adaboost Classifier

Rashed Mustafa,[1,2,3] Yang Min,[4] and Dingju Zhu[1,2,5]

[1] *Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China*
[2] *University of Chinese Academy of Sciences, Beijing 100049, China*
[3] *Department of Computer Science and Engineering, University of Chittagong, Chittagong 4331, Bangladesh*
[4] *Department of Computer Science, The University of Hong Kong, Hong Kong 999077, Hong Kong*
[5] *School of Computer Science, South China Normal University, Guangzhou 510631, China*

Correspondence should be addressed to Dingju Zhu; dj.zhu@siat.ac.cn

Received 31 March 2014; Accepted 9 May 2014; Published 5 June 2014

Academic Editor: Yu-Bo Yuan

Large exposure of skin area of an image is considered obscene. This only fact may lead to many false images having skin-like objects and may not detect those images which have partially exposed skin area but have exposed erotogenic human body parts. This paper presents a novel method for detecting nipples from pornographic image contents. Nipple is considered as an erotogenic organ to identify pornographic contents from images. In this research Gentle Adaboost (GAB) haar-cascade classifier and haar-like features used for ensuring detection accuracy. Skin filter prior to detection made the system more robust. The experiment showed that, considering accuracy, haar-cascade classifier performs well, but in order to satisfy detection time, train-cascade classifier is suitable. To validate the results, we used 1198 positive samples containing nipple objects and 1995 negative images. The detection rates for haar-cascade and train-cascade classifiers are 0.9875 and 0.8429, respectively. The detection time for haar-cascade is 0.162 seconds and is 0.127 seconds for train-cascade classifier.

1. Introduction

Online video and images are now easily accessible due to availability of high-speed Internet and rapid growth of multimedia technology. A report shows that a large number of teens and children search pornographic contents everyday [1]. This is a threat for the society and a concern of Internet safety. Taking care of this issue, scientists are working hard and initiated different filter techniques to screen malicious contents. Most techniques were texts-based and could not identify objectionable materials from the sites appropriately. The reason for this is that there are countless websites which do not contain sensitive texts; hence, content-based image processing especially identifying obscenity has now been a challenging research area. It has been almost two decades when Forsyth et al. [2] published the first paper in this issue on "Finding Naked People." After that, a large number of works were accomplished by different researchers all around the globe [2–4]. The prior works concentrated mainly on skin color, which is not suitable because of skin-like objects and partially exposed images that are not considered obscene.

In this paper we focused on nipple detection for identifying objectionable images from pornographic sites. It is a challenging task because nipples are nonrigid objects varying in shape, size, scale, illumination, and partial occlusion [5]. The appearance also differs due to different ethnicity. Considering the above factors, in this research we extracted haar-like features from some cropped nipple images and used Gentle Adaboost (GAB) haar-cascade classifier for ensuring accuracy; in addition we have compared it with train-cascade classifier in order to satisfy detection time. It has been shown that haar-cascade classifier is suitable for accurately detecting nipples, but for ensuring faster detection and little accuracy train-cascade classifier is better.

The rest of this paper can be organized according to the following ways: in Section 2 some related work will be discussed, some background knowledge including color model, haar-like features, and Gentle Adaboost algorithm has been illustrated in Section 3, experimental setup will be elucidated in Section 4, results will be analyzed in Section 5, and finally a discussion in Section 6 concludes the paper.

2. Literature Review

Content-based image processing for identifying objectionable materials is not a new idea. The first paper was published more than twenty years ago [2]. In the past, research on this ground was followed using skin color model. A large percentage of skin was used as a measure of pornographic contents [2–9]. But due to large varieties of skin-like objects this only technique is not suitable.

There is a suitable idea to find objectionable material which is nipple detection. Nipples are considered erotogenic human body parts and have unique characteristics in all pornographic images. Fuangkhon et al. [5, 10–12] presented an object detection using image processing and neural network entitled "*nipple detection for obscene pictures*." The authors claimed that the detection rate was 65.4%; so far it was the only paper on nipple detection until 2010. In 2010 Wang et al. [9] proposed another robust method entitled "*Automatic Nipple Detection Using Shape and Statistical Skin Color Information*;" in this paper a new approach on nipple detection for adult content recognition has been presented and it combines the advantages of Adaboost algorithm, that is, the rapid speed in object detection and the robustness of nipple features for adaptive nipple detection. The detection rate of this approach was 75.6%. Kejun et al. [7] proposed another method called "*Automatic Nipple Detection Using Cascaded AdaBoost Classifier*." In this research they used extended haar-like features, color features, and texture and shape features to train and obtain cascaded Adaboost classifier. The authors claimed that the detection rate was 90.37%. There are some other methods of nipple detection, but this is limited for digital mammogram. According to the literature, those above-mentioned three works were significant for nipple detection research, which was devoted to identify objectionable materials from images. All works have lacked appropriate quantitative measures to classify whether an image contains nipple objects or not.

3. Background Knowledge

In this section significant skin color model, haar-like features, and Gentle Adaboost algorithm will be discussed.

3.1. Color Model (YCbCr). In this research we used YCbCr color model for skin filtering. It belongs to orthogonal color spaces, which reduce the redundancy present in RGB, and color channels and it represents the color with statistically independent components [6]. The components are luminance and chrominance that are explicitly separated and lead to the suitability of skin color detection. YCbCr can be obtained from RGB color transformation. The color space transformation is assumed to decrease the overlap between skin and nonskin pixels, which in turn makes the process robust thereby aiding skin-pixel classification under a wide range of illumination conditions. YCbCr is an encoded nonlinear RGB, commonly used by European televisions and for image compression. Here, the color is represented by luma (which is luminance or brightness) computed from nonlinear RGB constructed as a weighted sum of the RGB values and two color difference values Cb and Cr that are formed by subtracting the luma value from red and blue components of RGB model. The following equations are the transformation from RGB to YcbCr [2–5]:

$$\begin{gathered} Y = 0.299R + 0.587G + 0.114B, \\ Cb = R - Y, \\ Cr = B - Y, \\ [Y \ \ Cb \ \ Cr] = [R \ \ G \ \ B] \begin{bmatrix} 0.299 & -0.168935 & 0.499813 \\ 0.587 & -0.331665 & -0.418531 \\ 0.114 & 0.50059 & -0.081282 \end{bmatrix}. \end{gathered} \tag{1}$$

This model is suitable for use under some predefined conditions within specific systems. The Y component describes brightness and the other two values describe a color difference rather than a color itself, making the color space unintuitive. The transformation simplicity and explicit separation of luminance and chrominance components make this color space perfect for skin color modeling. In YCbCr the RGB components are separated into luminance (Y), chrominance blue (Cb), and chrominance red (Cr). And thus YCbCr space is one of the most popular selections for skin detection and has been used by many researchers [6, 8, 13].

3.2. Haar-Like Features. Haar-like features are applicable to classify generic objects. They are particularly familiar for face detection, where the system determines whether an object is a generic face. Simply knowing that an object is a face is useful for segmenting the image, narrowing down a region of interest, or simply doing some other fun tricks [14, 15]. Technically, haar-like features refer to a way of slicing and dicing an image to identify the key patterns. The template information is stored in a file known as a haar-cascade, usually formatted as an XML file [14]. This requires a fair amount of work to train a classifier system and generate the cascade file. Some simple haar-like features are described in Figure 1.

The calculation method of haar-like features is faster by introducing integral image or summed area table [16]. This is the reason that haar-cascade and train-cascade classifiers are computing features very quickly.

3.2.1. Integral Image. Rectangular two-dimensional image features can be computed rapidly using an intermediate representation called the integral image [17]. The integral image, denoted by $ii(x, y)$, at location (x, y) (2)-(3) contains the sum of the pixel values above and to the left of (x, y) (Figure 2). The value of the integral image at point $(x; y)$ is

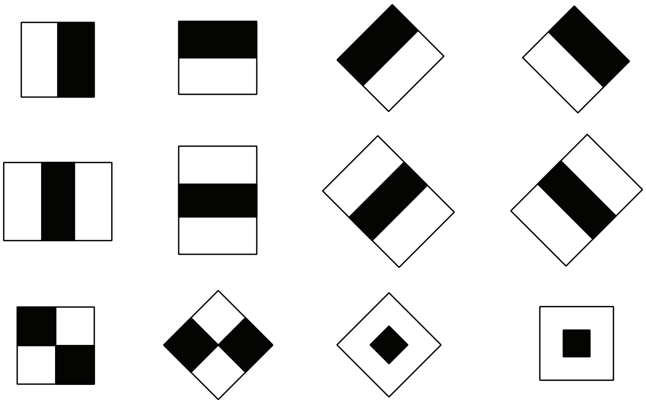


Figure 1: Simple haar-like features.

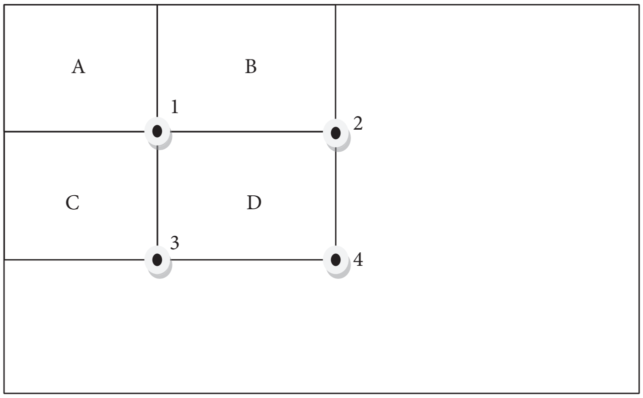


Figure 2: Calculation of summed area table.

the sum of all the pixels above and to the left. Consider the following:

$$ii(x, y) = \sum_{x' \le x, y' \le y} i(x', y'), \tag{2}$$

where $ii(x; y)$ is the integral image and $i(x; y)$ is the original image using the following pair of recurrences:

$$\begin{aligned} s(x, y) &= s(x, y-1) + i(x, y), \\ ii(x, y) &= ii(x-1, y) + s(x, y). \end{aligned} \tag{3}$$

The integral image can be computed in one pass over the original image.

Figure 2 demonstrates the calculation method of summed area table. This is the reason that Adaboost calculates feature using this technique. For example by using only four array references, the sum of the pixels within rectangle D can be calculated according to the following way:

at location 1 (sum of the pixels in rectangle A);

at location 2 (A + B);

at location 3 (A + C);

at location 4 (A + B + C + D);

finally the sum within rectangle D is 4 + 1 − (2 + 3).

3.2.2. Gentle Adaboost Algorithm (GAB). In this research we used Gentle Adaboost algorithm (GAB) [1, 12] to train a number of haar-like features (over 85000) using haar-cascade and train-cascade methodologies. Among four different types of Adaboost algorithm, in real Adaboost algorithm, logarithm of the sample's posterior probability is applied to check the competent weak classifier, which will greatly boost the weight of "noise" in the training set. But, "noise" samples are difficult to be completely eliminated, which leads to overfitting during training stage. As a result, the node classifier's generalization ability will be weakened. In order to improve the node classifier's generalization ability, Gentle Adaboost has been utilized in [1]. The pseudocode of the algorithm is as follows.

(a) Let $(x_1, y_1) \cdots (x_n, y_n)$ be example images where $y_i = -1, 1$ for negative and positive examples accordingly.

(b) Now the weights needed to be initialized:

$w_{1,i} = 1/2p, 1/2q$ for $y_i = -1, 1$; accordingly p and q are the numbers of negatives and positives.

(c) For $t = 1 \cdots T$, consider the following.

(1) Weights normalization is

$$w_{l,i} \longleftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}. \tag{4}$$

(2) For each feature j, train a classifier h_j which is limited to use a single feature. The error is evaluated with respect to w_i, $\epsilon_j = \sum_i w_j |h_j(x_i) - y_i)|$.

(3) Classifier (h_t) should be chosen with minimum error rate ϵ_t.

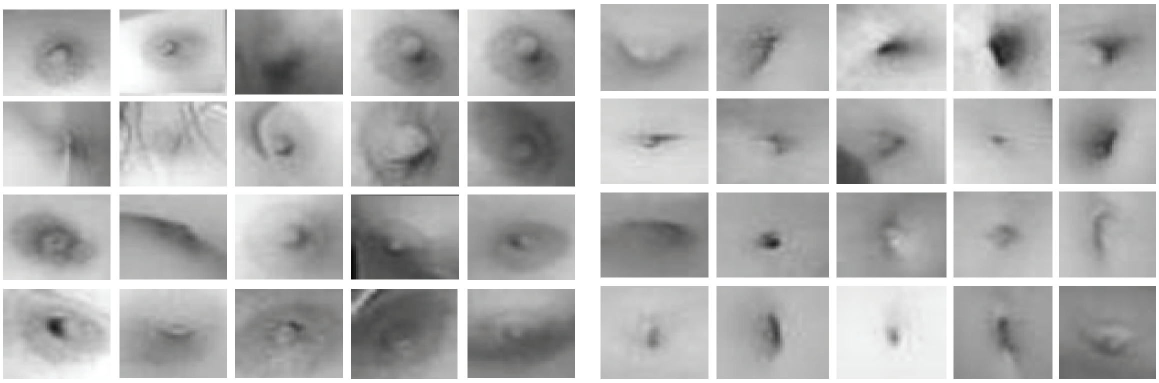
(4) Weights update is $w_{t+1,i} = w_{t,i}\beta_t^{1-e_i}$.
While $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise and $\beta_t = \epsilon_t/(1-\epsilon_t)$.

(5) The strong classifier is

$$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \ge \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ -1 & \text{Otherwise,} \end{cases} \tag{5}$$

where $\alpha_t = \log(1/\beta_t)$.

3.2.3. Boosted Haar-Cascade. It is a built-in package of OpenCv [12], which supports only haar-like features [16]. The main focus of this method is the accuracy of object detection and less false detection. The word "cascade" means that the resultant classifier consists of several simpler classifiers that are applied subsequently to a region of interest until at some stage the candidate is barred or all the stages are passed. The word "boosted" means that the classifiers at every stage of the cascade are complex themselves and they are built out of basic classifiers using one of four different boosting techniques (weighted voting). Currently Discrete Adaboost, Real Adaboost, Gentle Adaboost, and Logitboost are supported. In this research Gentle Adaboost (GAB) has been applied to improve classifier's generalization ability.


(a) Positive samples

(b) Negative samples

Figure 3: Training samples.

3.2.4. Boosted Train-Cascade. OpenCV train-cascade package supports both the haar-like features [16] and LBP (local binary pattern) [18] and the multicore platform for object detection [18]. The main focus of this method is faster detection. There is a drawback, that is, substantial false positive rate. Without this limitation this method would be more suitable for object detection. The main difference between haar-cascade and train-cascade is the structure of feature set data. Train-cascade uses binary data for storing feature set whether haar-cascade uses double type data [12, 15, 19].

4. Experiment

The OpenCV library is designed to be used in conjunction with applications that pertain to the field of human computer interaction (HCI), biometrics, robotics, image processing, and other computer vision related areas where visualization is important and includes an implementation of haar-classifier detection and training [8]. To train the classifiers, two sets of images are needed. One set contains an image or scene that contains the object of interest, in this case a nipple feature, which is going to be detected. This set of images is referred to as the positive images. The other set of images, the negative images, contains one or more instances of the object. The location of the objects within the positive images is specified by the image name, the upper left pixel, and the height and width of the object [16]. In this research we used Gentle Adaboost haar-cascade and train-cascade classifiers for training nipple dataset. We have 1198 positive training samples and 1995 negative images. At first positive images were filtered using YCbCr skin color model, after nipple objects were cropped and scaled to 20 × 20 pixels. This would help significant false minimization. For faster computation we used Gentle Adaboost (GAB) classifier. Minimum hit rate and maximum false alarm were set as 0.995 and 0.5, respectively. After training 1155 weak classifiers, we obtained 15 staged strong Gentle Adaboost classifiers. Figure 3 shows some cropped positive and negative nipple images.

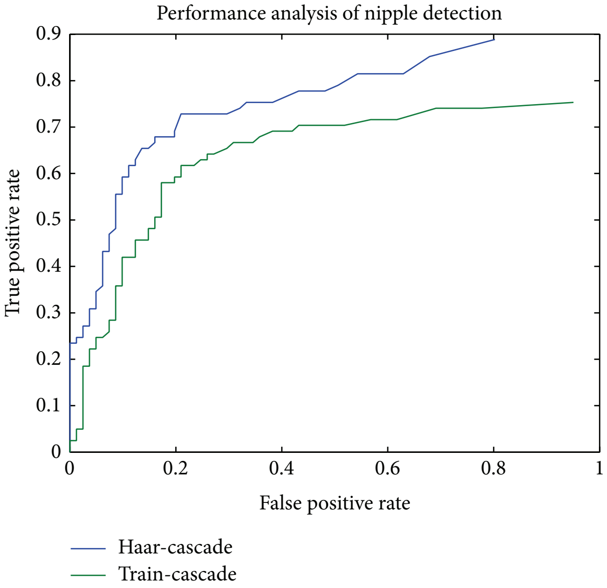


Figure 4: ROC for two classifiers.

5. Results

Figure 4 illustrates the robustness of our experiments. The performance was illustrated through a receiver operating characteristics (ROC) curve. We tested our classifier with 400 classified nipple images and 125 nonnipple images. It is shown that the performance is better for haar-cascade classifier, but in order to satisfy detection time train-cascade performed well. For instance, Haar-cascade classifier takes 0.162 seconds for checking each positive sample, while train-cascade needs 0.127 seconds.

5.1. Comparison with Existing Nipple Detection Methods. According to the review there was only three papers published based on nipple detection. A comparative analysis between existing methods and our methods is shown in Table 1.

Table 1: Strength and weakness of different nipple detection methods.

Methods	Detection rate (%)	False positive (FP) %	False negative (FN) %
Self-organizing map (SOM) [10]	65.40	0.22	34.60
Adaboost [20]	75.64	17.40	24.40
Cascaded Adaboost (haar-cascade) [21]	90.37	7.46	4.86
Gentle Adaboost with haar-cascade (our approach)	98.75	1.00	1.25
Gentle Adaboost with train-cascade (our approach)	84.29	22.22	15.71

Table 1 documents a comparative analysis on detection rate, false positive rate, and false negative rate between three existing methods and our two proposed methods using Gentle Adaboost haar-cascade and train-cascade. Gentle Adaboost haar-cascade outperformed the highest detection rate and lowest false negative rate. The lowest false positive rate was achieved by using self-organizing map [7] but it has a significant false negative rate.

6. Conclusion

Obscenity is a vital issue for Internet safety. For ensuring safe browsing, researchers are working hard to find a concrete methodology. Unfortunately it is impossible and hence there are a large number of different techniques available to address this issue. Existing systems are mainly focused on skin color tones. The main problem of those techniques is huge false detection due to skin-like objects and color. Also it identifies nudity with partially exposed images. In this situation erotogenic human body parts detection technique solves the problems. The literature was addressed only on human body parts. In our research we combined skin color and a vital part of human body part, which can address offensive images easily. In this paper we tried to develop a novel method for accurately detecting nipples from pornographic images. Exposed nipples are considered erotogenic human body parts and vital issue for nudity. Our aim was to filter that kind of offensive images. Here, haar-cascade and train-cascade methods were analyzed using Gentle Adaboost algorithm and it was found that haar-cascade performed well in accordance with accuracy and train-cascade improves speedup of detection process. Moreover, skin filter prior to training made our system more robust and eliminated significant number of false images. Our experimental results are better than three prior works on nipple detection (Table 1), but still there is some false detection. This limitation can be overcome by using some heterogeneous classifiers with appropriate large dataset.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Authors' Contribution

Rashed Mustafa and Yang Min contributed equally to this work and should be considered co-first authors.

Acknowledgments

This research was supported in part by Shenzhen Technical Project (Grant no. HLE201104220082A), National Natural Science Foundation of China (Grant no. 61105133), and Shenzhen Public Technical Platform (Grant no. CXC201005260003A).

References

[1] J.-Q. Zhu and C.-H. Cai, "Real-time face detection using gentle AdaBoost algorithm and nesting cascade structure," in *Proceedings of the 20th IEEE International Symposium on Intelligent Signal Processing and Communications Systems (ISPACS '12)*, pp. 33–37, New Taipei, Taiwan, November 2012.

[2] D. Forsyth, M. Fleck, and C. Bregler, "Finding naked people," in *Proceedings of the 4th European Conference on Computer Vision*, pp. 593–602, 1996.

[3] R. Kjeldsen and J. Kender, "Finding skin in color images," in *Proceedings of the 2nd International Conference on Automatic Face and Gesture Recognition*, pp. 312–317, Killington, VT, USA, October 1996.

[4] P. Yogarajah, J. Condell, K. Curran, A. Cheddad, and P. McKevitt, "A dynamic threshold approach for skin segmentation in color images," in *Proceedings of the 17th IEEE International Conference on Image Processing (ICIP '10)*, pp. 2225–2228, Hong Kong, September 2010.

[5] P. Fuangkhon and T. Tanprasert, "Nipple detection for obscene pictures," in *Proceedings of the 5th International Conference on Signal, Speech and Image Processing*, pp. 315–320, Greece, 2005.

[6] D. Chai and A. Bouzerdoum, "Bayesian approach to skin color classification in YCbCr color space," in *Proceedings of the IEEE Region Ten Conference (TENCON '00)*, vol. 2, pp. 421–424, September 2000.

[7] X. Kejun, W. Jian, N. Pengyu, and H. Jie, "Automatic nipple detection using cascaded adaboost classifier," in *Proceedings of the 5th International Symposium on Computational Intelligence and Design (ISCID '12)*, pp. 427–432, Hangzhou, China, October 2012.

[8] J.-G. Wang and E. Sung, "Frontal-view face detection and facial feature extraction using color and morphological operations," *Pattern Recognition Letters*, vol. 20, no. 10, pp. 1053–1068, 1999.

[9] Y. Wang, J. Li, H. Wang, and Z. Hou, "Automatic nipple detection using shape and statistical skin color information," in *Advances in Multimedia Modeling*, vol. 5916 of *Lecture Notes in Computer Science*, pp. 644–649, 2010.

[10] N. Pengyu and H. Jie, "Pornographic image filtering method based on human key parts," in *Proceedings of the International Conference on Information Technology and Software Engineering*, Lecture Notes in Electrical Engineering, Springer, Berlin, Germany, 2013.

[11] X. Shen, W. Wei, and Q. Qian, "The filtering of internet images based on detecting erotogenic-part," in *Proceedings of the 3rd International Conference on Natural Computation (ICNC '07)*, pp. 732–736, Haikou, China, August 2007.

[12] Q.-F. Zheng, W. Zeng, G. Wen, and W.-Q. Wang, "Shape-based adult images detection," in *Proceedings of the 3rd International Conference on Image and Graphics*, pp. 150–153, December 2004.

[13] Y. Wang and B. Yuan, "A novel approach for human face detection from color images under complex background," *Pattern Recognition*, vol. 34, no. 10, pp. 1983–1992, 2001.

[14] C. Messom and A. Barczak, "Fast and efficient rotated haar-like features using rotated integral images," in *Proceedings of the Australasian Conference on Robotics and Automation (ACRA '06)*, pp. 1–6, December 2006.

[15] J. Shah, M. Sharif, M. Raza, and A. Azeem, "A survey: linear and nonlinear PCA based face recognition techniques," *International Arab Journal of Information Technology*, vol. 10, no. 6, pp. 536–545, 2013.

[16] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. I511–I518, December 2001.

[17] F. C. Crow, "Summed-area tables for texture mapping," in *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '84)*, pp. 207–212, 1984.

[18] S. Liao, X. Zhu, Z. Lei, L. Zhang, and S. Li, "Learning multi-scale block local binary patterns for face recognition," in *Proceedings of the International Conference on Biometrics (ICB '07)*, pp. 828–837, 2007.

[19] A. Azeem, M. Sharif, M. Raza, and M. Murtaza, "A survey: face recognition techniques under partial occlusion," *IAJIT Issues*, vol. 11, no. 1, 2014.

[20] P. Wilson and J. Fernandez, "Facial feature detection using haar classifiers," *The Journal of Computing Sciences in Colleges*, vol. 21, pp. 127–133, 2006.

[21] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting," *The Annals of Statistics*, vol. 28, no. 2, pp. 337–407, 2000.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 628516, 9 pages
http://dx.doi.org/10.1155/2014/628516

Research Article

Cost-Sensitive Learning for Emotion Robust Speaker Recognition

Dongdong Li,[1] Yingchun Yang,[2] and Weihui Dai[3]

[1] *School of Information Science and Engineering, East China University of Science and Technology, Shanghai 200237, China*
[2] *Department of Computer Science and Technology, Zhejiang University, No. 38, Yuquan Road, Zhejiang 310027, China*
[3] *School of Management, Fudan University, No. 220, Handan Road, Shanghai 200433, China*

Correspondence should be addressed to Dongdong Li; ldd@ecust.edu.cn and Weihui Dai; whdai@fudan.edu.cn

Received 19 April 2014; Accepted 22 April 2014; Published 4 June 2014

Academic Editor: Yu-Bo Yuan

In the field of information security, voice is one of the most important parts in biometrics. Especially, with the development of voice communication through the Internet or telephone system, huge voice data resources are accessed. In speaker recognition, voiceprint can be applied as the unique password for the user to prove his/her identity. However, speech with various emotions can cause an unacceptably high error rate and aggravate the performance of speaker recognition system. This paper deals with this problem by introducing a cost-sensitive learning technology to reweight the probability of test affective utterances in the pitch envelop level, which can enhance the robustness in emotion-dependent speaker recognition effectively. Based on that technology, a new architecture of recognition system as well as its components is proposed in this paper. The experiment conducted on the Mandarin Affective Speech Corpus shows that an improvement of 8% identification rate over the traditional speaker recognition is achieved.

1. Introduction

Biometric security systems are based on human exclusive and unique characteristics, such as fingerprints, face, voice, iris, and retina [1, 2]. These systems are used as an extra barrier to prevent unauthorized access to protect data by recognizing the users by their specific physiological or behavioral characteristic. This method is more reliable than the conventional method because it is based on “something one is” rather than “something one knows/has.”

With improved research of vocal signals, people’s interactions through internet and mobile devices, such as phone banking, internet browsing, and secured information retrieval by voice, are becoming popular in a very rapid way [3]. There exists a need for greater security as these human-machine interactions over telephone lines and internet. At the same time, the powerful and ubiquitous handheld devices, such as smart phones and handheld computers, may contain a myriad of sensitive or personal information. All the applications mentioned above put great demand on speaker recognition based on modeling the speaker vocal tract characteristics, providing secure access to financial information (e.g., credit card information, bank account balance, etc.) or other sensitive customer information (e.g., healthcare records) [4]. Speaker verification provides an extra barrier to prevent unauthorized access to protect data and enhances the security offered by personal identification numbers or user selected passwords. It allows for contactless activation and mitigates the risks of stolen or lost keys, passwords, or keycards.

Automatic speaker recognition can be grouped into the following two classes: speaker verification and speaker identification. Speaker verification is the process to confirm the claim of identity and declare the person to be true or imposter. It is inclined to be used in security system using user specified passcodes for secure user logins. Speaker identification is the process to determine which one best matches the input voice sample from a pool of speakers’ voices. Its main application area is forensics and investigation, where there is a need to determine the identifier of a person. According to the type of spoken utterances, speaker recognition can also be divided into three categories: text-independent, text-dependent, and text-prompted. In text-independent systems, an arbitrary phrase is uttered to recognize the speaker. In text-dependent systems, a fixed "voice password" is uttered. In "text-prompted" systems, an instruction is given to ask the speaker to utter a certain phrase.

Previous work on security-based speaker recognition systems largely falls within the domain of dealing with interspeaker variables, like channel and background noise. However, intraspeaker variables, like emotion and health state, can also cause an unacceptably high error rate, which limits the commercial viability of speaker recognition systems [5]. Driven by rapid ongoing advances in affective computing, speaker recognition with affective speech (SRAS) is now becoming a popular consideration of modern speaker recognition research. In real life, we cannot expect the speaker to be always in neutral or normal mood. Most of the speaker recognition systems enroll the speaker model with neutral speech. Such systems could distinguish speakers from the others accurately when the speaker provides neutral speech to identify. However, when the recognition step is faced with emotional speech, like angry speech or delighted speech, the systems suffer emotional state mismatch between training and testing stage and the performance deteriorates. We cannot afford to develop the speaker models in all possible emotions for improving the performance, which degrades the user-friendliness of the speaker recognition system. SRAS becomes important because of the difficulty in acquiring large number of affective speeches from the speakers.

In sophisticated human-machine interaction, equipping the computer with the affective computing ability so that it can recognize the identity of the user is urgently needed in many different applications [6]. In telecommunications, the telephone-based speech recognition performance can be enhanced with the SRAS systems. For example, in route emergency call applications which service for high priority emergency calls, the callers experience a panic and scary scene. Their voice is not neutral any more. In the meanwhile, SRAS can also facilitate the applications of call centre. In many cases, the speaker gets disappointed, anxious, or angry when they call to deal with after-sale services problems. SRAS can identify and assess the speaker and help the call centre quickly respond to the disputes as well as achieving the customers' satisfaction.

For such applications, it becomes necessary to take into account the affective impact of speaker recognition so that the speakers could be recognized even when there is only affective speech provided for testing. The focus of this work is to develop a robust intelligent human-machine interface, which is more adaptive and responsive to a speaker's identity in emotional environments. In this paper, we further the research work in [7] and apply cost-sensitive learning to optimize the classification procedure and system performance.

This paper is organized as follows. In the next section, we give a review to the related work. The emotional corpus and emotional speech analysis are introduced in Section 3. Section 4 is committed to cost-sensitive learning and its application to speaker recognition. Section 5 discusses the system architecture. The experiments comparison and result discussion are presented in Section 6. We close with a conclusion section.

2. Related Work

In the literature, there are a few studies that focus on speaker recognition with affective speech. Structure training [8, 9] is first proposed and noted as a promising approach to address this problem. The method attempts to elicit different manners of speaking during the enrollment and makes the system become familiar with the variation likely to be encountered in that person's voice. Emotion-added modeling method [10] also attempts to elicit different manners of speaking during the enrollment. The goal of the systems is to learn not only the true distribution of the speaker-dependent speech features but also the influences of various emotions that corrupt this distribution.

Most of such systems model the speakers with a variety of affective speech and achieve great success. Dongdong and Yingchun [11] construct the speaker models with clustered affective speech. This approach aims at the maximum utilization of the limited affective training speech data. The prosodic difference is exploited to cluster affective speech, and the corresponding models are built with the clustered speech for a given speaker.

Along the way, all these methods mentioned above ask users to provide additional reading (emotional) speech in the training stage, which would lead to the unfriendliness of the system.

On the contrary, other researchers aim to promote the SRAS performance with only neutral enrolled for training, by means of adding artificial affective information to neutral training speech or eliminating the affective information in the emotional testing speech. Feature domain compensation aims at adding emotional information to neutral acoustic features prior to model training. One example is the rules based feature modification method based on the statistics of prosodic features [12, 13]. Specifically, the rules of prosodic features modification of duration, pitch, and amplitude parameters are derived from a small number of the content matched source-target pairs. The speaker model is trained with an aggregation of data with all kinds of the converted affective speech and the neutral speech.

Krothapalli et al. [6] believe that performance of the speaker identification system developed using neutral features is better with transformed features compared to emotional features. He proposes neural network based feature transformation framework for mapping the time-aligned syllable level features from any specific emotion to neutral. Shahin investigates emotion identification when the database is biased towards different emotions based on each of HMMs [14] and SPHMMs [15, 16].

Besides, score domain compensation attempts to remove model score scales and shifts caused by varying affective conditions of speakers. An example of score domain compensation techniques is E-Norm [17]. By investigating the pitch distribution variation under different emotional states, Li et al. [7] propose an improved pitch envelope based frame-level score reweighted (PFLSR) algorithm to compensate the affective effect in both speaker verification and identification system. The PFLSR aims to separate the frames that have large speaker-emotion variability from the ones that are slightly affected by speakers' moods.

Most of the existing speaker recognition systems fail during affective speech due to emotional mismatch in the training and testing environments. Considering both the system friendliness and the algorithm complexity, a probability reweighted score domain compensation approach is proposed. The idea of score normalization has been long acknowledged to speaker verification at both utterance and frame level [18, 19]. It is widely used for its simplification, convenience, and excellent result. This work has furthered the study in [7] and used the supervised learning method to refine the final score.

3. Database and Speech Analysis

3.1. Database. The affective speech database evaluated in this paper is Mandarin Affective Speech Corpus (MASC) [20], which is distributed by the Linguistic Data Consortium. The speech in the database spans five different emotion types, which are neutral (unemotional), panic, anger, sadness, and elation. All the reading material is phonetically balanced, which covers all the phonemes in Chinese.

68 native speakers are elicited to utter 5 phrases and 20 sentences under five emotional states, as well as 2 extra neutral paragraphs speech. Each phrase and sentence is repeated for three times, respectively. Altogether the database contains 5,100 phrases (e.g., 5 phrases *3 times *68 subjects *5 emotional types), 20,400 utterances, and 136 paragraphs. The detailed material is described as follows.

(i) Five phrases: they are "shi de (yes)," "bu shi (no)," and three nouns as "ping guo (apple)," "huo che (train)," and "wang qiu (tennis ball)." In Chinese, these words contain many different basic vowels and consonants.

(ii) 20 sentences: these sentences include 12 semantically neutral ones and 2 emotional ones for each type portraying the four emotional states. Different syntactical types are also represented in the sentences, which follow the material design of RUSLANA [21].

(iii) Two paragraphs: they are two readings selected from a famous Chinese novel, stating a normal fact.

The MASC database is divided into three subsets: development set, training set, and evaluation set. The development set is composed of the speech from the first 18 people. The training set contains 50 speakers, whose 2 paragraphs of neutral speech are used to construct speaker models. The evaluation set is the utterance parts in five types of emotions. There are 50 such speakers, with 15000 authentic tests and 735000 imposter tests.

3.2. Speech Analysis. Referring to the affective speech, the prosody is a medium of emotional expression [22]. Phonetic research demonstrates that prosodic information involves complex articulatory and cognitive controls in speech [23–25]. Promising results of emotion classification have been achieved with the analysis of prosodic feature [26–28].

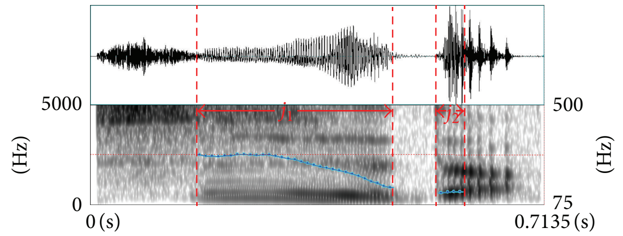


Figure 1: Example of segment boundaries estimation for the phrase "shi de." The vertical bars represent the segment boundaries from the critical points of pitch contours.

Prosody feature refers to the variables (range, contour, and jitter) of pitch, speaking rate, and intensity to convey nonlexical linguistic and high-lever cues. Among these features, pitch reveals the frequency at which the vocal folds vibrate and relies greatly on broad classes of sounds [29]. Pitch is investigated in this paper to indicate the characters of different emotion types.

The production of speech is the process of setting the air in rapid vibration. Periodic air pulses passing through vibrating vocal chords make voiced sounds, for example, "a" and "i" while unvoiced sounds such as "s" and "sh" are created by forcing air through a constriction in vocal tract, producing turbulence that is more noise-like. In this case, the pitch of an utterance is discontinuing. We can easily divide speech into voiced and unvoiced regions by detecting the points where the pitch contour appears or disappears. Figure 1 shows the waveform, spectrum, and pitch of phrase "shi de." The voiced segment alternates with the unvoiced one. The boundaries of voiced and unvoiced speech are represented by vertical dotted bars. The pitch contour of the voiced speech is defined as pitch envelope here. The statistics and analysis of pitch parameters take the pitch envelope as a unit, as it could indicate the average level of the speaker's voice frequency, which varies greatly under different emotional states.

Definition 1. Let $J = \{j_1, j_2, \dots, j_T\}$ be a pitch sequence of an utterance and let T be the frame numbers. $j_i = 0$ for the pitch of unvoiced segments and $j_i > 0$ for the pitch of voiced segments. The pitch envelope is denoted by $J^* = \{j_i \mid i = n, n+1, \dots, m\}$ and satisfies the following constraint:

(1) $j_i \neq 0$,

(2) $j_{n-1} = 0$, $j_{m+1} = 0$,

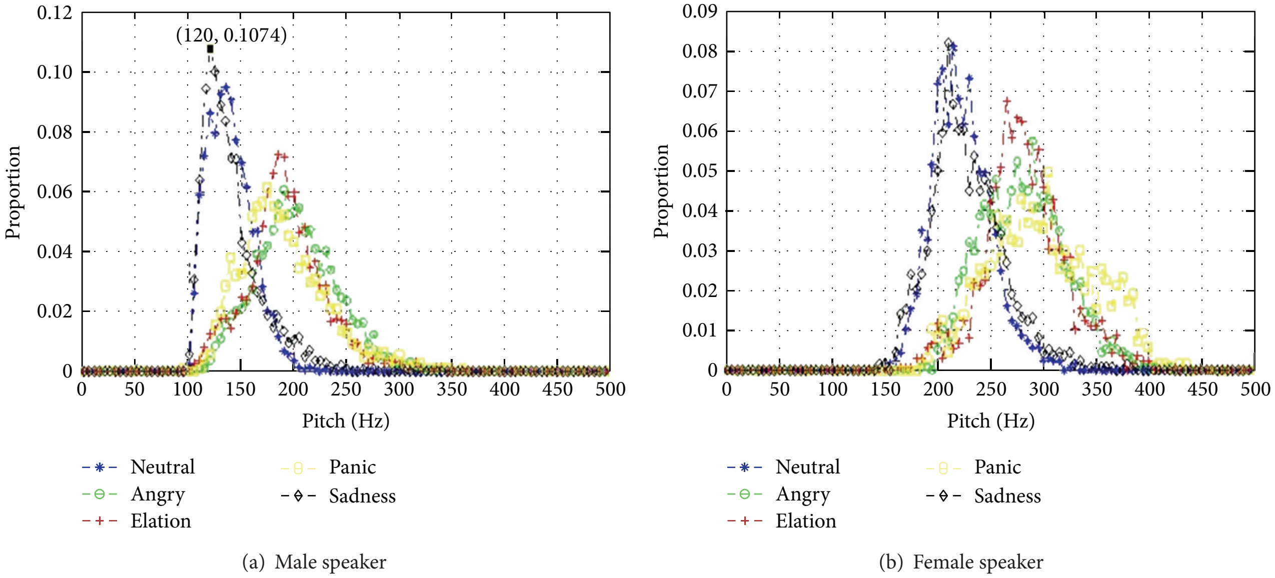
(3) $0 \leq n \leq i \leq m \leq T$.

The mean value of pitch envelope (PEM) can be calculated as

$$\overrightarrow{J}^{*} = \frac{1}{n-m+1} \sum_{i=m}^{n} j_i, \qquad (1)$$

where m and n are the numbers of the start and the end frame of a pitch envelope.

The pitch of a man's voice falls under low pitch frequency (60–250 Hz), whereas woman's voice is of the high pitch type (120–500 Hz). The distributions of PEM for male and



(a) Male speaker

(b) Female speaker

Figure 2: The probability distribution of PEM for the male (a) and female speakers (b) under the five emotion states.

female are studied separately. Figure 2 shows the probability distribution of PEM for all the sentences in MASC under the five emotion states. In particular, Figure 2(a) is the probability distribution of PEM with male's speech, while Figure 2(a) is the probability distribution of PEM with female's speech. The whole pitch frequency from 0 Hz to 500 Hz is equally divided into 100 scales with 5 Hz width each. For example, the point "100 Hz" on the abscissa represents the scales from 95 Hz to 100 Hz. The value on the ordinate corresponding to 100 Hz represents the proportion of voiced speech whose PEM falls in (95, 100) for each emotion. The point (120, 0.1074) means that there is 10.74% of PEM that falls in (115, 120) scope. Figure 2 demonstrates the probability distribution of PEM over 5 emotion types that can be divided into two groups. The neutral and sadness speech have similar distribution with smaller mean and variance value. The PEM probability distribution of anger, elation, and panic has larger mean and variance value. In this case, we assume that the voiced speech that has high PEM value is heavily affected by speaker's emotional mood. We partition the voiced speech into two groups according to a threshold pitch value: (1) the class that is highly different from neutral speech mainly includes the pitch envelop of anger, elation, and panic; (2) the class that is slightly different from neutral speech, mainly includes the pitch envelop of neutral and sadness. Both the male and female's speech have similar distribution, except that all the PEM of female are much higher than that of males'.

Thus, we can draw two kinds of important information. First, the PEM selection parameters should be set differently for male and female speakers as their dissimilarity distribution. Second, not all frames of the utterance are impacted dramatically by affective speech. In the speaker recognition task, the majority of frames of the test speech give the maximum probability (rank 1 score) to the target speaker (TS). In this case, the utterance could be correctly classified. However, the target speaker could not get the rank 1 score from all the test frames. In particular when the mood state of speakers shifts from neutral to affective, the vocal and prosody features are also changed. Some test frames give their confidence to a nontarget speaker (NTS) mistakenly because of the mismatch of emotions between the speaker models and the test utterances. With the number of frames that assign the maximum probability to the NTS becoming enormous, the score of the NTS could be comparable or even higher than that of the TS. Figure 3 shows the frame-level score rank's probability density functions for target speakers and nontarget speakers. The number on the abscissa represents the rank of score for each frame. For instance, the point (1, 0.149) in the red curve of target speaker means 14.9% of frames give the rank 1 score to the target speaker.

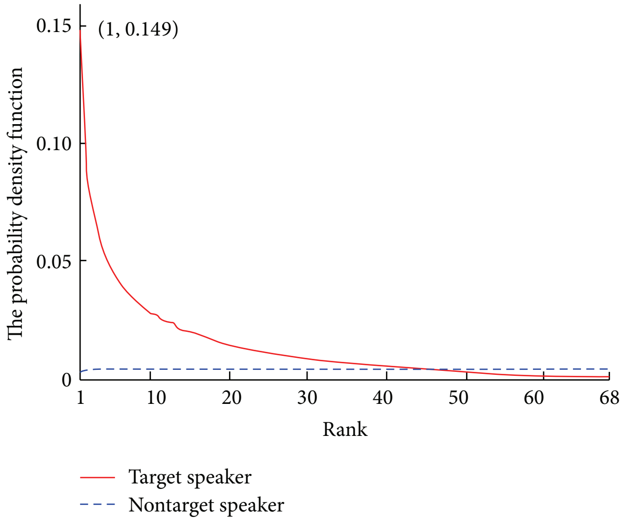


Figure 3: The frame-level score rank's probability density functions for target speakers and nontarget speakers over 68 subjects in MASC.

The reason why the test utterance is misclassified is not due to a nontarget speaker doing well but rather to a true speaker's model doing poorly. It is assumed that most frames still give the maximum likelihood to the target model, as they are not easily changed with the slight expressive corruption, while part of the frames that have large variations in F_0 may be affected by the emotion state change of speakers, and we define these frames as bad frames in this paper. To overcome the mistaken decisions induced by bad frames, we strengthen the roles of the good frames by giving them weight to exhibit their importance based on cost-sensitive learning.

Combined with the analysis of pitch, we divided the frames into two parts according to the variation of the F_0 value. The voiced part with high PEM that is heavily affected by the expressive speech (HA) is taken as bad frames. On the contrary, the voiced part that is slightly affected (SA) together with the unvoiced is considered as good frames.

4. Cost-Sensitive Speaker Recognition

4.1. Definition of Cost-Sensitive Learning. Cost-sensitive classification is based on a set of weights defining the expected cost when an object is misclassified. First, we give the definition for cost-sensitive classification problem.

Let $x \in \mathfrak{R}^n$ be a feature vector, let $\{1, 2, \dots, N\}$ be the label set of N classes, and let C be a $N * N$ cost matrix with entries $c_{i,j}$. $c_{i,j}$ are the cost of misclassifying an example of class i to class j; both i and j belong to $\{1, 2, \dots, N\}$. $c_{i,j} > 0$ if $i \neq j$ and $c_{i,j} = 0$ if $i = j$:

$$C = \begin{bmatrix} 0 & c_{1,2} & \dots & c_{1,N} \\ c_{2,1} & 0 & \dots & c_{2,N} \\ \dots & c_{i,j} & 0 & \dots \\ c_{N,1} & \dots & c_{N,N-1} & 0 \end{bmatrix}. \quad (2)$$

Here, $c_{i,i} = 0$ is the correct classification. The expectation cost of class i can be computed by

$$c_j = \sum_{i=1}^{N} c_{i,j}. \quad (3)$$

The cost-sensitive learning can be defined as follows.

Definition 2. Let $P(X \mid Y)$ be the unknown joint distribution of X and Y. Let F be a set of mappings from X to Y. The cost-sensitive learning procedure is to select a mapping f $(f \in F)$, to minimize the risk functional $R(f)$, defined as

$$R(f) = E_{P(X|Y)} c_{y,f(x)} = \int \left[\sum_{y=1}^{N} c_{y,f(x)} P(y \mid X) \right] p(x)\, dx. \quad (4)$$

It is easy to recognize that when given $c_{i,j} = 1$ if $i \neq j$, (4) reduces to the standard classification.

4.2. Speaker Recognition Task. In the speaker recognition task, given a group of N known speakers model $\Lambda = \{\lambda_1, \lambda_2, \dots, \lambda_N\}$ and a sample of test speech with T frames, the likelihood of X that belongs to the ith speaker can be written as $P(X \mid \lambda_i)$ according to Bayes' rule. In the case that frames in the utterance are independent, $P(X \mid \lambda_i)$ could be expressed as

$$P(X \mid \lambda_i) = \prod_{t=1}^{T} p(x_t \mid \lambda_i). \quad (5)$$

Obviously, there are relations between frames and frames, and the $P(X \mid \lambda_i)$ can be rewritten as

$$P(X \mid \lambda_i) = \prod_{t=1}^{T} f(p(x_t \mid \lambda_i)). \quad (6)$$

Equation (5) is a special case of (6) when $f(x) = x$.

In the process of computing the test utterance on the speaker model, the score is always mapped to the log domain for calculation facilitation as follows:

$$\begin{aligned} \text{score}(X \mid \lambda_i) &= \log p(X \mid \lambda_i) \\ &= \log \prod_{t=1}^{T} f(p(x_t \mid \lambda_i)) \\ &= \sum_{t=1}^{T} f(\log p(x_t \mid \lambda_i)). \end{aligned} \quad (7)$$

According to the frame selection conducted in Section 3, the frames in the SA part and unvoiced part are reweighed to strengthen their confidence to the maximum likelihood model. Thus, for a special speaker i, the utterance level score of a T frame speech sequence is defined as

$$\text{score}(X \mid \lambda_i) = \sum_{t=1}^{T_1} f(\log p(x_t \mid \lambda_i)) + \sum_{t=1}^{T_2} \log p(x_t \mid \lambda_i), \quad (8)$$

where T_1 is the number of frames in SA and unvoiced part, T_2 is the number of frames in HA part, respectively, and $T = T_1 + T_2$.

Given the frame vector x_t in the SA part and the speaker i, the cost-sensitive function can be assumed as

$$f(\log p(x_t \mid \lambda_i)) = \sum_{y=1}^{N} c_{y,i} * \log p(x_t \mid \lambda_i). \quad (9)$$

Note that cost matrix C only needs to be computed once when function L is defined.

It is obvious that the reweight function should meet the rule that the frame score and the cost matrix L are in direct ratio.

4.3. Cost-Sensitive Parameters Learning. To deal with the class-dependent cost-sensitive learning problem, the data space expansion technique is adapted [30, 31]. Each example is expanded to N examples of different classes. The weights of N examples are decided based on the loss of the corresponding misclassifications. When a test utterance is compared with a certain speaker model, the sum loss of its expanded N examples is in proportion to the loss of classifying it to that speaker model. The details of the expansion technique are given as follows.

Assume that in a speaker classification task, sample X is assigned to speaker model Y by classifier $f(X)$. C_y is positive as well as being not less than the largest misclassification loss in order to keep weights of expanded examples positive. The loss of $f(X)$ on the example (X, Y) is defined as

$$\begin{aligned} & c_{y,f(x)} \\ & = \sum_{i=1}^{N} c_{y,i} - \sum_{i=1}^{N} c_{y,i} I\left(f(x) \neq i\right) \\ & = \sum_{i=1}^{N} c_{y,i} - \sum_{i=1}^{N} c_{y,i} I\left(f(x) \neq i\right) + (K-1)\, c_y - (K-1)\, c_y \\ & = \sum_{i=1}^{N} \left(c_y - c_{y,i}\right) I\left(f(x) \neq i\right) - \sum_{i \in \{1,\ldots,N\} \setminus y}^{N} \left(c_y - c_{y,i}\right), \end{aligned} \tag{10}$$

where $I(x)$ is a step function with value 1 if the condition in the parenthesis is true and 0 otherwise.

The expanded examples (X^n, Y^n) with weights $w_{y,n}$ are defined as

$$X^n = X, \qquad Y^n = n, \qquad w_{y,n} = \left(c_y - c_{y,n}\right). \tag{11}$$

Substituting (11) into (10), we can get

$$c_{y,f(x)} = \sum_{i=1}^{N} w_{y,i} I\left(f(x) \neq i\right) - \varphi\left(y\right), \tag{12}$$

where $\varphi(y) = \sum_{i \in \{1,\ldots,N\} \setminus y}^{N} w_{y,i}$.

The loss of $f(x)$ on the example (X, Y) could be computed by a weighted loss of $f(x)$ on expanded examples minus a variable irrelevant to $f(x)$. The weights can modify the distribution on (X, Y) and produce a new one as well. In other words, cost-sensitive learning can be reduced to the standard classification [30].

5. System Architecture

Our previous work presented a pitch envelope based frame-level score reweighted speaker recognition framework [7]. The main contribution of this work is to introduce the cost-sensitive learning to reweigh the score. The testing process of the proposed speaker recognition system relies on 3 modules: gender identification, PEM based pitch envelop selection, and frame-level probability reweighed.

The purpose of the gender identification is to set different PEM threshold for male speakers and female speakers. This process is taken before frame selection. Given an utterance, the Mel frequency cepstral coefficients (MFCC) feature is extracted and tested with both male and female GMM models to produce the likelihood scores. The utterance is classified to the gender that has higher likelihood score. Corresponding frame selection thresholds are set and adopted based on the result of gender identification.

In the process of the PEM based pitch envelop selection, the variation of pitch distribution under different emotional states is analyzed and compared with PEM threshold. The voiced envelop frames whose mean pitch value is smaller than threshold and the unvoiced part are chosen for reweighting.

The score reweight step aims to strengthen the confidence of the selected speech segments and optimize the final accumulated frame scores over the whole test utterance.

6. Experiment and Discussion

6.1. Experiment Settings. The evaluation task conducted in the experiments is text-independent and closed-set speaker recognition. The front end processing of speech single is as follows. A 13-dimensional Mel frequency cepstral coefficients (MFCC) vector is extracted from the preemphasized speech signal every 16 ms using a 32 ms Hamming window. A simulated triangular filter bank on the DFT spectrum is used to compute the Mel cepstral vector. Delta-cepstral coefficients are then computed and appended to the 13-dimensional cepstral vector, producing a 26-dimensional feature vector. The speaker classification, the GMM, consists of 32 mixture components. In the gender identification, two 1024 mixture GMMs, male and female model, are built with MAP method using the speech from the development subset. The statistical F_0 thresholds of PEM for the female and male speakers are set as 289 Hz and 189 Hz, respectively.

6.2. The Baseline System with Neutral Models. The aim of the first experiment is to capture the fluctuation in the system performance with various affective speeches. The speaker models are built with paragraph part on neutral speech, and the test utterances are in anger, elation, panic, sadness, and neutral, respectively. Figure 4 gives the verification results with the five types of affective speech tested independently with neutral speaker models. The verification performance will decline greatly when the system is involved with affective utterances for testing. It is clear that the consistence affective state of the training and testing utterances is important. The verification performance for speech in anger, elation, and panic drops more sharply than that in sadness. It is reported that the speakers would have a much higher arousal level mood when they are in the emotion of anger, elation, and panic than that of sadness [32]. This is one of the possible reasons that the EER of test speech in sadness state goes down to 26.1%, while the EER of test speech in other three affective states have a sharper drop.

6.3. Experiment Results and Discussions. The identification rates (IR) of the standard accumulated approach and the CSSR on emotional speech of anger, elation, neutral, panic, and sadness are shown in Table 1. The enhancement of IR for speech in anger, elation, and panic achieves 11.94%, 13.53%, and 9.84%, respectively, which is significantly greater than that achieved for speech in sadness and neutral. A possible reason is that when speakers are in the emotion of anger,

Table 1: Comparison of system performance under different types of affective speech (%).

Method	Baseline	CSSR
Anger	21.80	33.74
Elation	22.70	36.23
Neutral	94.40	95.63
Panic	26.30	36.14
Sadness	51.13	54.67
Total	**43.27**	**51.28**

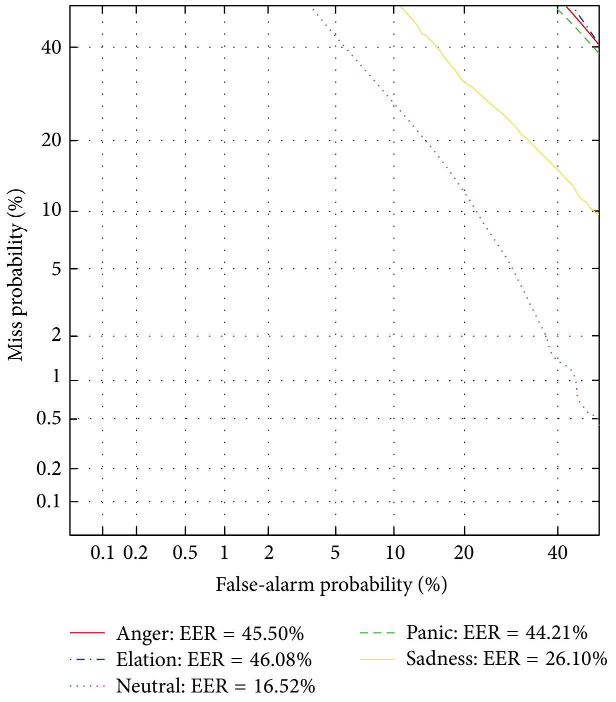


Figure 4: DET curves for the traditional speaker models trained with neutral speech only.

elation, and panic, they would have a much higher arousal level mood which makes more speech envelops with high PEM. In other words, the speech in anger, elation, and panic has a much greater number of the bad frames. Once the confidence of good frames are strengthened for the speech, the identification rates of speaker recognition are easily promoted.

We also apply the proposed method to speaker verification task and compare it with other score normalization methods, as shown in Figure 5. The performance measured by the detection error tradeoff function (DET) as well as equal error rate (EER). The EER is calculated as the operating point on the DET curve where the false-alarm and missed-detection rates are equal. Figure 5 shows the promise of the new approach. Evaluation results clearly show that CSSR technique outperforms the standard accumulated approach, T-norm, and ENORM methods for speaker verification on affective speech.

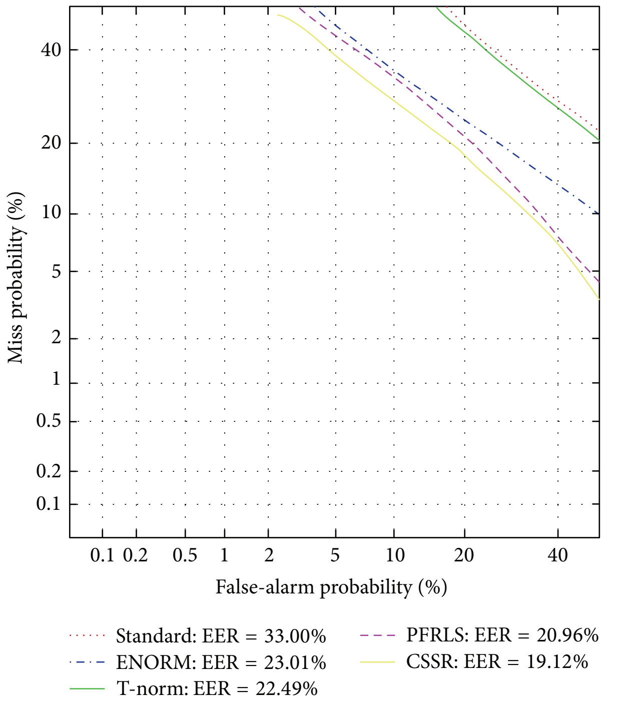


Figure 5: DET curves for the baseline, T-norm, ENORM, PFLSR, and CSSR based speaker verification system.

7. Conclusion and Discussion

In this paper, we introduce cost-sensitive learning to speaker recognition system with emotional speech. Based on an emotion robustness framework, cost-sensitive parameters are used to refine the probability of the slightly affected envelops and to strengthen the confidence of the target speakers. Promising results are achieved in both speaker identification and speaker verification system. In future work, more effective algorithms of the frame selection and clustering recognition method [33] may be suggested to be employed in this system. On the other hand, the emotional parameters associated with specific speakers can also be considered as the characteristics in the recognition of their speeches [34].

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research was supported by Nature Science Foundation of Shanghai Municipality, China (no. 11ZR1409600) and partly supported by the Natural Science Foundation of China (no. 91324010), the 2nd Regular Meeting Project of Science and Technology Cooperation between China and Serbia (no. 2-9), and China Scholarship Council (no. 201308310364).

References

[1] O. A. Esan, S. M. Ngwira, and I. O. Osunmakinde, “Bimodal biometrics for financial infrastructure security,” in *Proceedings of the Information Security for South Africa*, pp. 1–8, IEEE, August 2013.

[2] S. Rane, W. Ye, S. C. Draper, and P. Ishwar, “Secure biometrics: concepts, authentication architectures, and challenges,” *IEEE Signal Processing Magazine*, vol. 30, no. 5, pp. 51–64, 2013.

[3] A. Alarifi, I. Alkurtass, and A. S. Alsalman, “SVM based Arabic speaker verification system for mobile devices,” in *Proceedings of the International Conference on Information Technology and e-Services (ICITeS ’12)*, pp. 1–6, March 2012.

[4] K. S. Rao, A. K. Vuppala, S. Chakrabarti, and L. Dutta, “Robust speaker recognition on mobile devices,” in *Proceedings of the International Conference on Signal Processing and Communications (SPCOM ’10)*, pp. 1–5, July 2010.

[5] I. R. Murray and J. L. Arnott, “Synthesizing emotions in speech: is it time to get excited?” in *Proceedings of the International Conference on Spoken Language Processing (ICSLP ’96)*, Invited Paper, pp. 1816–1819, Philadelphia, Pa, USA, October 1996.

[6] S. R. Krothapalli, J. Yadav, S. Sarkar, S. G. Koolagudi, and A. K. Vuppala, “Neural network based feature transformation for emotion independent speaker identification,” *International Journal of Speech Technology*, vol. 15, no. 3, pp. 335–349, 2012.

[7] D. Li, Y. Yang, and T. Huang, “Pitch envelope based frame level score reweighed algorithm for emotion robust speaker recognition,” in *Proceedings of the 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops (ACII ’09)*, pp. 1–4, September 2009.

[8] K. R. Scherer, “Vocal communication of emotion: a review of research paradigms,” *Speech Communication*, vol. 40, no. 1-2, pp. 227–256, 2003.

[9] G. Klasmeyer, T. Johnstone, T. Bänziger, C. Sappok, and K. R. Scherer, “Emotional voice variability in speaker verification,” in *Proceedings of the ISCA Workshop on Speech and Emotion*, pp. 212–217, Belfast, Ireland, 2000.

[10] T. Wu, Y. Yang, and Z. Wu, “Improving speaker recognition by training on emotion-added models,” in *Affective Computing and Intelligent Interaction: Proceedings of the 1st International Conference (ACII ’05), Beijing, China, October 22–24, 2005*, vol. 3784 of *Lecture Notes in Computer Science*, pp. 382–389, 2005.

[11] L. Dongdong and Y. Yingchun, “Emotional speech clustering based robust speaker recognition system,” in *Proceedings of the 2nd International Congress on Image and Signal Processing (CISP ’09)*, pp. 1–5, Tianjin, China, October 2009.

[12] L. Dongdong, Y. Yingchun, W. Zhaohi, and W. Tian, “Emotion-state conversion for speaker recognition,” in *Affective Computing and Intelligent Interaction: Proceedings First International Conference (ACII ’05), Beijing, China, October 22–24, 2005*, vol. 3784 of *Lecture Notes in Computer Science*, pp. 403–410, 2005.

[13] W. Zhaohui, L. Dongdong, and Y. Yingchun, “Rules based feature modification for affective speaker recognition,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP ’06)*, pp. I661–I664, Toulouse, France, May 2006.

[14] I. Shahin, “Using emotions to identify speakers,” in *Proceedings of the 5th International Workshop on Signal Processing and Its Applications (WoSPA ’08)*, Sharjah, United Arab Emirates, 2008.

[15] I. Shahin, “Speaker identification in the shouted environment using Suprasegmental Hidden Markov Models,” *Signal Processing*, vol. 88, no. 11, pp. 2700–2708, 2008.

[16] I. Shahin, “Speaker identification in emotional environments,” *Iranian Journal of Electrical and Computer Engineering*, vol. 8, no. 1, pp. 41–46, 2009.

[17] W. Wu, T. F. Zheng, M.-X. Xu, and H.-J. Bao, “Study on speaker verification on emotional speech,” in *Proceedings of the INTERSPEECH and 9th International Conference on Spoken Language Processing (INTERSPEECH ’06—ICSLP)*, pp. 2102–2105, September 2006.

[18] D. A. Reynolds, “Speaker identification and verification using Gaussian mixture speaker models,” *Speech Communication*, vol. 17, no. 1-2, pp. 91–108, 1995.

[19] K. P. Markov and S. Nakagawa, “Text-independent speaker recognition using non-linear frame likelihood transformation,” *Speech Communication*, vol. 24, no. 3, pp. 193–209, 1998.

[20] T. Wu, Y. Yang, Z. Wu, and D. Li, “MASC: a speech corpus in mandarin for emotion analysis and affective speaker recognition,” in *Proceedings of the IEEE Workshop on Speaker and Language Recognition (Odyssey ’06)*, pp. 1–5, San Juan, Puerto Rico, June 2006.

[21] V. Makarova and V. Petrushin, “RUSLANA: Russian language affective speech database,” in *Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP ’02)*, pp. 2041–2044, 2002.

[22] S. Kemal, S. Elizabeth, H. Larry, and W. Mitchel, “Modeling dynamic prosodic variation for speaker verification,” in *Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP ’98)*, pp. 3189–3192, Sydney, Australia, 1998.

[23] R. W. Frick, “Communicating emotion. The role of prosodic Features,” *Psychological Bulletin*, vol. 97, no. 3, pp. 412–429, 1985.

[24] Santiago-Omar and Caballero-Morales, “Recognition of emotions in Mexican Spanish speech: an approach based on acoustic modelling of emotion-specific vowels,” *The Scientific World Journal*, vol. 2013, Article ID 162093, 13 pages, 2013.

[25] J. Hirschberg, “Communication and prosody: functional aspects of prosody,” in *Proceedings of ESCA Workshop Dialogue and Prosody*, pp. 7–15, 1999.

[26] N. Minematsu and S. Nakagawa, “Modeling of variations in cepstral coefficients caused by Fo changes and its application to speech processing,” in *Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP ’98)*, pp. 1063–1066, Sydney, Australia, 1998.

[27] M. Schröder, “Emotional speech synthesis: a review,” in *Proceedings of the 7th European Conference on Speech Communication and Technology (EUROSPEECH ’01)*, pp. 561–564, Aalborg Congress and Culture Centre, Aalborg, Denmark, September 2001.

[28] D. Ververidis and C. Kotropoulos, “Emotional speech recognition: resources, features, and methods,” *Speech Communication*, vol. 48, no. 9, pp. 1162–1181, 2004.

[29] P. Yang, Y. Yang, and Z. Wu, “Exploiting glottal information in speaker recognition using parallel GMMs,” in *Proceedings of the 5th International Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA ’05)*, pp. 804–812, Hilton Rye Town, NY, USA, July 2005.

[30] F. Xia, Y.-W. Yang, L. Zhou, F. Li, M. Cai, and D. D. Zeng, “A closed-form reduction of multi-class cost-sensitive learning to weighted multi-class learning,” *Pattern Recognition*, vol. 42, no. 7, pp. 1572–1581, 2009.

[31] N. Abe, B. Zadrozny, and J. Langford, “An iterative method for multi-class cost-sensitive learning,” in *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD ’04)*, pp. 3–11, August 2004.

[32] R. Cowie, R. Corive, E. Douglas-Cowie et al., "Emotion recognition in human-computer interaction," *IEEE Signal Processing Magazine*, vol. 18, no. 1, pp. 32–80, 2001.

[33] W. Dai, S. Liu, and S. Liang, "An improved ant colony optimization cluster algorithm based on swarm intelligence," *Journal of Software*, vol. 4, no. 4, pp. 299–306, 2009.

[34] Y. Wang, X. Hu, W. Dai, J. Zhou, and T. Guo, "Vocal emotion of humanoid robots: a study from brain mechanism," *The Scientific World Journal*, vol. 2014, Article ID 216341, 7 pages, 2014.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 494387, 9 pages
http://dx.doi.org/10.1155/2014/494387

Research Article

Multilabel Image Annotation Based on Double-Layer PLSA Model

Jing Zhang,[1,2] Da Li,[1] Weiwei Hu,[1] Zhihua Chen,[1] and Yubo Yuan[1]

[1] *School of Information Science and Engineering, East China University of Science and Technology, Shanghai 200237, China*
[2] *State Key Lab. for Novel Software Technology, Nanjing University, Nanjing, China*

Correspondence should be addressed to Zhihua Chen; czh@ecust.edu.cn and Yubo Yuan; ybyuan@ecust.edu.cn

Received 12 April 2014; Accepted 14 April 2014; Published 4 June 2014

Academic Editor: Shan Zhao

Due to the semantic gap between visual features and semantic concepts, automatic image annotation has become a difficult issue in computer vision recently. We propose a new image multilabel annotation method based on double-layer probabilistic latent semantic analysis (PLSA) in this paper. The new double-layer PLSA model is constructed to bridge the low-level visual features and high-level semantic concepts of images for effective image understanding. The low-level features of images are represented as visual words by Bag-of-Words model; latent semantic topics are obtained by the first layer PLSA from two aspects of visual and texture, respectively. Furthermore, we adopt the second layer PLSA to fuse the visual and texture latent semantic topics and achieve a top-layer latent semantic topic. By the double-layer PLSA, the relationships between visual features and semantic concepts of images are established, and we can predict the labels of new images by their low-level features. Experimental results demonstrate that our automatic image annotation model based on double-layer PLSA can achieve promising performance for labeling and outperform previous methods on standard Corel dataset.

1. Introduction

With the advent of the information era, the usage of the Internet is increasingly prevalent and the scale of the multimedia database is fast growing. How to organize, index, and retrieve the image data set has become an important issue and absorbed more attention in recent years. The existence of the semantic gap [1] leads to the fact that the images with similar visual characteristics may not be similar in semantics. For solving this problem, many image automatic annotation methods have been proposed for large-scale image understanding. Inspired by the techniques of text analysis, probability topic models are used to learn the relationships between the low-level visual features and high-level semantic concepts for automatic image annotation. Using probability topic model can effectively map high-dimensional image feature vectors to a low-dimensional space, greatly reducing the redundant information of the image and the time complexity of the algorithm. The widely used topic models include Latent Dirichlet Allocation (LDA) [2] and Probabilistic Latent Semantic Analysis (PLSA) [3]. The LDA topic model exploits complex bayesian structure and needs to determine prior parameters of the model, which makes its applicability less wide than PLSA model.

2. Related Work

Automatic image annotation is mainly to predict the semantic labels according to the visual content of images, which can be roughly divided into two categories [4]. The first one regarded automatic image annotation as a supervised classification problem. Specifically each word is viewed as a unique class and a binary classifier is trained for each class, or a multiclass classifier is trained by low-level features independently to predict the labels of unlabeled images. The second one represents the labels and visual features of images as different semantic spaces, in which the correlations between visual content and text labels are trained by the labeled images. Then the semantic concepts of unlabeled images will be predicted via statistical inference.

The first category is developed into the machine translation and multimodel fusion method in recent years from exploiting the simple SVM (support vector machine) or

GMM (Gaussian mixture model) for image classification in early years. In 2002, the translation model (TM) was proposed by Duygulu et al. [5], in which an image is cut into regions and each region corresponds to an object. Then these regions are clustered into blobs based on the features of regions, and labeling can be regarded as a process which translates blobs into labels. Kobus Barnard et al. [6] proposed a multimodel annotation algorithm which fused hierarchical clustering model, TM, and LDA model with information of different aspects. While the method considers annotation of both the whole image and the image regions, the joint probability distribution of blobs and keywords is obtained by learning, and the image annotation and regional annotation problems are translated into the associated problems among images, regions, blobs, and keywords.

The cross media relevance model (CMRM) proposed by Jeon et al. [7] also applies the segmentation regions to represent the image. Different from TM, this method considers that the keywords of the image do not get one-to-one correspondence to regions, and the images annotation is realized by learning the joint probability distribution between the keywords and regions of image. Lavrenko et al. [8] proposed continuous-space relevance model (CRM), which can be trained and modeled on the continuous features. And it is not dependent on the clustering of the low-level features, which makes it will not be affected by the clustering granularity and can achieve better performance. In the early-stage work, we also proposed an image annotation algorithm based on multiple models [9]. Two different models are used to analyze semantic concepts of foreground and background: Multiple Nystrom-Approximating Kernel Discriminant Analysis and Region Semantic Analysis, and then Latent Semantic Analysis assists in amending the annotation for error correction.

The most important part of the second category is to learn the link between the low-level visual features and high-level semantic features by the probability topic model. Blei and Jordan [10] model the keywords and image by correlation LDA model. Firstly, a series of latent topics are generated by associate visual features with semantic concepts, and image is decomposed into a series of collections of the latent topics. Then, a subset selected from these latent topics is converted to a number of hybrid models based on LDA, through which the image semantic annotation is generated. PLSA-WORDS annotation algorithm is proposed by Monay and Gatica-Perez [11], in which asymmetric learning algorithm is used to learn a latent space from text data and maintains the relation with visual features and achieves fairly good performance of image annotation and retrieval.

PLSA-FUSION [12] algorithm proposed by Li et al. learns not only the latent topics in the label aspect, but also the latent topics in the visual aspect. Then these two latent semantic topics are weight-fused through an adaptive asymmetric learning method, and a new model is generated, which can predict semantics of the unlabeled images. Akcay and Aksoy also adopt PLSA model [13] to detect the remote sensing image. They combined Principal Component Analysis (PCA) with mathematical morphology method for remote sensing image segmentation. Next they extracted features from the segmented regions and applied PLSA model from the perspective of pixels to measure the similarity between image regions. Then the segmented regions are clustered, and the information of spectrum and structure are combined to realize the semantic detection of remote sensing image. Different from above algorithms, Zhuang et al. analyzed PLSA model from a semisupervised perspective and applied it to image classification [14].

Traditional PLSA based image annotation models usually encounter a problem that the scale of visual and text words do not match [15]. Namely, the scale of vocabulary of text features is usually the power to 10, but that of vocabulary in visual aspect is up to about 500. So during analyzing, deviation will be produced. However, Semisupervised PLSA overcomes this problem well, which added label information of images into the EM algorithm for the calculation of model parameters.

We propose a new multilabel annotation method of images based on double-layer PLSA model, which applies relevant knowledge of latent topic space for image semantic annotation. In this method, we represent the image visual features by BoW model [16] and generate the latent semantic topic spaces through using first-layer PLSA, respectively, from the visual space and text space. Then two topic subspaces are integrated by second-layer PLSA, and the top layer latent semantic topics are obtained to realize the connection between the low-level visual features and high-level semantic concepts. Experimental results based on Corel5K illustrated that this method can effectively narrow down the semantic gap and achieve better performance on labeling multilabel image.

The rest of the paper is organized as follows. Section 3 presents the double-layer PLSA model and the image automatic annotation algorithm is introduced in detail in Section 4. Experiments and results analysis are illustrated in Section 5, followed by conclusions in Section 6.

3. Image Annotation Model Based on Double-Layer PLSA

Image automatic annotation algorithm based on double-layer PLSA fully utilizes the visual and semantic information of images to construct the annotation model by using PLSA. The second-layer latent semantic topics are looked on as the "bridge" to connect the visual features and semantic concepts. The framework of our annotation model is illustrated in Figure 1. There are two main parts in this framework, including representation of the image content and double-layer PLSA model. We will introduce them in detail as follows.

3.1. Representation of the Image Content. The model of BoW is widely applied in the natural language processing and information retrieval, in which texts (sentences or documents) are taken as unordered data sets and the influence of the word order is ignored. Nowadays this model is widely used in the field of computer vision. In our annotation model, BoW model is also used to represent the image content and the processing procedure is described as follows.

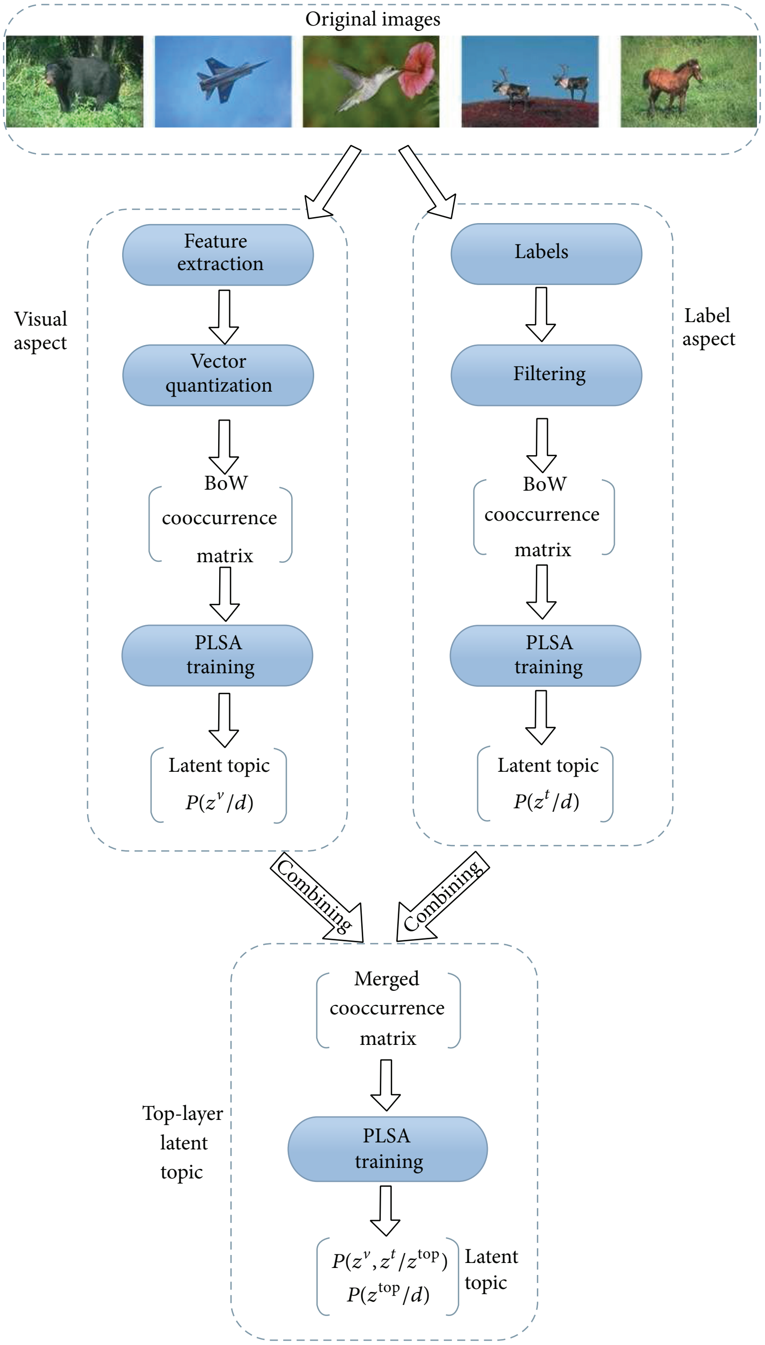


Figure 1: The flow diagram of the double-layer PLSA model.

In the semantic label space, the limited vocabularies annotated manually are looked upon as a label set, which is originally a series of unordered keywords set and can be represented by BoW directly. Suppose that the number of the given label t is N_t; then the label information of the image I_i can be represented as a vector with N_t dimensions, illustrated as follows:

$$T(I_i) = \{n(I_i, t_1), \ldots, n(I_i, t_j), \ldots, n(I_i, t_{N_t})\}, \quad (1)$$

where, $n(I_i, t_j)$ is the quantity of the label t_j appearing in the given image I_i; usually the values are 0 or 1.

In the visual feature space, the image is divided into small pieces with same size, and visual features of the small pieces are extracted and clustered to generate a visual vocabulary. Then image can be simply expressed as a collection of several visual words. Considering that the different low-level features of images can express various aspects of image content, and each has its advantages in specific aspects, so the combination of different features is a wise method to minimize the loss of feature discretization. Assuming that the N_v is the amount of category, which is gained by clustering, each cluster can be represented by a visual word. All visual words make up

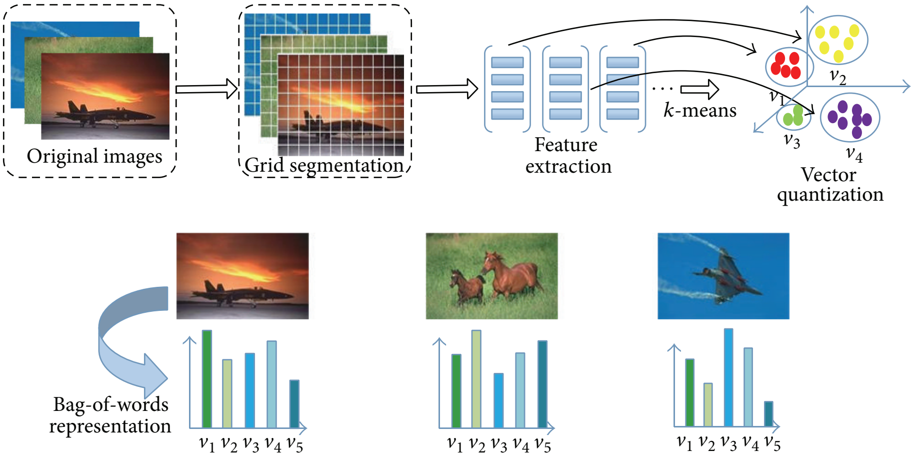

Original images
Grid segmentation
Feature extraction
k-means
v_1
v_2
v_3
v_4
Vector quantization
Bag-of-words representation
v_1 v_2 v_3 v_4 v_5
v_1 v_2 v_3 v_4 v_5
v_1 v_2 v_3 v_4 v_5

Figure 2: The flow chart of getting BoW representation of the image low-level feature.

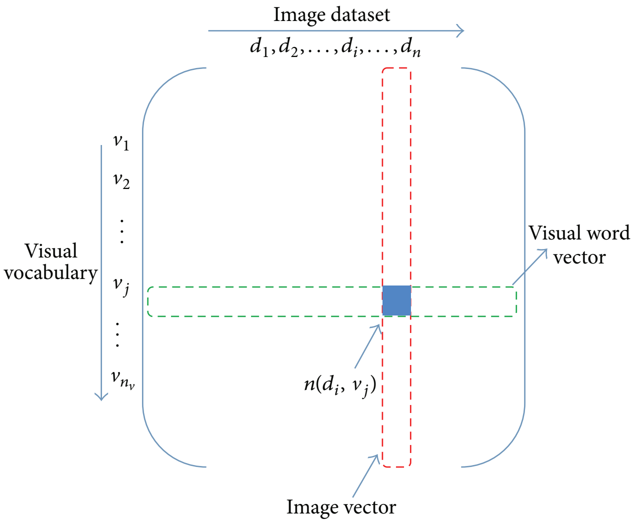


Figure 3: The "image visual-word" matrix of the image low-level feature.

the vocabulary, and an image I_i can be represented as a visual word histogram, which can be expressed as a vector with N_v dimensions and illustrated as (2). Consider

$$V(I_i) = n(I_i, v_1), \ldots, n(I_i, v_i), \ldots, n(I_i, v_{N_v}), \qquad (2)$$

where $n(I_i, v_i)$ represents the number of visual word v_i in the given image I_i. Unlike the elements of the vector in (1), $n(I_i, v_i)$ can be any integers greater than 0, and the combination of different visual features can be simply connected via feature vectors [17].

The BoW model provides an unified form to represent the image content, in which each feature plays an effective role in representation of image content. Figure 2 represents how to get the BoW representation of image low-level visual features. The first part describes grid segmentation, feature extraction, and the feature vectors quantization. The second part illustrates the process of image representation by BoW model.

To construct BoW representation by visual features, three steps are implemented as follows. Firstly, we extract the pyramid gradient direction histogram feature (Pyramid Histogram of Oriented Gradients, PHOG) [18, 19] on the whole image and obtain PHOG histogram. Secondly, the image is segmented into fixed grids, and the scale color descriptor (SCD) and Gabor texture feature are extracted for each small grid. Then, k-means is used to cluster the SCD and Gabor features of all grids, respectively, and the visual words are achieved. Finally, these two categories of visual word vectors are fused, and the BoW representation of each image is gained.

Through the BoW model the continuous low-level feature information of image is transformed into the discrete form, so that each image can be represented by the visual vocabulary vector simply. By putting all the feature vectors together, we can get the visual word cooccurrence matrix of the whole image set and the specific form is shown in Figure 3. Each column of the matrix represents an image vector, in which $n(d_i, v_j)$ represents the number that occurrence of visual word v_j appeared in image d_i.

3.2. Model of Double-Layer PLSA. According to the framework of double-layer PLSA model illustrated in Figure 1, the low-level features and labels in an image can be transformed to two different cooccurrence matrixes; then PLSA model is used to analyze the visual information and semantic information, respectively. Therefore, the low-level feature space and the high-level semantic space are mapped to two latent topic subspaces. However, there is no link between the two topic subspaces.

In order to achieve the labels of image semantics, the most important step is to establish the connection between the high-level semantics and the low-level features.

Assuming that the image set is $D = \{d_1, d_2, \ldots, d_i, \ldots, d_n\}$, and the vocabulary (visual-words or labels) set is $W = \{w_1, w_2, \ldots, w_j, \ldots, w_m\}$. then $N = \{n(d_i, w_j)\}$ represents the cooccurrence matrix of image visual-word and $n(d_i, w_j)$ is the frequency of the visual word w_j in the image d_i. In the analysis process of image by PLSA, a latent topic space $Z = \{z_1, z_2, \ldots, z_k\}$ is introduced to map the high-dimensional image visual-word cooccurrence matrix to a low-dimensional latent semantic topic space. At the same time the abstract relationship would be explored, and the conditional probability of the "image visual-word" can be described as follows:

$$P\left(d_i, z_k, w_j\right) = P\left(d_i\right) P\left(\frac{z_k}{d_i}\right) P\left(\frac{w_j}{z_k}\right). \qquad (3)$$

The joint probability of a word (or visual word) w_j with image d_i is a marginalization over the topics z_k as

$$P\left(d_i, w_j\right) = P\left(d_i\right) \sum_{k=1}^{K} P\left(\frac{z_k}{d_i}\right) P\left(\frac{w_j}{z_k}\right), \qquad (4)$$

where $P(z_k/d_i)$ is the conditional probability of the latent topic z_k given the image d_i, and $P(w_j/z_k)$ represents the conditional probability of the visual word w_j given the latent topic z_k. Moreover, PLSA can be illustrated in Figure 4; the nodes in the rectangular box express the three key elements "image" d, "latent topic" z, and "label" w, where the black nodes represent the observable random variables, and the white node represents the unobservable random variable.

If we regard the cooccurrence matrix of "image visual-word" as a $N * M$ matrix, then the matrix decomposition of PLSA can be illustrated in Figure 5.

The most direct way of that apply PLSA model to the image understanding is combining the "visual-word image" matrix N_{M*N_v} and "label image" matrix N_{M*N_t} into a new matrix $N_{M*(N_v+N_t)}$, in which "visual-word image" matrix and "label image" matrix has the same format. But in this way it will arise a new problem of that scale is not consistent. In general, each image usually has thousands of visual words, but in most cases the labels of the image will not be more than 20. Therefore, the visual vocabulary is in a dominant position, which affects the results much more.

Although some normalization methods can alleviate the adverse effects, the normalization process needs a lot of experiments to determine the proper weight of different topics, which increases the time complexity of the algorithm. Therefore, in this paper we bring forward a double-layer PLSA model to solve this problem, and the specific process of the model is shown in Figure 1.

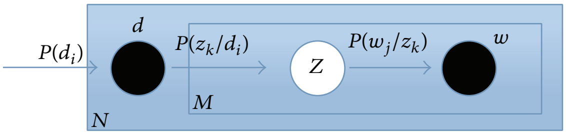


Figure 4: The graph representation of PLSA model.

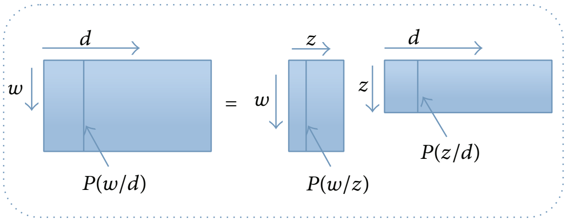


Figure 5: The matrix decomposition of PLSA model.

In the double-layer PLSA model the first layer includes two PLSA models, which conduct the latent semantic analysis, respectively, on image visual feature space and label space. The further analysis on latent semantic topics obtained from the first layer PLSA are processed to extract the second-layer latent semantic topics for forming the connection between low-level visual features and high-level semantic concepts.

As shown in Figure 1, the input of the second layer PLSA is cooccurrence matrix merged by the two latent topic matrixes generated from the first layer PLSA. As shown in Figure 6, in this merged cooccurrence matrix the column vectors are still the image collections $D = \{d_1, \ldots, d_n\}$, and the row vectors are the two latent topic subspaces $Z = \{z_1, \ldots, z_i, \ldots, z_K\}$ $(Z = \{z_1^v, \ldots, z_{K^v}^v, z_1^t, \ldots, z_{K^t}^t\})$ obtained from the first layer PLSA. z^v represents the latent topic distribution of the visual features, and z^t represents the latent topic distribution of the labels. $n(d_i, z_j)$ represents the probability of the first layer latent topic z_j in given document d_i. It is obvious that the image latent-topic cooccurrence matrix has the same form as the previous image visual-word cooccurrence matrix (as Figure 3); hence the PLSA model can be used again.

The calculation principle of the double-layer PLSA model is shown in Figure 7. There are 6 nodes in the rectangular box, including black nodes which represent observable random variables: images d, visual words v, and text labels t. White nodes represent unobservable random variables, namely, the latent topics, containing high latent topics Z^{top}, visual feature space latent topics Z^v, and the latent topics Z^t of the labels. Figure 7 describes the corresponding relations between training parameters. Firstly select image d according to the probability $P(d)$. Secondly obtain top layer topic probability distribution in the image d according to $P(z^{\text{top}}/d)$; then get the label latent topic distribution of image d in the first layer according to $P(z^t/z^{\text{top}})$. Finally, get the text label distribution of image d according to $P(t/z^t)$. The same to the visual aspect.

4. Image Annotation Algorithm

We analyze images from two aspects (x, $x \in \{v, t\}$, v represents visual feature, and t represents label) by double layer PLSA model. The image annotation algorithm and image

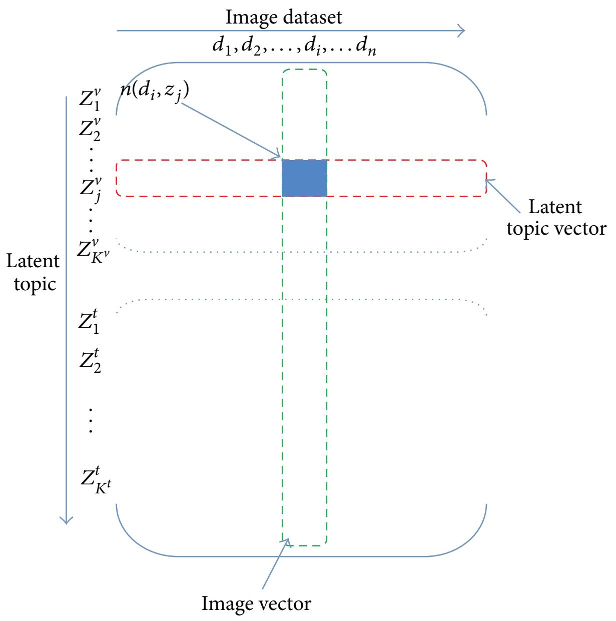


FIGURE 6: The cooccurrence matrix of image latent-topic.

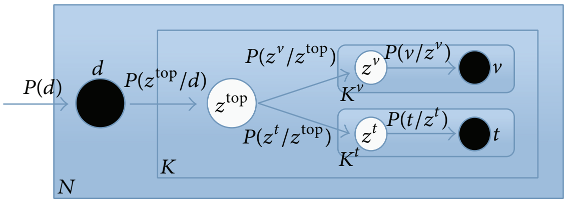


FIGURE 7: The calculating principle of the double-layer PLSA.

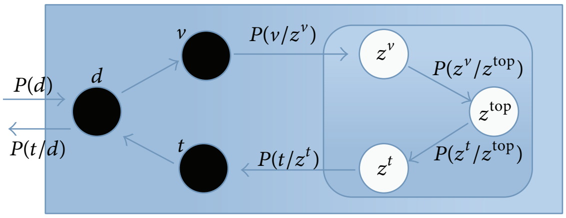


FIGURE 8: The tagging process of the image.

annotation process are, respectively, shown in Algorithm 1 and Figure 8.

In terms of the related theory of PLSA and training process of double PLSA model, $P(v/z^v)$, $P(t/z^t)$, $P(v/z^{top})$, and $P(t/z^{top})$ are not restricted to specific image and can be applied to all other images. The folding-in [12] algorithm, as the simplified EM algorithm, is adopted to realize the auto-image-annotation. In this algorithm, the known parameters are kept unchanged in the iteration process, constantly updating unknown parameters until the maximum likelihood function gets the max value.

5. Experiments

In order to validate the effectiveness of image semantic content analysis by the proposed model, experiments have been done on Corel5K, and the experimental results are compared with other algorithms on the same image set.

5.1. Image Database and Evaluation Measures. Corel5K is currently the widely used image set in image retrieval and annotation field, which contains 5000 images with 50 different categories, and each category has 100 copies with the similar high-level image semantics. The images in Corel5K have a total of 371 labels, which are defined by LSCOM. Most of these labels occurred frequently in image set, but there are several labels, such as "pool," "farms," and "coast," and that only occurred in 7 images. In order to reduce the influences of low-frequency words, we removed the tags appearing less than 8 times and finally constructed the ideal vocabulary with 260 labels [20].

We compared the annotation results of our double-layer PLSA model with the ground truth to verify the effectiveness of the proposed algorithm. The evaluation method based on label is used, including "recall" and "precision" of each label. In this experiment, we only took labels with the top five largest posterior probabilities as annotation result of each image and calculated the precision and recall for each label. For a given label w, the calculating formula of precision and recall are shown in the following:

$$\text{Precision} = \frac{N_{WT}}{N_W}, \qquad \text{Recall} = \frac{N_{WT}}{N_{WGT}}, \tag{5}$$

where N_{WT} is the correct number of w annotated by this algorithm, N_W represents the number of images containing w after image annotation, and N_{WGT} is the number of image including w in the ground truth.

```
(1) Input: a new untagged image $I$
(2) Get the BoW representation of $I$, a $N_v$ dimensions vector:
    $V(I) = \{n(I, v_1), \ldots, n(I, v_i), \ldots, n(I, v_{N_v})\}$
(3) According to the training parameter $P(v/z^v)$, applying the folding-in algorithm to get
    the visual latent topic distribution:
    $Z^v = \{n(I, v_1), \ldots, n(I, v_i), \ldots, n(I, v_{N_v})\}$
(4) The same as the step (3), according to $P(z^v/z^{\text{top}})$, using the folding-in algorithm to get
    the top-layer latent topic distribution of image $I$:
    $Z^{\text{top}} = \{P_{z_1^{\text{top}}}, \ldots, P_{z_i^{\text{top}}}, \ldots, P_{z_K^{\text{top}}}\}$
(5) Combine the training parameter $P(z^t/z^{\text{top}})$ and the top-layer latent topic distribution
    gained from the step (4) to get the label latent topic distribution of image $I$:
    $Z^t = \{P_{z_1^t}, \ldots, P_{z_i^t}, \ldots, P_{z_{K^t}^t}\}$
(6) According the training parameter $P(t/z^t)$ and the distribution of $z^t$, the probability of
    each image semantic label appearing on the image $I$ can be calculated:
    $T(I) = \{P_{t_1}, \ldots, P_{t_i}, \ldots, P_{t_{N_t}}\}$
    then choose $n$ ($n$ can be selected as need, in this paper $n = 5$) labels with the largest
    probabilities to construct the label set of image $I$
(7) Output: the $n$ labels of image $I$
```

Algorithm 1: The image concept detection algorithm.

5.2. The Contents of Experiment. In our experiments the whole data set is divided into two parts of which 4500 images are taken as the training set and the rest of them as the testing set. The visual features of image are represented by BoW model. First, segment the image into small grids. If the scale of segmentation is small enough, all the content in the image can be expressed in detail. But at the same time, it may increase the computing complexity of the algorithm. If the scale is too large, the image content will not be represented accurately enough. According to our research results on the image content representation with different segmentation scales [21], we adopt 15 $*$ 15 fixed-size, which makes the image content representation accurate and the computing complexity of the algorithm ideal. In our experiments, we chose PHOG descriptor (Pyramid Histogram of Oriented Gradients, PHOG) [18, 19], SCD, and Gabor texture as low level features, in which PHOG histogram has 425 dimensions. By amounts of experiences, we found that if the cluster number of SCD and Gabor texture are 325 and 250, respectively, the dimensions of BoW model are 1000, which is the best way to represent the image content.

An important parameter of PLSA model is the number of the latent topics, which determines the time needed for model training to a large extent. If the predefined latent topics are too few, they may not be good enough to express the potential relationship between the visual information and concepts. However, if they are too many, it will take a lot of time for training and the efficiency of the model will decrease. Meanwhile, it may increase the possibility of over-fitting. Considering the amounts' difference between the text label sets and visual vocabulary, and the corresponding parameters value mentioned in the reference [12], we defined the number of latent topics of the label text as 120 and the number of latent topics of the visual as 80. Then we obtained a total of 200 latent topics after the first-layer PLSA analysis. On the second-layer PLSA processing, we found that the results with using 50 top-layer latent topics to learn "image latent-topic" co-occurrence matrix obtained in the first layer were best by large amounts experiments.

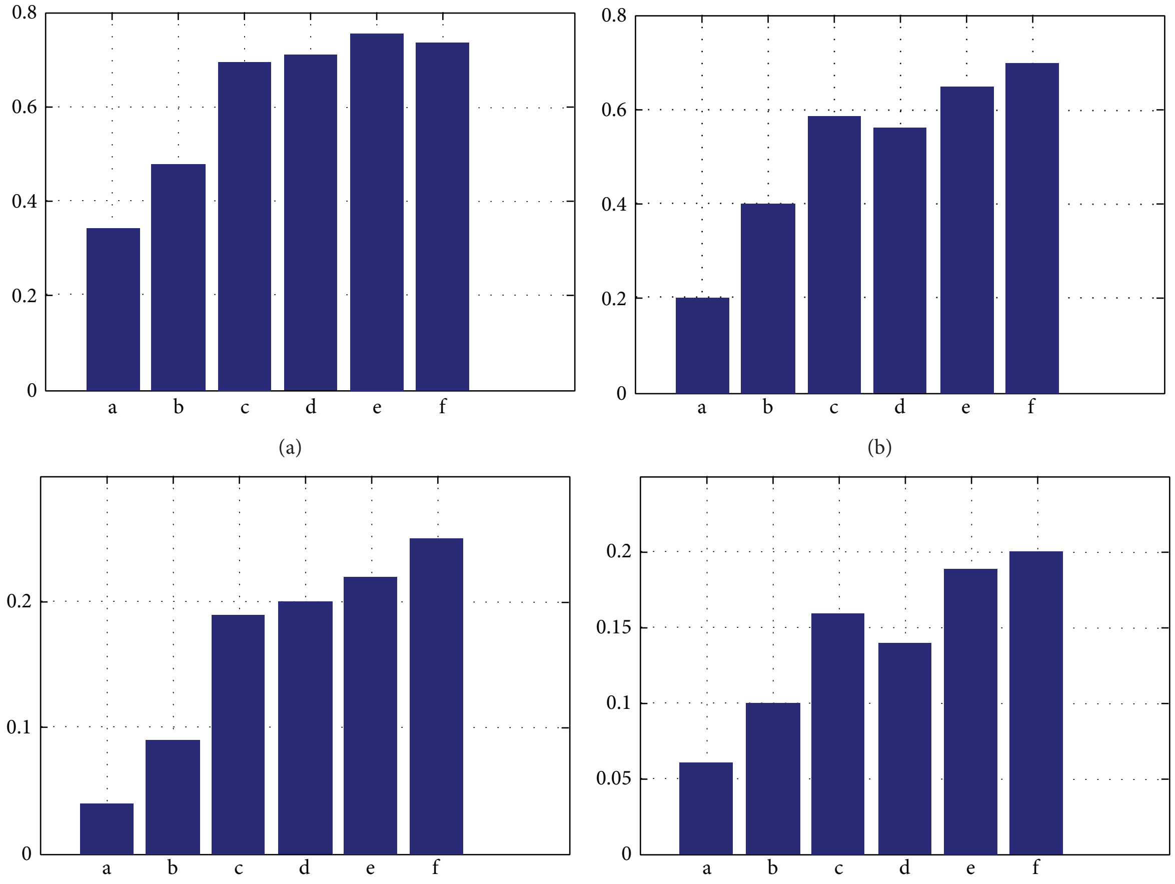
Table 1: Results on Corel5K by our algorithm: AR (average recall) and AP (average precision).

	49 labels		260 labels	
	AR	AP	AR	AP
Double-layer PLSA	0.74	0.70	0.25	0.20

5.3. Experimental Results and Analysis. The experimental results on Corel5K by the proposed algorithm are shown in Table 1. For the 49 labels with the optimum performance, the double-layer PLSA achieved the satisfactory experimental results, in which the average recall and precision were both over 70%. But in the 260 high frequency label set for the reason of the imbalance of label distribution, the average recall and precision were reduced to 25% and 20%.

In order to express the advantage of double PLSA model, we compared our algorithm with the algorithm PLSA-FUSION proposed by Li et al. [12], TM [5], CMRM [7], CRM [8], and PLSA-WORDS [11] mentioned in paper [12]. In these experiments we used the same experiments data and evaluation method, and evaluated the results in two same label vocabularies: the 49 labels with the optimum performance and the 260 high frequency label sets.

Table 2 and Figure 9 illustrate the evaluation results of the proposed algorithm, TM, CMRM, CRM, PLSA-WORDS and PLSA-FUSION algorithms on the two predefined labels. It can be seen from the contrast histograms, the double-layer PLSA model obtained the promising performance on both two label sets. In the label set with 49 optimum labels, the average precision of the double-layer PLSA model exceeds all



(a)

(b)

(c)

(d)

Figure 9: The comparison histogram of experiments ((a) the average recall of 49 labels, (b) the average precision of 49 labels, (c) the average recall of 260 labels, and (d) the average precision of 260 labels, where, a.TM b.CMRM c.CRM d.PLSA-WORDS e.PLSA-FUSION f.OURS).

Table 2: The results of contrast experiments: AR (average recall) and AP (average precision).

	49 labels		260 labels	
	AR	AP	AR	AP
TM	0.34	0.20	0.04	0.06
CMRM	0.48	0.40	0.09	0.10
CRM	0.70	0.59	0.19	0.16
PLSA-WORDS	0.71	0.56	0.20	0.14
PLSA-FUSION	0.76	0.65	0.22	0.19
Double-layer PLSA	0.74	0.70	0.25	0.20

other algorithms, and the average recall is 2% less than that of PLSA-FUSION. In the set of 260 labels, the double-layer PLSA model outperforms all the other algorithms, which exceeded the PLSA-FUSION 3% and 1% on the average recall and precision, respectively.

6. Conclusion

In this paper, we analyzed the image content from the perspective of text and proposed an image multilabel annotation model based on a double-layer PLSA model. The low-level features of images are represented by BoW model, which converted continuous visual information into discrete visual histograms to represent the visual content of the image. Then the first-layer PLSA was used, respectively, in the label aspect and visual aspect to get two kinds of latent semantic topics. In the second-layer, PLSA was applied on these two unrelated latent semantic topic spaces to get the top-layer latent topics, which can create the connection between the visual features and labels. Finally, with the double-layer PLSA model, the image annotation was completed effectively. In order to prove the effectiveness of the double-layer PLSA model in image annotation, we completed experiments on Corel5K and compared with other related algorithms. The experimental results illustrate that the double-layer PLSA model can achieve outstanding performance for multilabel automatic annotation and outperform other related algorithms.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research has been supported by the National Nature Science Foundation of China (Grant 61370174) and Nature Science Foundation of Shanghai Province of China (11ZR1409600).

References

[1] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1349–1380, 2000.

[2] D. M. Blei, M. David, Y. Andrew, and I. Michael, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, no. 4-5, pp. 993–1022, 2003.

[3] T. Hofmann, "Unsupervised learning by probabilistic latent semantic analysis," *Machine Learning*, vol. 42, no. 1-2, pp. 177–196, 2001.

[4] Y. Wang, T. Mei, S. Gong, and X. S. Hua, "Combining global, regional and contextual features for automatic image annotation," *Pattern Recognition*, vol. 42, no. 2, pp. 259–266, 2009.

[5] P. Duygulu, K. Barnard, N. de Freitas, and D. Forsyth, "Object recognition as machine translation: learning a lexicon for a fixed image vocabulary," in *Computer Vision*, A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, Eds., vol. 2353 of *Lecture Notes in Computer Science*, pp. 97–112, Springer, Berlin, Germany, 2002.

[6] K. Kobus Barnard, P. Duygulu, N. de Freitas, D. Forsyth, D. Blei, and M. I. Jordan, "Matching words and pictures," *Journal of Machine Learning Research*, vol. 3, no. 2, pp. 1107–1135, 2003.

[7] J. Jeon, V. Lavrenko, and R. Manmatha, "Automatic image annotation and re-trieval using cross-media relevance modelsIn," in *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval (SIGIR '03)*, pp. 119–126, ACM Press, New York, NY, USA, 2003.

[8] V. Lavrenko, R. Manmatha, and J. A. Jeon, "A model for learning the semantics of pictures," in *Advances Neural Information Processing Systems*, S. Thrun, L. K. Saul, and B. Schölkopf, Eds., vol. 16, pp. 553–560, MIT Press, Cambridge, Mass, USA, 2004.

[9] J. Zhang and W. Hu, "Multi-label image annotation based on multi-model," in *Proceedings of ACM International Conference on Ubigiutous Information Management and Communication (ICUIMC '13)*, article 21, Kota Kinabalu, Malaysia, January 2013.

[10] D. M. Blei and M. I. Jordan, "Modeling annotated data," in *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval (SIGIR '03)*, pp. 127–134, ACM Press, 2003.

[11] F. Monay and D. Gatica-Perez, "Modeling semantic aspects for cross-media image indexing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 10, pp. 1802–1817, 2007.

[12] Z. X. Li, Z. P. Shi, Z. Q. Li, and Z. Z. Shi, "Automatic image annotation by fusing semantic topics," *Journal of Software*, vol. 22, no. 4, pp. 801–812, 2011.

[13] H. G. Akcay and S. Aksoy, "Automated detection of objects using multiple hierarchical segmentations," in *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS '07)*, pp. 1468–1471, Barcelona, Spain, July 2007.

[14] L. Zhuang, L. She, Y. Jiang, K. Tang, and N. Yu, "Image classification via semi-supervised pLSA," in *Proceedings of the 5th International Conference on Image and Graphics (ICIG '09)*, pp. 205–208, Xi'an, China, September 2009.

[15] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '99)*, pp. 50–57, 1999.

[16] D. Larlus, J. Verbeek, and F. Jurie, "Category level object segmentation by combining bag-of-words models and markov random fields," Institut National de Recherche en Informatique et en Automation, 2008.

[17] L. Wu, S. C. H. Hoi, and N. Yu, "Semantics-preserving bag-of-words models for efficient image annotation," in *Proceedings of the 1st ACM Workshop on Large-Scale Multimedia Retrieval and Mining (LS-MMRM '09)*, pp. 19–26, ACM, October 2009.

[18] N. Dalal and W. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, vol. 1, pp. 886–893, San Diego, Calif, USA, June 2005.

[19] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: spatial pyramid matching for recognizing natural scene categories," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, vol. 2, pp. 2169–2178, June 2006.

[20] S. L. Fang, R. Manmatha, and V. Lavrenko, "Multiple bernoulli relevance models for image and video annotation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '04)*, vol. 2, pp. 1002–1009, IEEE Computer Society Press, July 2004.

[21] J. Zhang, Y. Zhao, D. Li, Z. Chen, and Y. Yuan, "Representation of image content of image content with multi-scale segmentation," in *Proceedings of the IEEE International Conference of Machine Learning and Cybernetics (ICMLC '13)*, pp. 14–17, Tianjin, China, July 2013.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 849069, 12 pages
http://dx.doi.org/10.1155/2014/849069

Research Article

An Adaptive Superpixel Based Hand Gesture Tracking and Recognition System

Hong-Min Zhu and Chi-Man Pun

Department of Computer and Information Science, University of Macau, Macau

Correspondence should be addressed to Chi-Man Pun; cmpun@umac.mo

Received 24 March 2014; Accepted 11 May 2014; Published 27 May 2014

Academic Editor: Yu-Bo Yuan

We propose an adaptive and robust superpixel based hand gesture tracking system, in which hand gestures drawn in free air are recognized from their motion trajectories. First we employed the motion detection of superpixels and unsupervised image segmentation to detect the moving target hand using the first few frames of the input video sequence. Then the hand appearance model is constructed from its surrounding superpixels. By incorporating the failure recovery and template matching in the tracking process, the target hand is tracked by an adaptive superpixel based tracking algorithm, where the problem of hand deformation, view-dependent appearance invariance, fast motion, and background confusion can be well handled to extract the correct hand motion trajectory. Finally, the hand gesture is recognized by the extracted motion trajectory with a trained SVM classifier. Experimental results show that our proposed system can achieve better performance compared to the existing state-of-the-art methods with the recognition accuracy 99.17% for easy set and 98.57 for hard set.

1. Introduction

Being a significant part in interaction of communication in our daily life (human-human or human-computer), hand gestures provide us a natural and user friendly way of interaction. With the progress of gesture tracking and recognition techniques, the computer vision field has experienced a new opportunity of applying a practical solution for building a variety of systems [1, 2] such as surveillance, smart home, and sign language recognition. Early systems that make use of gestures as interaction usually require an additional pointing device (e.g., data gloves and markers) to detect the movement; these sensor-based solutions can provide accurate measurements of hand pose and movement while they require extensive calibration, restrict natural hand motion, and are usually expensive. Recent systems focused on gestures performed by hand freely in 3D space without any physical attachments, and gestures are captured by various cameras which are analyzed and recognized with video-based solutions. Locating the hands and segmenting them from the background usually encounter difficulties when there are occlusions, lighting variances, fast motion, or other objects present with similar appearance. There are many vision-based hand gesture recognition algorithms proposed in past several decades which attempted to provide robust and reliable systems, as reviewed in [3, 4]. The common methods for hand detection are skin-color maps [5] and cascaded classifiers on Haar-like features [6]. Skin-color based approaches may be easily affected by lighting changes. Another set of hand detection approaches are clustering [7] and region growing [8] which are both time consuming processes. The hand tracking solution can benefit from visual object tracking solutions [9–13] which are based on cues ranging from low-level visual features to high-level structural information. The PROST method [9] extends the idea of tracking-by-detection such as [10] with multiple modules to reduce the drifts and object deformation; however the tracker is easily distracted by object with similar appearance. The visual tracking decomposition approach (VTD) [11] gets the tracking result with significant amount of noise from the background patches which combined particle filter with multiple observation and motion models; the tracker encounters failures when distinguishing the target object and its background. Spatiotemporal structural context based tracker (STT) [12] captured the historical appearance information to prevent the target object from drifting to the background in a long sequence; the supporting field built from spatial contributors provides more information to predict the target. Another potential solution is superpixel tracking (SPT) [13], which used mid-level clustering of histogram information captured in superpixels and a discriminative appearance model formulated with target-background confidence map, which tried to find proper appearance models that distinguish one object with all other targets or background. However, this approach is not very reliable when severe deformation or background confusion exists. In the area of hand gesture recognition, there are less works relayed on hand's motion trajectories, compared to gestures represented by palm and finger's appearance and motions. Alon et al.'s work [1] proposed a classifier-based pruning framework for early rejecting of the poor matches, and a subgesture reasoning algorithm to identify falsely matched parts in longer gestures; however they detect the hand location in each frame independently with color and motion information and the appearance changes are not adaptively learnt, the multiple hand region candidates may cause confusion between the palm and the arm.

In this paper an adaptive superpixel based hand gesture tracking and recognition system was proposed, in which hand gestures drawn in free air are recognized from the extracted motion trajectory. The overall system framework is shown in Figure 1. With the given input video sequence, the moving target hand is first detected to construct its appearance model by the proposed Initial Hand Detection and Model Construction algorithm using the first few video frames. Then the hand gesture motion trajectory is tracked by the proposed Adaptive Hand Gesture Tracking algorithm. Finally the normalized B-Spline feature vector is extracted from motion trajectory and fed to a trained SVM classifier to output recognized hand gesture. The rest of the paper is organized as follows. In Section 2 we describe the details of our proposed Initial Hand Detection and Model Construction algorithm. In Section 3, the proposed Adaptive Hand Gesture Tracking algorithm will be described. Then the procedure of feature extraction and classification is introduced in Section 4. Experimental results are given and discussed in Section 5, and finally the conclusions are drawn in Section 6.

2. Initial Hand Detection and Model Construction

As shown in Figure 1, the first step of our proposed hand gesture recognition system is to detect the moving target hand and construct its appearance model. In order to locate the position of the moving target hand, we employed the motion detection of superpixels and unsupervised image segmentation on the first few frames of the input video sequence. The simple linear iterative clustering (SLIC) superpixels [14] solution has been widely used in the area of image segmentation and object recognition with some good results; the method over-segments the image into numerous superpixels of which object regions are composed, and the boundaries are not significantly destroyed. We employed the SLIC superpixel as the slight hand motion, which can be detected from corresponding superpixels changes in between adjacent frames. The first frame I_1 is segmented into P superpixels S_p (Figure 2(a)), from which the object boundaries are approximated. The accumulated intensity changes D_p of each superpixel S_p between I_1 and I_i can be computed as

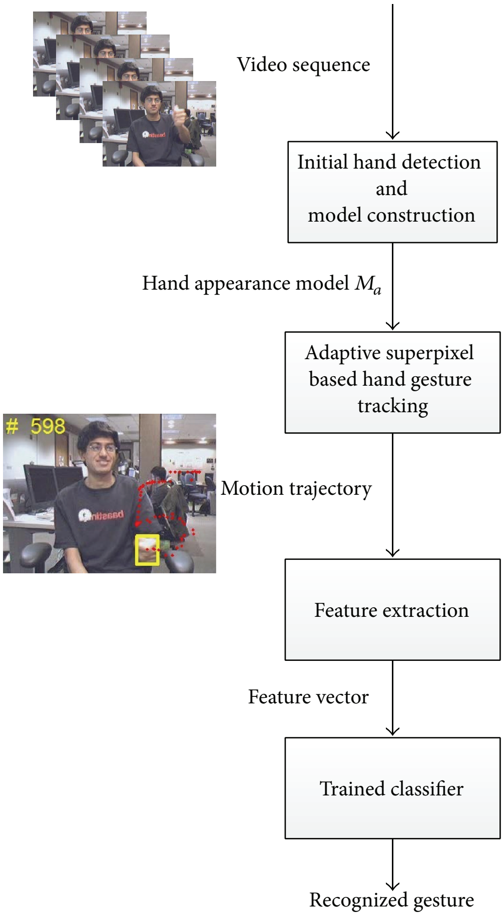

Video sequence
Initial hand detection and model construction
Hand appearance model M_a
Adaptive superpixel based hand gesture tracking
# 598
Motion trajectory
Feature extraction
Feature vector
Trained classifier
Recognized gesture

FIGURE 1: Overall framework of the proposed hand gesture recognition system.

$$D_p = \sum_{i=2}^{M} \left| I_i\left(S_p\right) - I_1\left(S_p\right) \right|, \quad p = 1, \ldots, P. \tag{1}$$

And the slight motion of a superpixel is detected (Figure 2(b)) if

$$\frac{D_p}{\left|S_p\right|} > T_0, \tag{2}$$

where T_0 is a threshold of the normalized distance and $|S_p|$ is the size of pth superpixel. After we merged neighbored superpixels with intensity changes as R candidate regions of the hand (Figure 2(c)), we used the compression-based texture merging (CTM) [15] based image segmentation to select the hand region from candidates. CTM used lossy compression-based clustering of texture features for the

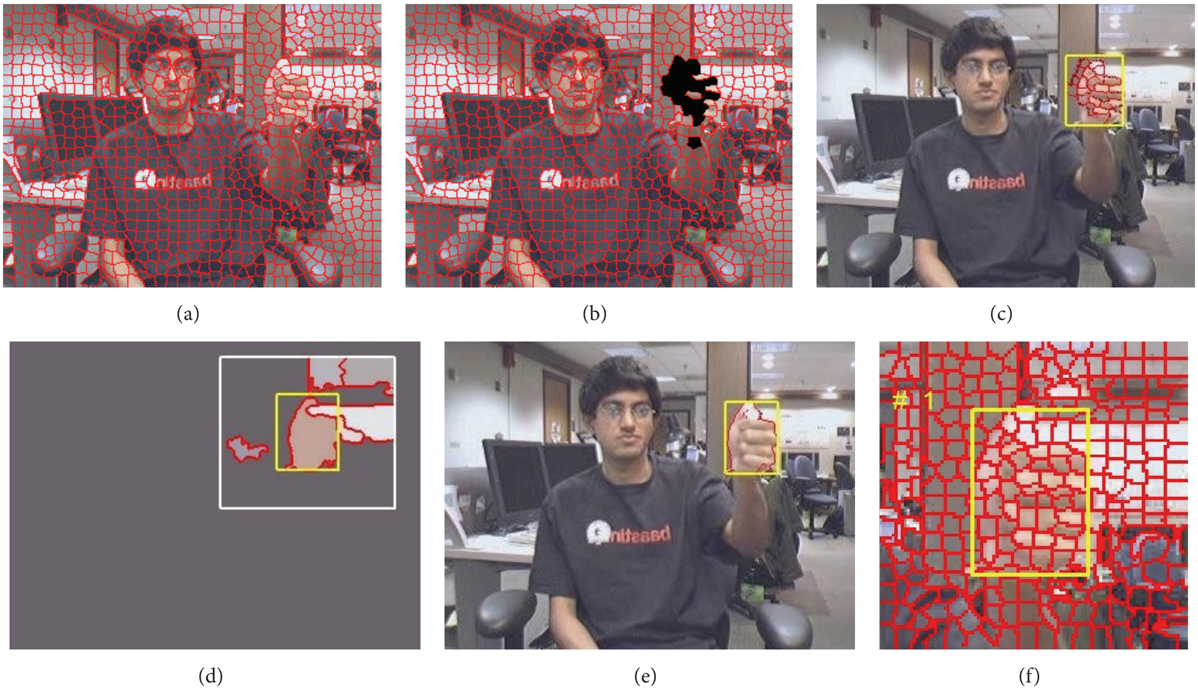


Figure 2: SLIC and CTM hand detection. (a) SLIC superpixels on the first frame. (b) Superpixels with slight motions. (c) Candidate hand region on the connected superpixels. (d) CTM objects on the surrounding of candidate hand region. (e) Refined hand region. (f) Superpixel on the surrounding of the hand region.

superpixels which are merged to form the object regions. The texture is modeled with a mixture of Gaussian distributions which can be degenerated; the approach shows precise segmentation on various images. We used the SLIC superpixel approach instead of the superpixel solution used in CTM. We get the K CTM object regions O_k with areas A_k $(k = 1, \ldots, K)$ on the surrounding of candidate hand region (twice the area size) in the first frame (Figure 2(d)), and the region with maximum percentage of the region area overlapped with hand candidates R is stated as detected hand R_H (Figure 2(e)):

$$R_H = O_k \mid A_k = \max\left(\frac{\operatorname{size}(R \cap O_i)}{\operatorname{size}(R \cup O_i)}\right), \quad i = 1, \ldots, K. \tag{3}$$

As we can see from the example, motion detection based on SLIC superpixels locate the hand region as shown in Figure 2(c), which include the region besides the left side of the hand since the hand moves from right to left in this case. The result is then refined by CTM segmentation to exclude the false region part. The initial hand detection is represented by a bounding box of the hand region in the first frame, although the motion information with changed intensity is accumulated from the first M frames.

With the gesture hand detected in the first frame, we use a simple strategy to track the hand in first M frames (except the first frame) and construct an initial hand appearance model. Let $X_{t=1}$ be the hand location in the first frame (Figure 2(e)) which is represented by center of the hand region and its scale; we sample N hand candidates around $X_{t=1}$ in each frame t $(t = 2, \ldots, M)$ and the similarity between each candidate X_t^n $(n = 1, \ldots, N)$ and $X_{t=1}$ is

$$\begin{gathered} S(X_1, X_t^n) = \frac{S(X_1, X_t^n)}{\sum_{i=1}^{N} S(X_1, X_t^i)}, \\ \text{where} \quad S(X_1, X_t^n) = \exp\left(\frac{-\sum (I_1 - I_t^n)^2}{c}\right), \end{gathered} \tag{4}$$

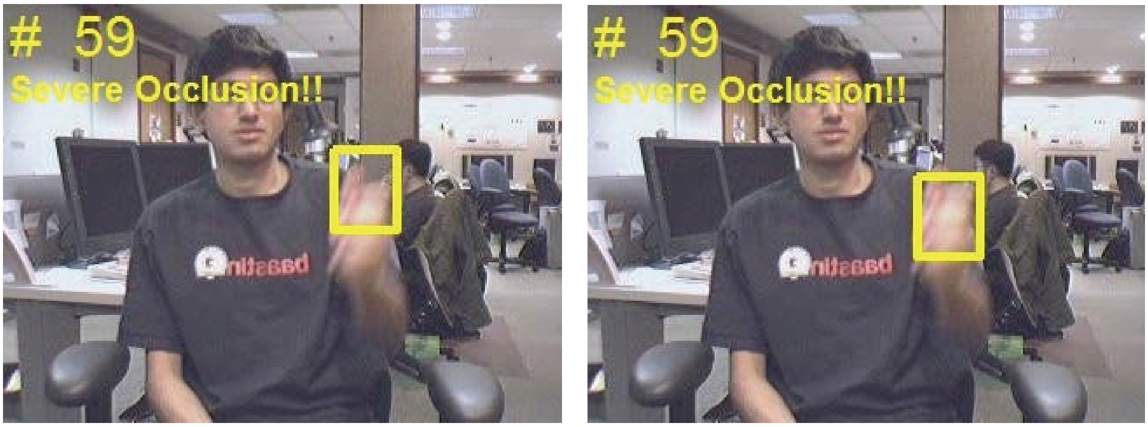
where I_t^n is the grayscale image patch of X_t^n, and c is the condensation constant parameter. The hand detection X_t is selected with maximal similarity. Then SLIC segmentation on the surrounding region of X_t gets the P_t superpixels (as in Figure 2(f)) in and the $YCbCr$ histogram f_t of each superpixel is calculated; here surrounding region is a square area centered at the same location as X_t and with size greater than X_t. Our targeted hand gestures are captured in indoor environment that the color appearance of the hand is greatly affected by lighting changes which makes the feature of the hand unstable. The $YCbCr$ color space encodes the illumination information in the separated component Y, which reduces the lighting problem by using the only Cb and Cr components. The accumulated feature set $\{f_t^r\}_{r=1}^{P}$ from M frames is clustered with mean shift clustering. The initial appearance model is then trained by calculating the target-background confidence for each cluster i:

$$C_i^c = \frac{\operatorname{Size}^+(i) - \operatorname{Size}^-(i)}{\operatorname{Size}^+(i) + \operatorname{Size}^-(i)}, \quad \forall i = 1, \ldots, n, \tag{5}$$

```
Initial hand detection
Input: $M$ frames $I_i \in \mathbb{R}^{H\times W}$, $i \in [1, M]$
(1) Segment $I_1$ into $P$ superpixels $S_p$, $(p = 1, \ldots, P)$ with SLIC.
(2) For each frame $I = I_2$ to $I_M$, Compute $D_p$ for each $S_p$ using (1).
(3) Detect $m$ superpixels $P_m$ $(m < P)$ with motions using (2), merge neighbored superpixels to get $R$ regions.
(4) Do CTM segmentation on surrounding of $R$ regions in $I_1$, get object regions $O_1, \ldots, O_K$ with area $A_1, \ldots, A_K$.
(5) Find the hand region from $R$ and $O_k$ regions using (3).
Output: $X_1$ (center of hand region and its scale in the first frame).

Hand appearance model construction
Input: $M$ frames $I_i \in \mathbb{R}^{H\times W}$, $i \in [1, M]$ and $X_1$
(1) For each frame $I_t$, $t = 2, \ldots, M$, detect the hand $X_t$ from $N$ candidates around $X_{t-1}$ using (4).
(2) For each frame $I_t$, $t = 1, \ldots, M$, Extract $\{f_t^r\}_{r=1}^{P}$ as the histogram in $YCbCr$ of $P$ superpixels from SLIC
    segmentation on surronding of $X_t$.
(3) Apply mean shift clustering on feature set $F = \{f_t^r \mid t = 1, \ldots, m; r = 1, \ldots, P_t\}$ to get $f_c(i)$, $r_c(i)$ and $\{f_t^r \mid f_t^r \in i\}$.
    Calclute $C_i^c$ using (5).
Output: Hand appearance model $M_a = \{C_i^c, f_c(i), r_c(i), \{f_t^r \mid f_t^r \in i\}\}$.
```

Algorithm 1: Initial hand detection and model construction.



(a)

(b)

Figure 3: Typical example results. (a) Occlusion occurred in SPT; (b) occlusion recovered in our hand tracking solution.

where Size$^+$ is the size of cluster i overlapping the object (area of X_t) and Size$^-$ is the size of i outside the object. Finally the hand appearance model is measured by cluster confidence C_i^c, cluster centers $f_c(i)$, cluster radius $r_c(i)$, and cluster members $\{f_t^r \mid f_t^r \in i\}$.

The initial hand detection and model construction procedure is summarized in Algorithm 1.

3. Adaptive Superpixel Hand Gesture Tracking

After the initial hand appearance model is constructed from the first few frames, the positions of the target hand need to be tracked in following video frames to obtain the motion trajectory for gesture classification. Object tracking has been widely studied [9–13] in the past decade with successful results. However, these tracking techniques are not very robust for hand tracking, especially when there exist hand deformation, appearance changes, fast motion, and background confusion. In order to tackle these problems, we employed an adaptive superpixel based hand gesture tracking approach. The existing superpixel tracking (SPT) method [13] proposed for general object tracking frequently encounters failures in our hand gesture tracking task. Figures 3, 4, and 5 give some typical examples that SPT fails to track the gesturing hand. We state that the *occlusion* in Figure 3(a) occurred when the match scores between the candidate hand region and the hand model below a threshold, which may be caused by hand deformation and blur of fast motion, but not necessarily by overlapping with other objects. The model updating strategy of SPT considers the contents inside the tracked hand region as foreground, which may introduce false information to the updated model when occlusion occurred. The first row in Figure 4 gives an example that SPT detects the background as the hand region when it is skin-color like. If the problem continuously appears, the appearance model will eventually be updated with features extracted from the background. The model cannot be recovered as the subsequent tracking

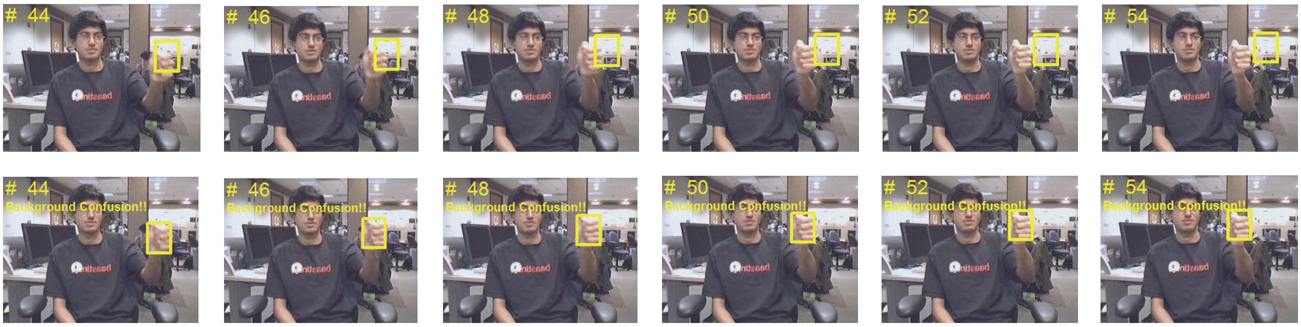
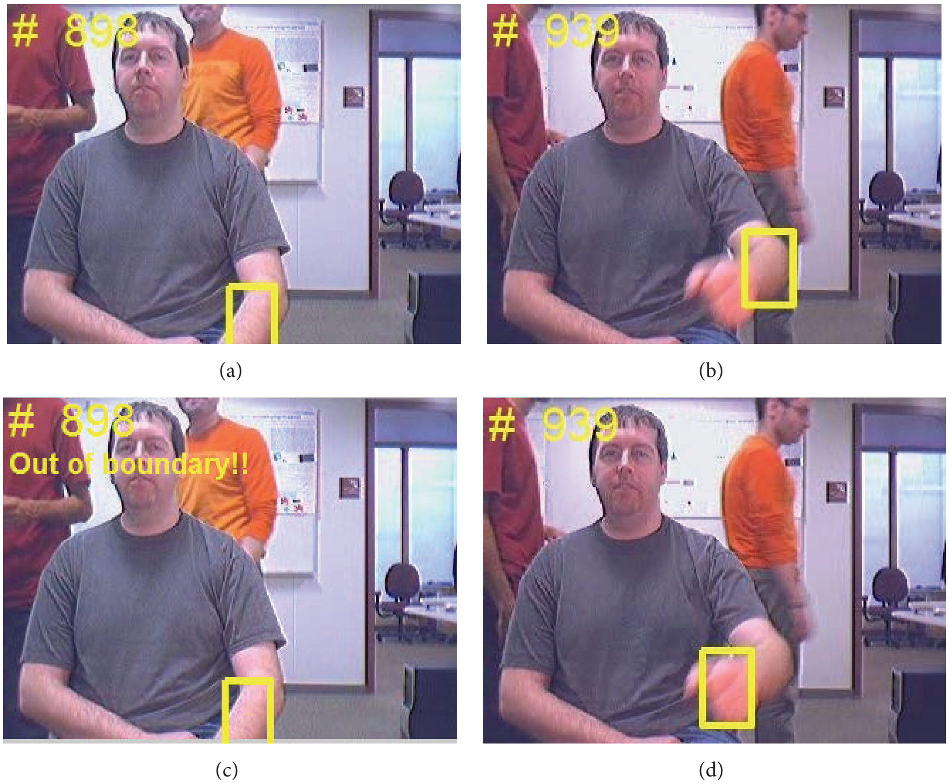


FIGURE 4: Typical example results. First row: background confusion occurred in SPT from frame 44 to frame 54. Second row: background confusion recovered in our hand tracking solution.



(a) (b)

(c) (d)

FIGURE 5: Typical example results. (a) Hand region disappeared in the scene. (b) Hand region tracked after it reappeared in SPT. (c) Detect the disappearance of hand region in our solution. (d) Hand region tracked after it reappeared in our solution.

will surely label the background as the target. We consider this problem as *background confusion*. Figures 5(a) and 5(b) show the example that if the target hand disappeared in the scene for a long period, the model will be updated with false information which is similar to *background confusion*, and the subsequent hand tracking will fail. Our proposed adaptive hand tracking solution recovers from these failures to provide reliable tracking results. Hand region candidates are prerefined by incorporating domain specific knowledge so that the retracking with template matching detects the hand more accurately.

In order to tackle the difficulties of hand deformation caused by the fast hand motion and confusion caused by background, we propose an adaptive superpixel based hand gesture tracking algorithm. Figure 6 summarizes the workflow of our proposed algorithm. Firstly we select hand detection from candidates by matching to the initial/updated model, in case any failure occurred as introduced in Figures 3, 4, or 5, we recover and retrack the hand with template matching to give positive detections. The detected hand will be continuously and periodically sampled and used to update the hand appearance model.

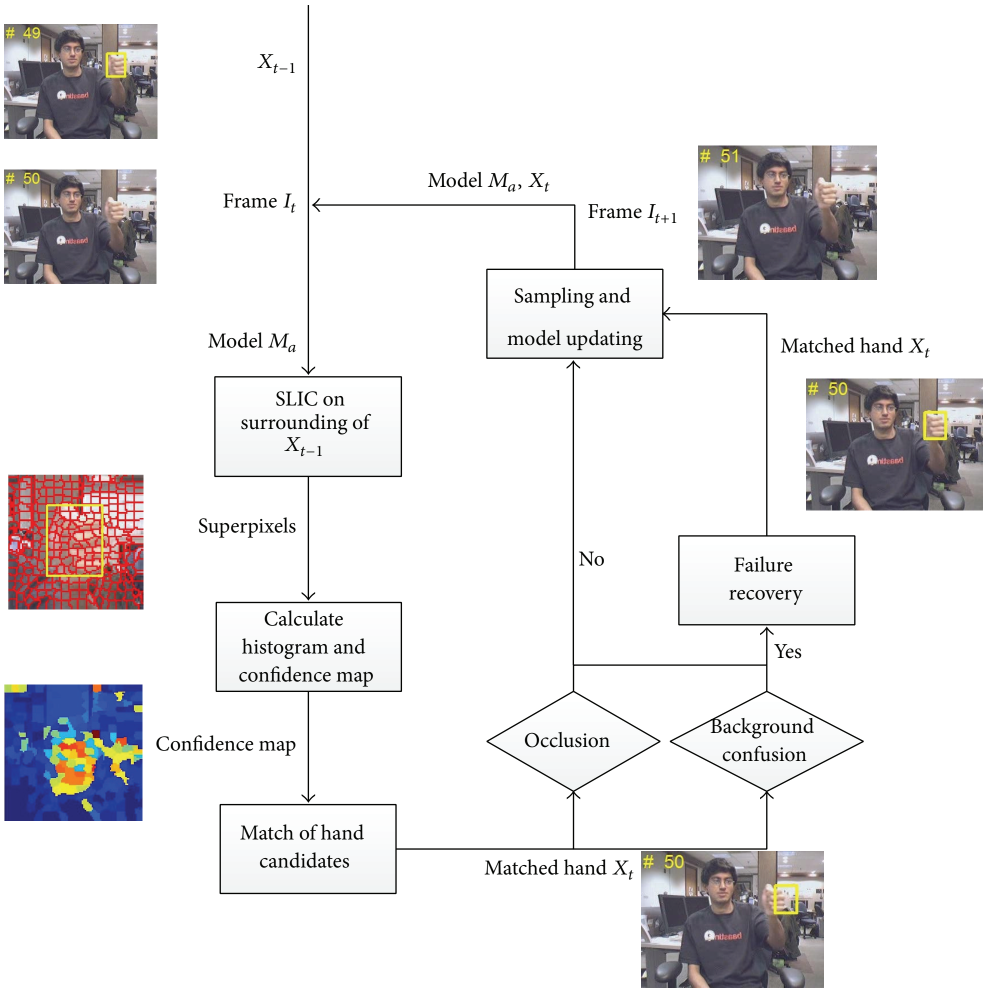


Figure 6: Workflow of the proposed adaptive superpixel based hand tracking.

From frame $t = M + 1$, the surrounding region of X_{t-1} is firstly segmented into P superpixels S_p, and then the confidence map C_r^s of each superpixel r can be computed from its histogram f_t^r of $YCbCr$ and clusters in the model:

$$\omega(r,i) = \exp\left(-2 \times \frac{\left\| f_t^r - f_c(i) \right\|^2}{r_c(i)}\right), \quad \forall r = 1,\ldots,P_t; \quad i = 1,\ldots,n, \tag{6}$$

$$C_r^s = \omega(r,i) \times C_i^c, \quad \forall r = 1,\ldots,P_t,$$

where $f_c(i)$ is the feature center of the cluster i that superpixel r belongs to, and $r_c(i)$ is the radius of feature space of cluster i.

We sample N hand candidates around X_{t-1} and we discard those candidates that the contents of samples are occupied by non-skin-like objects:

$$\frac{\sum SK_t}{\text{size}(SK_t)} < a, \quad SK_t = \left(B_t^c \in R_s\right), \tag{7}$$

where R_s is the interval of skin color region that is defined by a Gaussian model in $YCbCr$, SK_t is the binary skin image of cth sample candidate bounding box B_t^c, and a is a threshold. We also discard candidates that there's no object motion detected inside the regions compared to previous frame:

$$\frac{\sum S_t}{\text{size}(S_t)} < b, \quad S_t = x \text{ or } \left(S_t, S_{t-m}\right) \& \left(\sim S_{t-m}\right), \tag{8}$$

where S_t and S_{t-m} are the skin images of the same candidate location in (7) at time t and $t - m$, and b is a threshold.

For each remaining candidates X_t^n we calculate the motion parameters $p(X_t^n | X_{t-1})$ as Gaussian distribution

$$p\left(X_t^n \mid X_{t-1}\right) = \mathbb{N}\left(X_t^n; X_{t-1}, \Psi\right), \tag{9}$$

where Ψ is a diagonal covariance matrix of the standard deviations of location and scale. The likelihood C_t^n of each X_t^n is an accumulation of confidence C_r^s of superpixels r located inside X_t^n

$$C_t^n = \frac{S\left(X_t^n\right)}{S\left(X_{t-1}\right)} \times \sum_{r \in [1,P]} C_r^s, \tag{10}$$

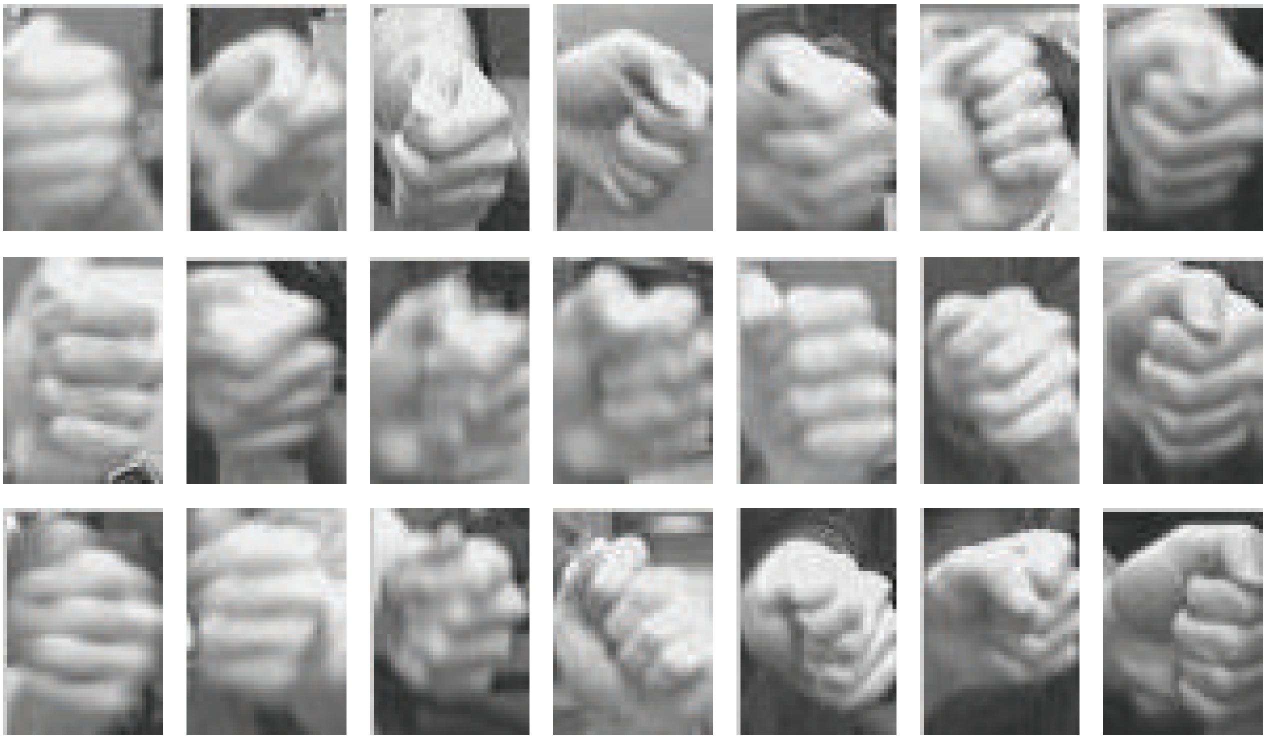


Figure 7: Gesture hand templates.

where $S(X_t)$ is the scale of hand X_t and the hand is detected as the best candidate according to the maximum a posteriori (MAP) estimate:

$$X_t = \underset{X_t^n}{\arg\max}\; p\left(X_t^n \mid X_{t-1}\right) C_t^n. \quad (11)$$

As we have discussed, the SPT may fail when *occlusion* or *background confusion* occurred. We recover from both failures to give more precise tracked hand and provide the positive samples to ensure updating with correct information. The only case discarded for sampling in our solution is the gesturing hand moves out of the frame, as shown in Figure 5(c). In our failure recovery process, we use the template matching to find the best match from the candidates. Figure 7 shows some hand templates which are automatically sampled during tracking with the occlusion rate of detection lower than a threshold. Compared to SPT which used only one hand template from the first frame, our template matching is adapted to different hand appearance to recover from the failure.

With remaining M sample candidates after discarding and N hand templates, we calculate the similarity between each pair of candidate and template using (4). And the best candidate matched to a hand template can be selected with maximum in $M \times N$ similarity matrix. Figure 3(b) shows an example of *occlusion* recovery which occurred in Figure 3(a); we can see that the hand location is more precisely detected, and the annotation "*Severe Occlusion*" indicated that it is a track result recovered from *occlusion* failure. We consider that the problem of *background confusion* occurs when the standard deviation of the recent L detected hand locations below a threshold T:

$$\operatorname{std}\left(\underset{i=t-L+1}{\overset{L}{X}}\right) < T. \quad (12)$$

Then we trace back to the time $t - L + 1$ and retrack each of K frames ($K < L$ and $K \leq H$, where H is the number of stored sampling frames used for updating the model) with the same method as for *occlusion* recovery. The appearance model may be updated with all samples from the period of *background confusion* which occurred (e.g., $L/U > H$ and $L > W$, U is the frequency of sampling and W is the frequency of updating), so we temporally set $U = 1$ and train the new model with all detections from the recovery of *background confusion*. The second row of Figure 4 shows an example of recovery of *background confusion*. Our proposed adaptive superpixel based hand tracking method tracks a frame in about 2.1 seconds with an Intel i7 CPU and 4 GB memory PC running Windows 7, where the SLIC segmentation is the main time consuming process.

The *First-In-First-Out* (FIFO) sampling strategy is used in SPT to discard the outdated hand detections, which may prematurely delete samples with high confidence. We try the deletion of samples considering the confidence of current detection, for chronologically stored samples $S_1, \ldots, S_H$; the sample S_h with confidence C_h is replaced by X_t with confidence C_t if S_h meets

$$\max\left(\frac{1}{2^h} \times \frac{H - h + 1}{H} \times \frac{1}{C_h}\right), \quad h = 1, \ldots, H \quad (13)$$

```
For each frame $t = M + 1$ to the end
Normal hand tracking
Input: frame $I_t$, $X_{t-1}$
(1) SLIC get $P$ superpixels $S_p$ on surrounding of $X_{t-1}$.
(2) For each superpixel $r$, Compute the$YCbCr$ histgram $f_t^r$, and confidence map $C_r^s$ using (6).
(3) Sample $N$ candidates $\{X_t^n\}_{n=1}^N$ around $X_{t-1}$ with $C_r^s$, discard unproper samples using (7) and (8).
(4) Calculate the motion parameter $p(X_t^n \mid X_{t-1})$ for each $X_t^n$ using (9).
(5) Calculate the likelihood $C_t^n$ for each $X_t^n$ using (10).
(6) Get the best match of hand $X_t$ with MAP estimate on $p(X_t^n \mid X_{t-1})$ and $C_t^n$ using (11).
Output: hand detection $X_t$ in frame $t$.

Failure recovery and updating

Input: current hand detection $X_t$
(1) Check the occurance of occlusion with threshold. Calculate $M \times N$ similarity matrix using (4). Detect the hand
    location $X_t$ to recover the occlusion.
(2) Check the occurance of background confusion using (12) and re-track $K$ frames to recover $X_k$.
(3) In case of background confusion, sample all detections of re-tracking ($U = 1$) and discard all previous samples.
(4) In case of occlusion or no failure, use one sample for every $U$ frames to replace a previous sample using (13).
(5) Replace the appearance model every $W$ frames by re-train on new samples.
Output: recovered hand detection $X_t$ if normal hand tracking fails, and new hand appearance model $M_a$.
```

Algorithm 2: Adaptive superpixel based hand gesture tracking.

which indicates that the early sample (smaller h) and sample with smaller confidence has more probability to be replaced. The new hand appearance models is retrained by performing mean shift clustering on updated sample set and recalculated the target-background confidence using (5).

Our adaptive superpixel based hand gesture tracking solution is summarized as in Algorithm 2.

4. Gesture Classification

With the gesture motion trajectories tracked by our proposed adaptive superpixel based hand gesture tracking algorithm, the normalized feature vector is extracted from motion trajectory for classifying the hand gesture. We applied multiclass support vector machines (SVM) to classify the gestures due to its property of discrimination on nonlinearly separable feature and efficiency. The duration of the hand gestures depends on their complexity, which caused the tracked motion trajectories with different lengths. We employed the B-form Spline approximation to interpolate the trajectories to a uniformed length as the SVM deals with feature instances of the unified dimension. Given a 2D trajectory with N points $\{X_i, Y_i\}_{i=1}^N$, we interpolate the two dimensions X_i and Y_i to N_1 points independently. For the case of X_i, we approximate the function defined by $\{i, X_i\}_{i=1}^N$ to a piecewise polynomial function $f(x)$ with order n:

$$f(x) = a_1 + a_2 x + \cdots + a_n x^{n-1} = \sum_{i=1}^{n} a_i x^{i-1}. \quad (14)$$

A Spline is a smoothed piecewise polynomial function that an interval $[a, b]$ (e.g., $[1, N]$) is divided into sufficiently small intervals $[\xi_i, \xi_{i+1}]$ with $a = \xi_1 < \cdots < \xi_{i+1} = b$. In each interval, a polynomial f_i of low degree can provide a good approximation to corresponding $\{i, X_i\}_{i=1}^N$. The B-form Spline describes the polynomial function as a weighted sum of order k:

$$f(t) = \sum_{i=1}^{n} B_{i,k}(t) \cdot a_i. \quad (15)$$

Each $B_{j,k}$ is defined on an interval $[\xi_i, \xi_{i+1}]$ and is zero elsewhere; t is called *knots* and is provided based on the smoothness required. B-splines are functions that

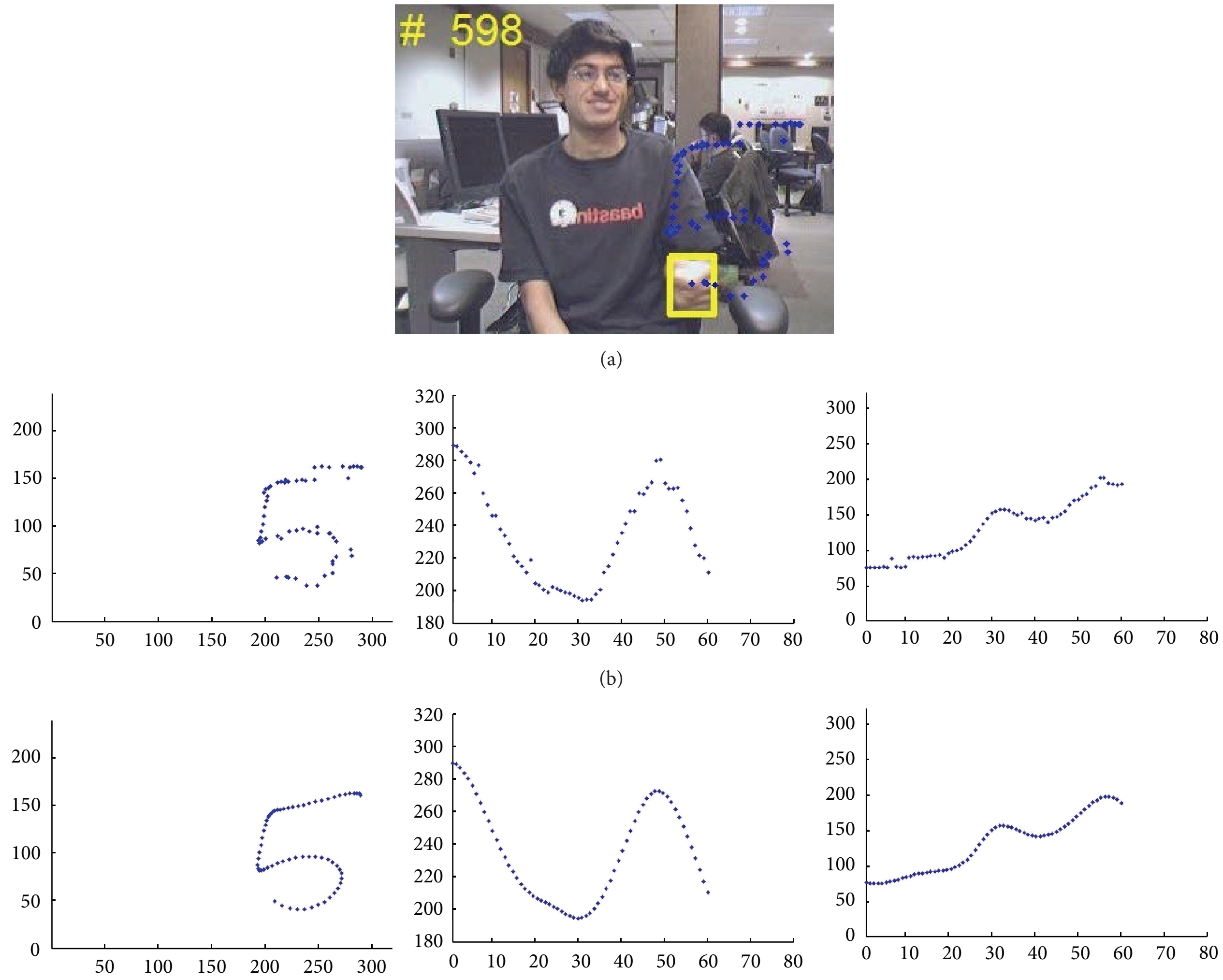
$$\sum_{i=1}^{n} B_{j,k}(x) = 1, \quad x \in [t_k, t_{n+1}]. \quad (16)$$

Figure 8 shows an example of trajectory interpolation on hand signed digit gesture "5". The second row shows the original tracked hand positions (60 points) and the third row shows the interpolated and smoothed trajectory (64 points). The first column is combined result of second and third columns, which are the interpolation of X and Y independently. We further normalize the trajectory points into the range of $[0, 1]$ as

$$X_i = \frac{X_i - \min\{X\}_1^{N_1}}{w}, \qquad Y_i = \frac{Y_i - \min\{Y\}_1^{N_1}}{h}, \quad (17)$$

where w and h are the sizes of the video frame.

We employed the SVM library from [16] for our multiclass hand gesture trajectories classification task, which used one-against-one approach to construct $k(k-1)/2$ classifiers that k is the number of gesture classes. A simple voting strategy is applied to decide the class of an input sequence in test. The two parameters c (cost of the quadratic problem) and g (gamma of RBF kernel) are optimized with 3-fold cross validation in the training set.



(c)

Figure 8: Trajectory interpolation. (a) Accumulated trajectory on the last frame of gesture "5"; (b) tracked trajectory; (c) interpolated and smoothed trajectory. Columns from left to right are trajectory plot with gesture's (X, Y); plot X of gesture trajectory; plot Y of gesture trajectory.

5. Experimental Results

In this section, our proposed adaptive superpixel based hand gesture tracking and recognition system were evaluated on the hand signed digit gesture dataset provided by Alon et al.'s work [1]; the dataset defined 10 classes of gesture from digit 0 to digit 9; Figure 9 gives a trajectory example for each class which is tracked with our Adaptive Superpixel Hand Tracking algorithm. There are three sets contained in the dataset, the *training set*, the *easy set,* and the *hard set*. We use only the *easy set* and the *hard set*, as the users in the *training set* (e.g., example frame in Figure 10(a)) wore colored gloves and long sleeve which simplifies the tracking from the confusion of skin-like objects. We do the cross validation inside the *easy set* (Figure 10(b)) and *hard set* (Figure 10(c)) to measure the performance of the system.

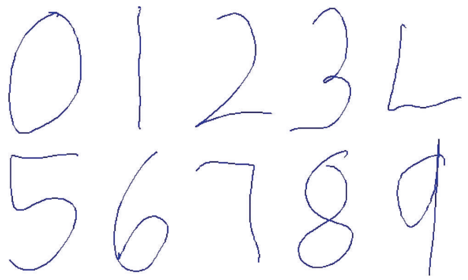
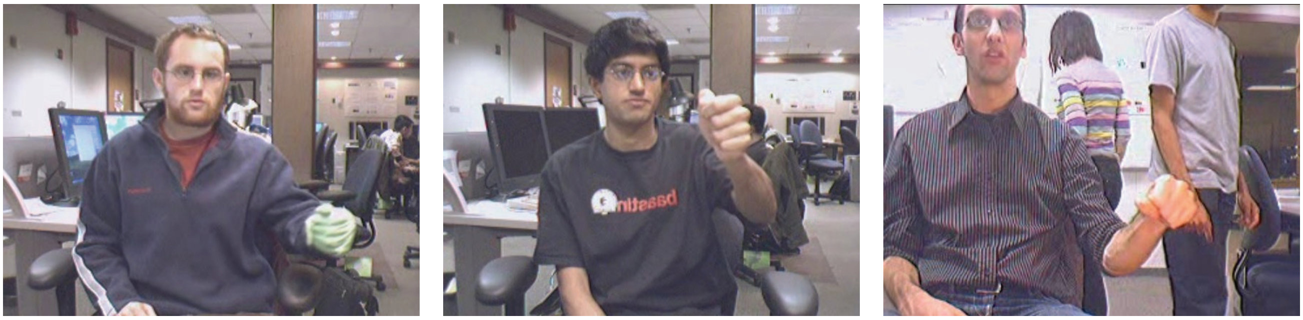


Figure 9: Ten classes of hand signed digit gestures.

5.1. Easy Test Set. The easy test set contains 30 video sequences, three from each of 10 users which are captured in office environment. The user signed each of 10 gestures once and wore short sleeves; totally there are 300 gesture instances in this set.

Firstly we use one sequences from each user for SVM training (100 gestures that 10 for each class), and test on the remaining sequences (200 gestures that 20 for each class). By



(a)

(b)

(c)

Figure 10: Sample frames from three gesture set.

Table 1: Confusion matrix of recognition result on easy set, using 1/3 data for training and 2/3 for testing. Gestures counts are accumulated from three tests by switch training/test data.

	0	1	2	3	4	5	6	7	8	9
0	*60*	0	0	0	0	0	2	0	0	0
1	0	*60*	0	0	0	0	0	0	0	0
2	0	0	*60*	0	0	0	0	0	0	0
3	0	0	0	*60*	0	0	0	0	0	0
4	0	0	0	0	*59*	0	0	0	0	0
5	0	0	0	0	0	*60*	0	0	0	1
6	0	0	0	0	0	0	*58*	0	0	0
7	0	0	0	0	0	0	0	*59*	0	0
8	0	0	0	0	0	0	0	0	*60*	0
9	0	0	0	0	1	0	0	1	0	*59*
False	0	0	0	0	1	0	2	1	0	1

switching the training/test video sequences, there are three tests. Table 1 gives the confusion matrix of the recognition results. The number of correctly and falsely recognized gestures for each class is accumulated from the three tests. The first row is the ground truth labels of gesture classes, and the first column is the recognized class labels. We see that totally 5 gestures are falsely classified out of 600 gestures from three tests; the recognition accuracy is 595/600 = 99.17%.

Similarly, we use two sequences from each user for SVM training (200 gestures that 20 for each class) and test on the remaining sequences (100 gestures that 10 for each class). There are totally 4 gestures misclassified out of 300 gestures from three tests. The recognition rate is 296/300 = 98.67%. Table 2 gives the confusion matrix of the results.

5.2. Hard Test Set. The hard test set contains 14 sequences, two from each of seven users; totally there are 140 gesture instances in this set. In this set there are one to three distractors moving around the gesturing user (see Figure 10(c)). We use half of the data (one sequence from each user, 70 gestures with 7 from each class) to train the SVM and test on the remaining. There are two tests by switching the training/test data. Table 3 shows the confusion matrix of recognition result for each class; there are only 2 gestures misclassified out of 140 gestures; the recognition accuracy is 138/140 = 98.57%.

We also compared our approach with the state of the art methods as shown in Table 4. To the best of our knowledge, we have referenced all publications that experiment the gesture recognition on the Alon et al.'s dataset [1]. We state that our hand gesture recognition approach outperforms the other solutions with significant improvement, which benefit mainly from our reliable hand motion tracking solution in long sequences.

6. Conclusion

We proposed an adaptive superpixel based hand gesture tracking and recognition system in this paper to address the gestures expressed by human hand motion trajectories. With the target hand detected in first few frames using SLIC segmentation and motion subtraction and then refined by segmented object regions of CTM, our adaptive hand motion tracking well handles the occlusion and background confusion problem. The trajectory classification using SVM models on hand signed digit gestures gives promising results. Experimental results show that our proposed system can achieve better performance compared to the existing state of the art methods with the recognition accuracy 99.17% for easy set and 98.57 for hard set. Future works may focus on multiobjects or two-hand gesture tracking system.

Table 2: Confusion matrix of recognition result on easy set, using 2/3 data for training and 1/3 for testing. Gestures counts are accumulated from three tests by switch training/test data.

	0	1	2	3	4	5	6	7	8	9
0	*30*	0	0	0	0	0	1	0	0	0
1	0	*30*	0	0	0	0	0	0	0	0
2	0	0	*30*	0	0	0	0	0	0	0
3	0	0	0	*30*	0	0	0	1	0	0
4	0	0	0	0	*29*	0	0	0	0	0
5	0	0	0	0	0	*30*	0	0	0	1
6	0	0	0	0	0	0	*29*	0	0	0
7	0	0	0	0	0	0	0	*29*	0	0
8	0	0	0	0	0	0	0	0	*30*	0
9	0	0	0	0	1	0	0	0	0	*29*
False	0	0	0	0	1	0	1	1	0	1

Table 3: Confusion matrix of recognition result on hard set, using 1/2 data for training and 1/2 for testing. Gestures counts are accumulated from two tests.

	0	1	2	3	4	5	6	7	8	9
0	*14*	0	0	0	0	0	2	0	0	0
1	0	*14*	0	0	0	0	0	0	0	0
2	0	0	*14*	0	0	0	0	0	0	0
3	0	0	0	*14*	0	0	0	0	0	0
4	0	0	0	0	*14*	0	0	0	0	0
5	0	0	0	0	0	*14*	0	0	0	0
6	0	0	0	0	0	0	*12*	0	0	0
7	0	0	0	0	0	0	0	*14*	0	0
8	0	0	0	0	0	0	0	0	*14*	0
9	0	0	0	0	0	0	0	0	0	*14*
False	0	0	0	0	0	0	2	0	0	0

Table 4: Comparsion with state of arts on the same datasets.

Approach	Accuracy of easy set (%)	Accuracy of hard set (%)
Correa et al. [17]	75.00	N/A
Malgireddy et al. [18]	93.33	N/A
Kulkarni [19]	N/A	80.71
Yao and Li [20]	95.67	86.43
Hanson [21]	100	76.40
Alon et al. [1]	94.60	85.00
Our	**99.17**	**98.57**

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors would like to thank the referees for their valuable comments. This research was supported in part by the Research Committee of the University of Macau (MYRG134-FST11-PCM and MYRG181-FST11-PCM) and the Science and Technology Development Fund of Macau (Project no. 034/2010/A2 and 008/2013/A1).

References

[1] J. Alon, V. Athitsos, Q. Yuan, and S. Sclaroff, "A unified framework for gesture recognition and spatiotemporal gesture segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 9, pp. 1685–1699, 2009.

[2] E. Sato, T. Yamaguchi, and F. Harashima, "Natural interface using pointing behavior for human-robot gestural interaction," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 2, pp. 1105–1112, 2007.

[3] G. R. S. Murthy and R. S. Jadon, "A review of vision based hand gesture recognition," *International Journal of Information Technology and Knowledge Management*, vol. 2, no. 2, pp. 405–410, 2009.

[4] S. Mitra and T. Acharya, "Gesture recognition: a survey," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 37, no. 3, pp. 311–324, 2007.

[5] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly, "Vision-based hand pose estimation: a review," *Computer Vision and Image Understanding*, vol. 108, no. 1-2, pp. 52–73, 2007.

[6] E.-J. Ong and R. Bowden, "A boosted classifier tree for hand shape detection," in *Proceedings of the 6th IEEE International*

Conference on Automatic Face and Gesture Recognition (FGR '04), pp. 889–894, May 2004.

[7] S. Malassiotis, N. Aifanti, and M. G. Strintzis, "A gesture recognition system using 3D data," in *Proceedings of the 1st International Symposium on 3D Data Processing Visualization and Transmission*, pp. 190–193, 2002.

[8] D. Droeschel, J. Stückler, and S. Behnke, "Learning to interpret pointing gestures with a time-of-flight camera," in *Proceedings of the 6th ACM/IEEE International Conference on Human-Robot Interaction (HRI '11)*, pp. 481–488, Lausanne, Switzerland, March 2011.

[9] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof, "PROST: parallel robust online simple tracking," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 723–730, June 2010.

[10] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-N learning: bootstrapping binary classifiers by structural constraints," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 49–56, June 2010.

[11] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 1269–1276, June 2010.

[12] L. Wen, Z. Cai, Z. Lei, D. Yi, and S. Z. Li, "Online spatio-temporal structural context learning for visual tracking," in *Proceedings of the 12th European Conference on Computer Vision*, pp. 716–729, Springer, Florence, Italy, 2012.

[13] S. Wang, H. Lu, F. Yang, and M.-H. Yang, "Superpixel tracking," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 1323–1330, November 2011.

[14] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[15] A. Y. Yang, J. Wright, Y. Ma, and S. S. Sastry, "Unsupervised segmentation of natural images via lossy data compression," *Computer Vision and Image Understanding*, vol. 110, no. 2, pp. 212–225, 2008.

[16] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, article 27, 2011.

[17] M. Correa, J. Ruiz-del-Solar, R. Verschae, J. Lee-Ferng, and N. Castillo, "Real-time hand gesture recognition for human robot interaction," in *RoboCup 2009: Robot Soccer World Cup XIII*, B. Jacky, Ed., vol. 5949 of *Lecture Notes in Computer Science*, pp. 46–57, Springer, 2010.

[18] M. Malgireddy, I. Nwogu, S. Ghosh, and V. Govindaraju, "A shared parameter model for gesture and sub-gesture analysis," in *Combinatorial Image Analysis*, J. K. Aggarwal, R. P. Barneva, V. E. Brimkov, K. N. Koroutchev, and E. R. Korutcheva, Eds., pp. 483–493, Springer, Berlin, Germany, 2011.

[19] A. Kulkarni, *Novel cost measures for robust recognition of dynamic hand gestures [M.S. thesis]*, University of Texas at Arlington, 2012.

[20] Y. Yao and C.-T. Li, "Real-time hand gesture recognition for uncontrolled environments using adaptive SURF tracking and hidden conditional random fields," in *Proceedings of the 9th International Symposium on Visual Computing (ISVC '13)*, pp. 29–31, Rethymnon, Crete, Greece, July 2013.

[21] D. A. Hanson, *Improving gesture recognition performance using the dynamic space-time warp algorithm [M.S. thesis]*, University of Texas at Arlington, 2013.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 105089, 7 pages
http://dx.doi.org/10.1155/2014/105089

Research Article

A High Accuracy Pedestrian Detection System Combining a Cascade AdaBoost Detector and Random Vector Functional-Link Net

Zhihui Wang,[1] Sook Yoon,[2] Shan Juan Xie,[3] Yu Lu,[1] and Dong Sun Park[1,4]

[1] *Department of Electronics Engineering, Chonbuk National University, Jeonju 561-756, Republic of Korea*
[2] *Department of Multimedia, Mokpo National University, Jeonnam 534-729, Republic of Korea*
[3] *Institute of Remote Sensing and Earth Science, Hangzhou Normal University, Hangzhou 311121, China*
[4] *IT Convergence Research Center, Chonbuk National University, Jeonju 561-756, Republic of Korea*

Correspondence should be addressed to Dong Sun Park; dspark@jbnu.ac.kr

Received 27 March 2014; Revised 4 May 2014; Accepted 5 May 2014; Published 19 May 2014

Academic Editor: Yu-Bo Yuan

In pedestrian detection methods, their high accuracy detection rates are always obtained at the cost of a large amount of false pedestrians. In order to overcome this problem, the authors propose an accurate pedestrian detection system based on two machine learning methods: cascade AdaBoost detector and random vector functional-link net. During the offline training phase, the parameters of a cascade AdaBoost detector and random vector functional-link net are trained by standard dataset. These candidates, extracted by the strategy of a multiscale sliding window, are normalized to be standard scale and verified by the cascade AdaBoost detector and random vector functional-link net on the online phase. Only those candidates with high confidence can pass the validation. The proposed system is more accurate than other single machine learning algorithms with fewer false pedestrians, which has been confirmed in simulation experiment on four datasets.

1. Introduction

Nowadays, pedestrian detection has drawn the attention of many researchers, due to its wide range of applications, such as driver assistant system [1–3], intelligent video surveillance system [4, 5], and victim rescue in case of emergency [6]. Numerous pedestrian detection algorithms have been proposed during the past decades, based on different techniques and strategies [7–10].

Pedestrians have properties of both rigid and flexible objects. Furthermore, the appearances of pedestrians are easily affected by view angle, occlusion, apparel, scale, pose variation, and illumination changes. All these issues have made pedestrian detection become a hot issue and one of the difficulties in the fields of computer vision. In current mainstream methods for pedestrian detection, machine learning algorithms are adopted to distinguish and identify pedestrians from candidates extracted by multiscale sliding windows. However, high accuracy detection rates of these algorithms are always obtained at the cost of a large amount of false pedestrians. These experiments show that high accuracy detection rates and low false positive rates are by no means simultaneously guaranteed.

The two-stage classifier [11], proposed by Guo et al., can further reduce false positive rates and this system has better performance than these single-stage algorithms. However, the detection rates cannot be further increased and maintained at a certain level as can these single-stage algorithms. In this paper, a novel two-stage detecting system is proposed based on a cascade AdaBoost detector [9] and random vector functional-link net [12, 13]. These two algorithms can simultaneously deal with the normalized candidates extracted by multiscale sliding windows, which can guarantee the detecting efficiency of the proposed system. These processing results of the cascade AdaBoost detector and random vector functional-link net are fused together, as the final evaluation criteria of whether these candidates are pedestrians or not. The cascade AdaBoost detector and random vector functional-link net are two of the significant high-efficient machine learning algorithms. They have both been applied in many research fields, such as multimedia processing, natural language processing, biological information processing, and network security.

The proposed system can achieve high accuracy detection rates on the basis of low false positive rates, which is benefited from the joint promotion of the cascade AdaBoost detector and random vector functional-link net. The high performance of the proposed system has been demonstrated on four datasets, with different types, during our simulation experiments. The remainder of the paper is organized as follows. We start by introducing the structure of the proposed system in Section 2, and the experimental comparison of the proposed system with other state-of-the-art detectors is demonstrated in Section 3. Finally, we summarize the characteristics of the proposed system and discuss its superiority over other detectors in Section 4.

2. Proposed Pedestrian Detection System

As there is seldom any single detector that can reach excellent performance with high detection rate and few false positives in complex scenarios, the proposed pedestrian detection system is based on machine learning algorithms. The cascade AdaBoost (CAB) detector [9] and random vector functional-link (RVFL) net [12, 13] have been employed and combined to enhance the corresponding performance of detection results.

2.1. System Architecture. The flow chart of the proposed pedestrian detection system is demonstrated in Figure 1. The proposed system contains the off-line training phase and the on-line detecting phase. During the off-line training phase, the CAB detector and RVFL net are trained separately with the given training dataset. Each training sample has the same size, called the standard size, which is demonstrated in Section 3. The CAB detector is trained as classification pattern, while RVFL net is trained as regression pattern. For the classification pattern of CAB detector, the positive samples are labeled as 1 and negative samples are labeled as 0. During the training process of RVFL net with regression pattern, the confidence scale is limited in $[0, 1]$.

During the on-line detecting phase, all the subimages are generated by multiscale sliding windows, and they are resized to be the standard size as testing candidates. Then the CAB detector and RVFL net are employed to judge whether each candidate is a pedestrian or not. The CAB detector estimates whether each candidate is a pedestrian or not, and RVFL net estimates a confidence score for each candidate. Their two results are combined to get the final matching score and, finally, only those candidates with higher matching scores than the given threshold are regarded as pedestrians. The details of the proposed system are as follows.

2.2. Feature Extraction. Feature extraction is a type of dimensionality reduction that efficiently represents the ROI region of an image in the fields of object detection and pattern recognition algorithms. These features are extracted as a compact feature vector, for subsequent processing. Therefore, effective image feature extraction is rather important, which concerns final objection detection accuracy. Common features extraction techniques include the RGB histogram, local binary patterns (LBP) [14], histogram of oriented gradients (HOG) [7], Haar-like feature, first-order image statistics (the mean standard deviation, skewness, and kurtosis of pixel intensities), second-order image statistics (the mean and range of contrast, correlation, energy, and homogeneity) [15], and Hu's invariant matrix [16].

Past research has shown that, in the past researches, Haar-like and LBP features have been used for detecting faces, as they have desirable properties for representing fine-scale textures. And the HOG features, which can capture the overall shape of an object, have been used for detecting objects such as people and cars. In this paper, HOG features are adopted to enhance the pedestrian detection performance of the proposed system. In our experiment, the parameters for the HOG feature extraction applied to the CAB detector and RVFL net are the same. For our system, the normalized candidates are divided into 16×16 pixel blocks; each block contains 2×2 cells of 8×8 pixels; linear gradient voting into 9 orientation bins in $(0^\circ, 180^\circ)$. Therefore, the HOG features for CAB detector and RVFL net can be extracted in one step.

2.3. Cascade AdaBoost Detector (CAB). The cascade AdaBoost algorithm [9] is adopted, to detect object categories whose aspect ratio does not significantly vary. This algorithm consists of a series of classifiers, where each classifier is an AdaBoost learner and its parameters are adjusted utilizing a boosting algorithm. The flow chart of the cascade AdaBoost algorithm is illustrated in Figure 2. The expression of the cascade AdaBoost algorithm is formed as

$$H(\mathbf{x}) = \begin{cases} 1, & H_i(\mathbf{x}) = 1,\ i = 1, \dots, n; \\ 0, & \text{otherwise}, \end{cases} \tag{1}$$

where $\mathbf{x}$ is sample inputs, n is the number of stages, and H_i is the strong classifier of stage i, which can be represented as

$$H_i(\mathbf{x}) = \begin{cases} 1, & \sum_{j=1}^{m} \alpha_{ij} h_{ij}(\mathbf{x}) \geq \frac{1}{2} \sum_{j=1}^{m} \alpha_{ij}; \\ -1, & \text{otherwise}, \end{cases} \tag{2}$$

where m is the number of weak classifiers of each stage, h_{ij} is the jth weak classifier, and α_{ij} is the corresponding ensemble weight of h_{ij}. Suppose the total number of positive samples is N, and the minimum true positive rate is η; then the number of positive samples to use at each stage is calculated by

$$N_{\text{stage}} = \left\lfloor \frac{N}{1 + (n-1) \times (1-\eta)} \right\rfloor, \tag{3}$$

where $\lfloor \cdot \rfloor$ is the floor function. The number of negative samples for each stage is always set to be $2N_{\text{stage}}$, twice the positive samples.

During the training process, a certain amount of positive samples and negative images are required. The feature type

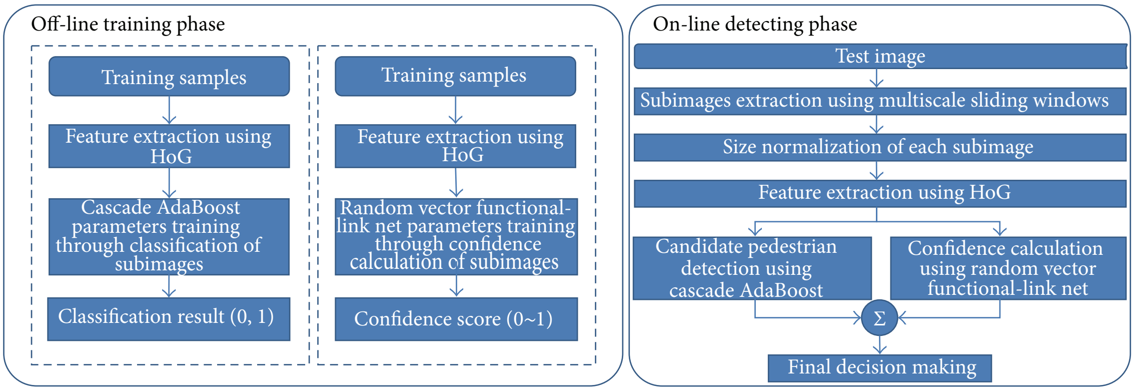

Off-line training phase
Training samples
Feature extraction using HoG
Cascade AdaBoost parameters training through classification of subimages
Classification result (0, 1)
Training samples
Feature extraction using HoG
Random vector functional-link net parameters training through confidence calculation of subimages
Confidence score (0~1)
On-line detecting phase
Test image
Subimages extraction using multiscale sliding windows
Size normalization of each subimage
Feature extraction using HoG
Candidate pedestrian detection using cascade AdaBoost
Confidence calculation using random vector functional-link net
Σ
Final decision making

Figure 1: Flow chart of the proposed pedestrian detection system. The proposed pedestrian detection system contains an off-line training phase and an on-line detecting phase. These parameters of CAB detector and RVFL net are trained on the off-line training phase with training samples. All these testing subimages are extracted and verified to be targets or not during the on-line phase.

and number of stages are set and other function parameters, which contain the minimum true positive and the maximum false alarm rates, are first initialized. Then, the parameters of each stage are estimated with partial positive and negative samples.

As mentioned above, true positives are usually not sufficiently given and worth taking the time to verify through the cascade stages. Furthermore, sufficient negative samples should be provided to ensure the training phase is carried out smoothly, and typical negative samples are supplied containing background information of the images to be detected. During the parameter estimation of each stage, the AdaBoost learner is trained by adding features, until the minimum true positive and the maximum false alarm rates are met. The number of stages is determined with proper final false positive and detection rates.

During the detection phase, as shown in Figure 2, all subwindows of the image are extracted through a multiscale sliding window. The structure of the cascade AdaBoost reflects that the vast majority of these subwindows are negative. As such, each stage of the cascade AdaBoost detector rejects the large possible number of nonpedestrian windows and lets potential targets pass to the next stage. Finally, only a few of these subwindows accepted by all stages of the detector are regarded as objects.

2.4. Random Vector Functional-Link (RVFL) Net. The random vector functional-link net [12, 13] is a special case of the single hidden layer feed-forward neural network. The hidden layer contains two different types of nodes: input patterns and enhancement patterns. Input patterns are simple linear combinations of sample inputs. These additional enhancements can be represented as $g(\mathbf{a}_j^t\mathbf{x} + b_j)$, where $\mathbf{a}_j$ is the weights of the input vector, b_j is the threshold parameter for the jth node, $\mathbf{x} = [x_1, \dots, x_N]$ is the sample inputs, and $g(\cdot)$ is the activation function. Therefore, the RVFL net can be interpreted as a mapping from N-dimensional space to $(J + N)$-dimensional space, where N is the dimensionality of training sample inputs, and separately, J is the number of additional enhancements. The output of the RVFL net can be represented as

$$f(\mathbf{x}) = \sum_{j=1}^{J} \beta_j g\left(\mathbf{a}_j^t\mathbf{x} + b_j\right) + \sum_{j=J+1}^{J+N} \beta_j x_j. \tag{4}$$

For the random vector functional-link net, $\mathbf{a}_j$ and b_j are randomly generated according to an appropriate given distribution (e.g., Gaussian distribution). Therefore, only the weight vector $\mathbf{B} = [\beta_1, \beta_2, \dots, \beta_{J+N}]$ needs to be learned, which largely reduces the time cost of the training phase. The optimal weight vector $\mathbf{B}$ is obtained by minimization of the system error

$$\mathbf{B} = \arg\min \left\{ \frac{1}{2P} \sum_{p=1}^{P} \left(t^{(p)} - \mathbf{B}\mathbf{d}^{(p)}\right)^2 \right\}, \tag{5}$$

where P is the number of training samples, $\mathbf{d} = [g(\mathbf{a}_1^t\mathbf{x} + b_1), \dots, g(\mathbf{a}_J^t\mathbf{x}+b_J), x_1, \dots, x_N]$ is the enhanced pattern vector, the subscript (p) is the sample index, and $t^{(p)}$ is the target value of the pth training sample.

The unique minimum of system error can be found by a learning phase, such as the conjugate gradient approach [17, 18]. If matrix inversion with the use of a pseudoinverse is feasible, then the optimal weight vector $\mathbf{B}$ is obtained by a single step, without any iteration. For this case, the pseudoinverse of optimal weight vector $\mathbf{B}$ was estimated by a single step in our experiments.

2.5. Matching Score Fusion. The proposed system deploys CAB and RVFL net to get more accurate detection rate. To obtain the final matching score for any subwindow, the proposed system fuses their two results: classification result $H(\mathbf{x})$, represented by 0 or 1, from CAB and confidence score $f(\mathbf{x})$, represented by continuous value with the range of $(0, 1)$, from RVFL net. Subimages with high matching scores can be accepted as objects. The function of matching score fusion is defined as

$$P_{\text{final}}(\mathbf{x}) = f(\mathbf{x}) + \lambda H(\mathbf{x}). \tag{6}$$

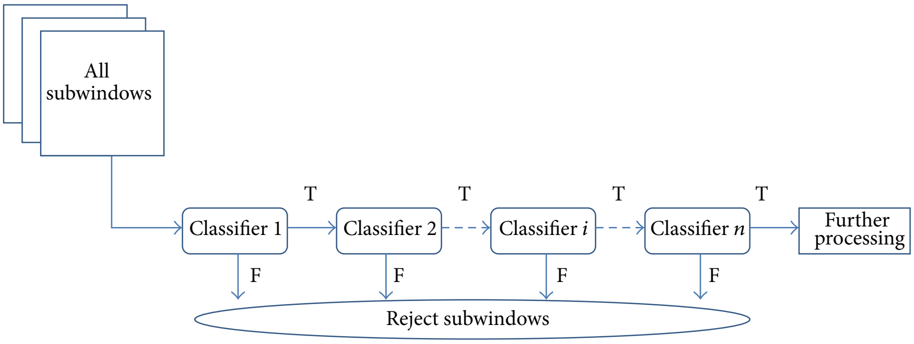


Figure 2: Flow chart of the cascade AdaBoost detector. Each classifier in cascade AdaBoost detector works independently, and the minimum true positive and the maximum false alarm rates of these stages are the same. Only these subwindows accepted as true positives by all stages of the detector are regarded as targets. "*T*" means that true candidates of these subwindows passed the verification of each classifier, and "*F*" means that these false candidates are rejected by the corresponding classifier.

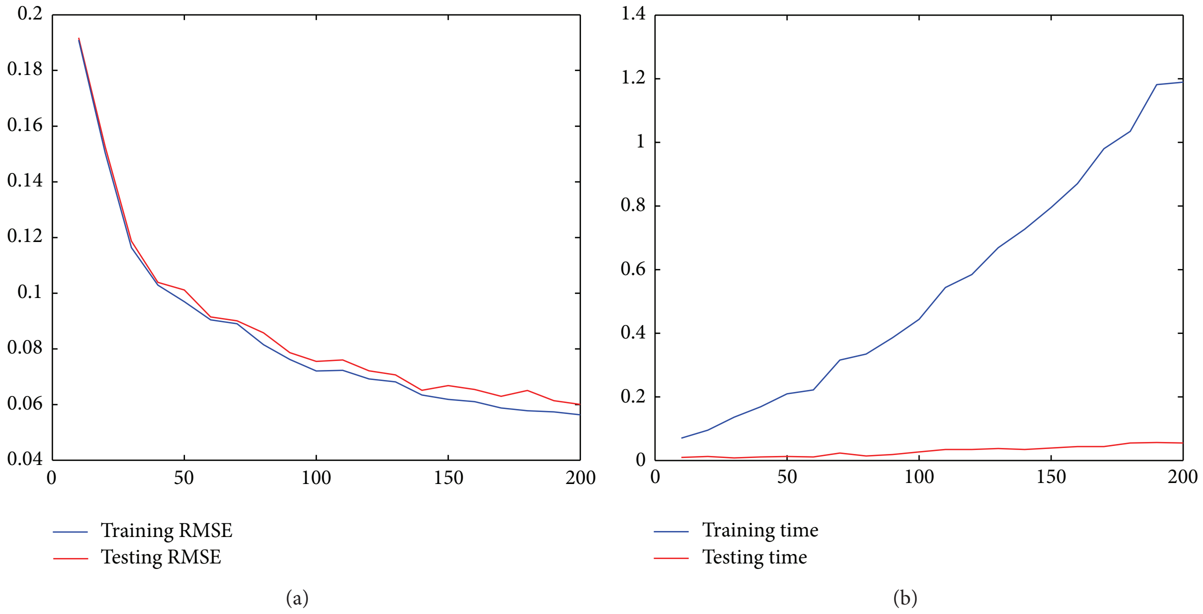


Figure 3: Performance comparison for RVFL net, with respect to hidden nodes. The x-axis of these two subimages represents the number of hidden nodes, which varied from 10 to 200 with step size 10. The left subimage is the training and testing RMSE and the right subimage is corresponding training and testing times.

With proper activation function, the enhancement patterns of RVFL net are more powerful than these input patterns, as the output of enhancement patterns has nonlinear correlation with sample inputs. During our experiment, only enhancement patterns of RVFL net are employed, and the activation function is set to be a sigmoid function. For this case, the final match score $P_{\text{final}}(\mathbf{x})$ in (6) can be simplified to be

$$P_{\text{final}}(\mathbf{x}) = \sum_{j=1}^{J} \beta_j g\left(\mathbf{a}_j^t \mathbf{x} + b_j\right) + \lambda H(\mathbf{x}). \tag{7}$$

3. Experiments

In this section, we compare our proposed two-stage detection system with four of the latest state-of-the-art detectors. To validate our proposed system, we have tested it on four publicly available sequences, which are PET'09 S3.MF (Multiple Flow) and PET'09 S0.CC (City Center) from PET benchmark [19], the "USC pedestrian set A" sequence from USC dataset [20], and the INRIA Person dataset [21]. The first two datasets are consecutive frames captured by one fixed camera, while the sequences of the latter two datasets are chosen from different scenarios. For the city center sequence, the first 100 frames are selected for testing, as the amount of this sequence is quite large, while all sequences of the other three datasets are adopted, during this experiment. For the INRIA dataset, parts of these images are resized, to guarantee that these pedestrians have similar size to the training dataset, as the pedestrian size scale of this dataset varied greatly.

The training data are the same for all these four testing dataset, which are selected from the NITCA pedestrian

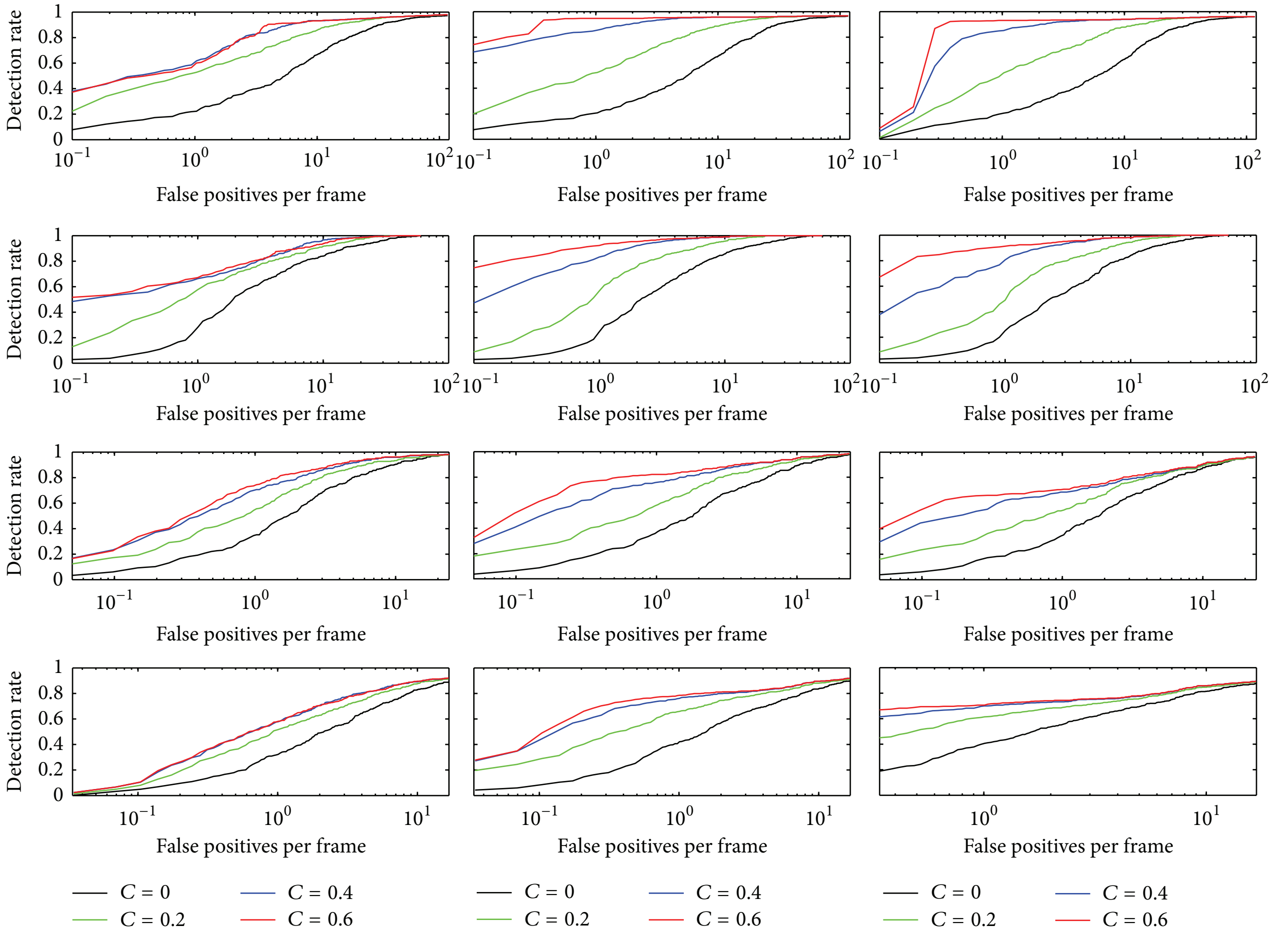


Figure 4: Comparison of parameter pair (λ, n). Each row of these subimages represents (detection rate/false positives per frame) curve for each dataset, which are "Multiple Flow," "City Center," USC(A), and INRIA dataset in order from top to bottom. The stage numbers n of these three columns are 8, 12, and 16 in order from left to right. Each subimage represents the performance of added confidence λ for given dataset with stage number n.

dataset [22], with image size of 32 × 80 pixels. However, in order to improve the performance of the proposed system, 600 nonpedestrian images from the Daimler dataset [23] are added to the negative training dataset of the cascaded AdaBoost algorithm. The number of positive training samples is 500 for the cascade AdaBoost algorithm while the number of negative samples is twice that of the positives. For RVFL net, the amount of positive and negative samples is the same and is set to be 3000.

Figure 3 shows the training and testing accuracies and times of RVFL net, with increasing number of hidden nodes from 10 to 200, by the step of 10. All these results are estimated by k-fold cross-validation [24]. During the cross-validation process, the whole dataset is randomly partitioned into 10 equal size subsets, and one single subset is selected as the validation data for testing the model, while the remaining 9 subsets are used as training data, on a case-by-case basis. Finally, the number of hidden nodes is set to be 180, with smooth and efficient training accuracy and high capability of generalization. When the number of hidden nodes is 180, the testing time is just 0.056 s, although the training time reaches 1.18 s. Therefore, the efficiency of RVFL net is guaranteed, during practical applications.

The minimum true positive and the maximum false alarm rates of CAB detector in our experiment are set to be 0.995 and 0.5, correspondingly. Figure 4 shows the (detection rate/false positives per frame) curve of the proposed system with the parameter pair (added confidence λ and stage number n of the CAB detector). The formulas of the pedestrian detection rate (PDR) and false positives per frame (FPPF) are demonstrated as follows:

$$\text{PDR} = \frac{TP}{TP + FN} \times 100\%,$$
$$\text{FPPF} = \frac{FP}{N_{\text{frame}}} \times 100\%, \tag{8}$$

where TP is the number of pedestrian samples correctly predicted to be pedestrians; FP is the number of nonpedestrian samples incorrectly predicted to be pedestrians; FN is the number of pedestrian samples incorrectly predicted to be nonpedestrians; N_{frame} is the number of total frames

Table 1: Pedestrian detection rate (%) and false positives per frame comparison of CAB-ELM and other state-of-the-art detectors.

Data sets		Propose (low)	Propose (high)	CAB	SVM	GAB	HF
Multiple Flow	PDR	94.55	94.72	94.39	94.22	61.55	78.99
	FPPF	0.56	0.75	0.30	1.68	1.70	1.66
City Center	PDR	90.49	95.29	93.10	95.15	66.04	80.22
	FPPF	0.46	1.80	0.71	0.960	1.43	1.50
USC	PDR	73.16	81.79	81.15	80.83	36.74	34.82
	FPPF	0.24	0.88	0.34	0.91	0.34	0.42
INRIA	PDR	76.28	89.25	71.84	89.08	62.29	34.82
	FPPF	0.59	9.24	3.54	9.55	0.89	0.42

(1) The number of cascade stages of CAB and CAB-RVFL is 12.

(2) (low) and (high) means two sets with (low/high) detection rates and corresponding false positives per frame.

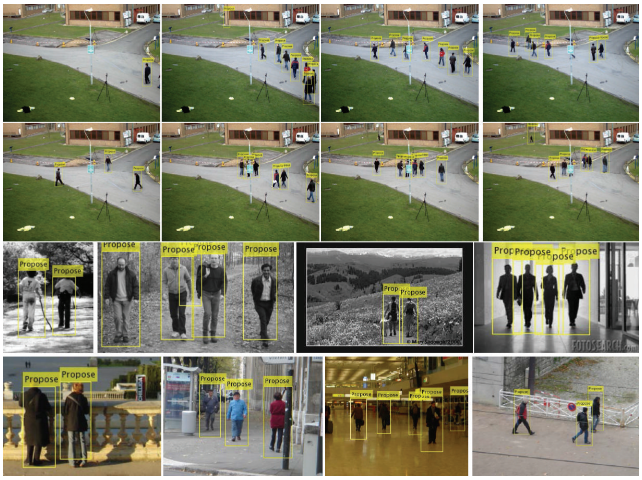


Figure 5: Examples of the four datasets. The first row is four samples of "Multiple Flow" dataset and the second row is "City Center" dataset. For demonstrated samples of these two datasets, each dataset has one pedestrian undetected due to the issue of heavy occlusion. The third row is USC(A) dataset and one false pedestrian is classified to be true positive target in the second subimage. The last row is INRIA dataset and all of the pedestrians are identified by the proposed system without any false pedestrians.

of the corresponding dataset sequences. We have tested the performance of added confidence λ with different values $\{0, 0.2, 0.4, 0.6, 0.8, 1\}$. The curve of $\lambda = 0.8, 1$ is very close to $\lambda = 0.6$, which means that the performance of the proposed system is beginning to stabilize when λ is growing larger than 0.6. The curve comparison of $\lambda = 0, 0.2, 0.4, 0.6$ is demonstrated in Figure 4. From all these 12 subfigures, the performance of $\lambda = 0.6$ is superior to $\lambda = 0, 0.2, 0.4$. Moreover, when the stage number is 12, the results are better than $n = 8, 16$, on the whole. Finally, the parameter pair is set to be (0.6, 12). Note that the single RVFL net can be regarded as a special case of the proposed system when $\lambda = 0$. Therefore, the proposed system has better performance than single RVFL net.

The comparison results of the proposed system and four other state-of-the-art detectors (CAB [9], SVM [7], GAB [25], and HF algorithm [8]) are shown in Table 1. In order to demonstrate the superiority of the proposed system, two sets of detection rates and the corresponding average number of false positives per frame of the proposed system are shown in the first two columns of Table 1. The second column shows high detection rates, at the cost of more false positives. However, its number of false positives per frame is still lower than SVM detector, in most cases. For "Multiple Flow" dataset, the low PDR of the proposed system is 94.55%, which is higher than those of the other four detectors. The corresponding FPPF, 0.56, is the lowest one among all these detectors. For "City Center" dataset, the low PDR and corresponding FPPF of the proposed system are 90.49% and 0.46, which are better than those for the GAB and HF algorithm. The high PDR reaches 95.29%, which is more accurate than CAB and SVM, at the cost of a few more false positives. For USC(A) dataset, the low PDR and corresponding FPPF are better than those for GAB and HF algorithm, and the high PDR and corresponding FPPF are better than those for SVM detector. The CAB algorithm has better performance of FPPF, while its detection rate is worse than the high PDR of the proposed system. For the INRIA dataset, the low PDR and corresponding FPPF of the proposed system are better than those for CAB, GAB, and HF algorithm, and the high PDR and corresponding FPPF are better than those for SVM detector.

Parts of the experimental results of the proposed system are depicted in Figure 5. During the detection results of these examples, the overwhelming majority of these pedestrians are detected with very few false pedestrians, through the validation of the proposed system.

4. Conclusion

In this paper, we presented a novel two-stage pedestrian detecting system based on a cascade AdaBoost detector and random vector functional-link net. The proposed system simultaneously enhances the detection accuracy and reduces the false positive rate, which improves the comprehensive performance for pedestrian detection. Numerous experiment comparisons with other state-of-the-art algorithms on four challenging datasets with different types demonstrate that the proposed system achieves favorable results, in terms of the detection rate and false positive rate simultaneously.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF), funded by the Ministry of Education (2013R1A1A2013778).

References

[1] D. Gerónimo, A. M. López, A. D. Sappa, and T. Graf, "Survey of pedestrian detection for advanced driver assistance systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 7, pp. 1239–1258, 2010.

[2] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: an evaluation of the state of the art," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 743–761, 2012.

[3] A. Shashua, Y. Gdalyahu, and G. Hayun, "Pedestrian detection for driving assistance systems: single-frame classification and system level performance," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 1–6, Parma, Italy, June 2004.

[4] M. Valera and S. A. Velastin, "Intelligent distributed surveillance systems: a review," *IEE Proceedings—Vision, Image and Signal Processing*, vol. 152, no. 2, pp. 192–204, 2005.

[5] H. Torresan, B. Turgeon, C. Ibarra-Castanedo, P. Hebert, and X. Maldague, "Advanced surveillance systems: combining video and thermal imagery for pedestrian detection," in *Thermosense XXVI*, vol. 5405 of *Proceedings of SPIE*, pp. 506–515, April 2004.

[6] M. Andriluka, P. Schnitzspan, J. Meyer et al., "Vision based victim detection from unmanned aerial vehicles," in *Proceedings of the 23rd IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '10)*, pp. 1740–1747, Taibei, Taiwan, October 2010.

[7] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, vol. 1, pp. 886–893, June 2005.

[8] O. Barinova, V. Lempitsky, and P. Kholi, "On detection of multiple object instances using hough transforms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1773–1784, 2012.

[9] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. I511–I518, December 2001.

[10] B. Leibe, A. Leonardis, and B. Schiele, "Robust object detection with interleaved categorization and segmentation," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 259–289, 2008.

[11] L. Guo, P.-S. Ge, M.-H. Zhang, L.-H. Li, and Y.-B. Zhao, "Pedestrian detection for intelligent transportation systems combining AdaBoost algorithm and support vector machine," *Expert Systems with Applications*, vol. 39, no. 4, pp. 4274–4286, 2012.

[12] Y.-H. Pao, G.-H. Park, and D. J. Sobajic, "Learning and generalization characteristics of the random vector Functional-link net," *Neurocomputing*, vol. 6, no. 2, pp. 163–180, 1994.

[13] G. H. Park and Y. H. Pao, "Unconstrained word-based approach for off-line script recognition using density-based random-vector functional-link net," *Neurocomputing*, vol. 31, no. 1-4, pp. 45–65, 2000.

[14] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face recognition with local binary patterns," *Proceedings of the 8th European Conference Computer Vision (ECCV '04)*, vol. 3021, pp. 469–481, 2004.

[15] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Elsevier Academic Press, New York, NY, USA, 5th edition, 2006.

[16] C.-C. Chen, "Improved moment invariants for shape discrimination," *Pattern Recognition*, vol. 26, no. 5, pp. 683–686, 1993.

[17] R. Fletcher and C. M. Reeves, "Function minimization by conjugate gradients," *The Computer Journal*, vol. 7, pp. 149–154, 1964.

[18] M. R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *Journal of Research of the National Bureau of Standards*, vol. 49, pp. 409–436, 1952.

[19] "PETS 2009: Eleventh IEEE International Workshop on Performance Evaluation of Tracking and Surveillance," 2009, http://pets2009.net/.

[20] B. Wu and R. Nevatia, "Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors," in *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV '05)*, vol. 1, pp. 90–97, Beijing, China, October 2005.

[21] INRIA dataset, http://lear.inrialpes.fr/data/.

[22] G. Overett, L. Petersson, N. Brewer, L. Andersson, and N. Pettersson, "A new pedestrian dataset for supervised learning," in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV '08)*, pp. 373–378, Eindhoven, The Netherlands, June 2008.

[23] C. G. Keller, M. Enzweiler, and D. M. Gavrila, "A new benchmark for stereo-based pedestrian detection," in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV '11)*, pp. 691–696, Baden, Germany, June 2011.

[24] G. J. McLachlan, K. A. Do, and C. Ambroise, *Analyzing Microarray Gene Expressiondata*, Wiley, New York, NY, USA, 2004.

[25] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting," *The Annals of Statistics*, vol. 28, no. 2, pp. 337–407, 2000.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 647380, 7 pages
http://dx.doi.org/10.1155/2014/647380

Research Article

A Vehicle Detection Algorithm Based on Deep Belief Network

Hai Wang,[1] Yingfeng Cai,[2] and Long Chen[2]

[1] *School of Automotive and Traffic Engineering, Jiangsu University, Zhenjiang 212013, China*
[2] *Automotive Engineering Research Institute, Jiangsu University, Zhenjiang 212013, China*

Correspondence should be addressed to Yingfeng Cai; caicaixiao0304@126.com

Received 25 March 2014; Accepted 22 April 2014; Published 15 May 2014

Academic Editor: Yu-Bo Yuan

Vision based vehicle detection is a critical technology that plays an important role in not only vehicle active safety but also road video surveillance application. Traditional shallow model based vehicle detection algorithm still cannot meet the requirement of accurate vehicle detection in these applications. In this work, a novel deep learning based vehicle detection algorithm with 2D deep belief network (2D-DBN) is proposed. In the algorithm, the proposed 2D-DBN architecture uses second-order planes instead of first-order vector as input and uses bilinear projection for retaining discriminative information so as to determine the size of the deep architecture which enhances the success rate of vehicle detection. On-road experimental results demonstrate that the algorithm performs better than state-of-the-art vehicle detection algorithm in testing data sets.

1. Introduction

Robust vision based vehicle detection on the road is to some extent a challenging problem since highways and urban and city roads are dynamic environment, in which the background and illuminations are dynamic and time variant. Besides, the shape, color, size, and appearance of vehicles are of high variability. To make this task even more difficult, the ego vehicle and target vehicles are generally in motion so that the size and location of target vehicles mapped to the image are diverse.

Although deep learning for object recognition has been an area of great interest in the machine-learning community, no prior research study has been reported that uses deep learning to establish an on-road vehicle detection method. In this paper, a 2D-DBN based vehicle detection algorithm is proposed.

The main novelty and contribution of this work include the following. (1) A deep learning architecture of 2D-DBN which preserves discriminative information for vehicle detection is proposed. (2) A deep learning based on-road vehicle detection system has been implemented and thorough quantitative performance analysis has been presented.

The rest of this paper is organized as follows. Section 2 will give a brief talking about vision based vehicle detection tasks and deep learning for object recognition. Section 3 introduces in detail the proposed 2D-DBN architecture and training methods for the vehicle detection tasks. The experiments and analysis will be given in Section 4 and Section 5 is the conclusion.

2. Related Research

In this section, a brief overview of two categories of work that is relevant to our research is introduced. The first set is about vision based vehicle detection and the second focuses on deep learning for object recognition.

2.1. Vision Based Vehicle Detection. Since only monocular visual perception is used in our project, this section will mainly refer to studies using monocular vision for on-road vehicle detection.

For monocular vision based vehicle detection, using vehicle appearance characteristics is the most common and effective approach. A variety of appearance features have been used in the field to detect vehicles. Some typical image features representing intuitive vehicle appearance information, such as local symmetry, edge, and cast shadow, have been used by many earlier works.

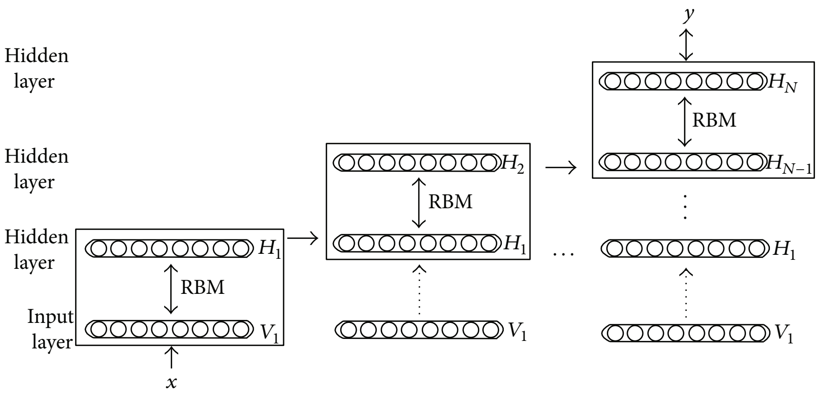


Figure 1: Architecture of deep belief network (DBN).

In recent years, there has been a transition from simpler image features to general and robust feature sets for vehicle detection. These feature sets, now common in the computer vision literature, allow for direct classification and detection of objects in images. For vehicle detection purpose, histogram of oriented gradient (HOG) features and Haar-like features are extremely well represented in literature. Besides, Gabor features, scale-invariant feature transform (SIFT) features, speeded up robust features (SURF), and some combined features are also applied for vehicle image representation.

Classification methods for appearance-based vehicle detection have also followed the general trends in the computer vision and machine learning literature. Compared to generative classifiers, discriminative classifiers, which learn a decision boundary between two classes (vehicles and unvehicles), have been more widely used in vehicle detection application. Support vector machines (SVM) and Adaboost are the most two common classifiers that are used for training vehicle detector. In [1], SVM classification was used to classify Haar feature vectors. The combination of HOG features and SVM classification has also been used [2–4]. Adaboost [5] has also been widely used for classification for Viola and Jones' contribution. The combination of Haar-like feature extraction and Adaboost classification has been used to detect rear faces of vehicles in [6–9]. Artificial neural network classifiers were also used for vehicle detection, but the training often failed due to local optimum [10].

2.2. Deep Learning for Object Recognition. Classifiers such as SVM and Adaboost referred to in last section are all indeed a shallow learning model because they both can be modeled as structure with one input layer, one hidden layer, and one output layer. Deep learning refers to a class of machine learning techniques, where hierarchical architectures are exploited for representation learning and pattern classification. Different from those shallow models, deep learning has the ability of learning multiple levels of representation and abstraction that helps to make sense of image data. From another point of view, deep learning can be viewed as one kind of multilayer neural networks with adding a novel unsurprised pretraining process.

There are various subclasses of deep architecture. Deep belief networks (DBN) modal is a typical deep learning structure which is first proposed by Hinton et al. [11]. The original DBN has demonstrated its success in simple image classification tasks of MNIST. In [12], a modified DBN is developed in which a Boltzmann machine is used on the top layer. This modified DBN is used in a 3D object recognition task.

Deep convolutional neural network (DCNN) with the ability to preserve the space structure and resistance to small variations in the images is used in image classification [13]. Recently, DCNN achieved the best performance compared to other state-of-the-art methods in the 2012 ImageNet LSVRC contest containing 1.2 million images with more than 1000 different classes. In this DCNN application, a very large architecture is built with more than 600,000 neurons and over 60 million weights.

DBN is a probabilistic model composed of multiple layers of stochastic, hidden variables. The learning procedure of DBN can be divided into two stages: generative learning to abstract information layer by layer with unlabelled samples firstly and then discriminative learning to fine-tune the whole deep network with labeled samples to the ultimate learning target [11]. Figure 1 shows a typical DBN with one input layer V_1 and N hidden layers $H_1, H_2, \ldots, H_N$, while x is the input data which can be, for example, a vector, and y is the learning target, for example, class labels. In the unsupervised stage of DBN training processes, each pair of layers grouped together to reconstruct the input of the layer from the output. In Figure 1, the layer-wise reconstruction happens between V_1 and H_1, H_1 and H_2, ..., H_{N-1} and H_n, respectively, which is implemented by a family of restricted Boltzmann machines (RBMs) [14]. After the greedy unsupervised learning of each pair of layers, the features are progressively combined from loose low-level representations into more compact high-level representations. In the supervised stage, the whole deep network is then refined using a contrastive version of the "wake-sleep" algorithm via a global gradient-based optimization strategy.

3. Deep Learning Based Vehicle Detection

In this section, a novel algorithm based on deep belief network (DBN) is proposed. Traditional DBN for object classification has some shortages. First, the training samples are regularized to first-order vector which will lead to the missing of spatial information contained by the image samples. This will obviously lead to a decline in the detection rate in vehicle detection tasks. Secondly, the size of layers (such as node number and layer number) in the traditional DBN is manually set which is often big and will lead to structural redundancy and increase the training and decision time of the classifier, while, for vehicle detection algorithm which is usually used in real-time application, decision time is a critical factor. The proposed 2D-DBN architecture for vehicle detection uses second-order planes instead of first-order vector of 1D-DBN as input and uses bilinear projection for retaining discriminative information so as to determine the size of the deep architecture. The bilinear projection maps original second-order output of lower layer to a small bilinear space without reducing discriminative information. And the size of the upper layer is that of the bilinear space.

In Section 3.1, the overall architecture of our 2D-DBN for vehicle detection will be introduced. In Section 3.2, the bilinear projection method of lower layer output will be given. In Sections 3.3 and 3.4, the training method of the whole 2D-DBN for vehicle detection will be deduced.

3.1. 2D Deep Belief Network (2D-DBN) for Vehicle Detection. Let X be the set of data samples including vehicle images and nonvehicle images, assuming that X is consisting with K samples which is shown below:

$$X = [\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_k, \ldots, \mathbf{X}_K]. \tag{1}$$

In X, $\mathbf{X}_k$ is training samples and in the image space $\mathbf{R}^{I\times J}$. Meanwhile, Y means the labels corresponding to X, which can be written as

$$Y = [y_1, y_2, \ldots, y_k, \ldots, y_K]. \tag{2}$$

In Y, y_k is the label vector of $\mathbf{X}_k$. If $\mathbf{X}_k$ belongs to vehicles, $y_k = (1, 0)$. On the contrary, $y_k = (0, 1)$.

The ultimate purpose in vehicle detection task is to learn a mapping function from training data X to the label data Y based on the given training set, so that this mapping function is able to classify unknown images between vehicle and nonvehicle.

Based on the task described above, a novel 2D deep belief network (2D-DBN) is proposed to address this problem. Figure 2 shows the overall architecture of 2D-DBN. A fully interconnected directed belief network includes one visible input layer V^1, N hidden layers $H^1, \ldots, H^N$, and one visible label layer La at the top. The visible input layer V^1 maintains $M \times N$ neural and equal to the dimension of training feature which is the original 2D image pixel values of training samples in this application. Since maximum discriminative ability wishes to be preserved from layer to layer with nonredundant layer size in this application for real-time requirement, the size of N hidden layers is dynamically decided with so-called bilinear projection. In the top, the La layer just has two units which is equal to the classes this application would like to classify. Till now, the problem is formulated to search for the optimum parameter space θ of this 2D-DBN.

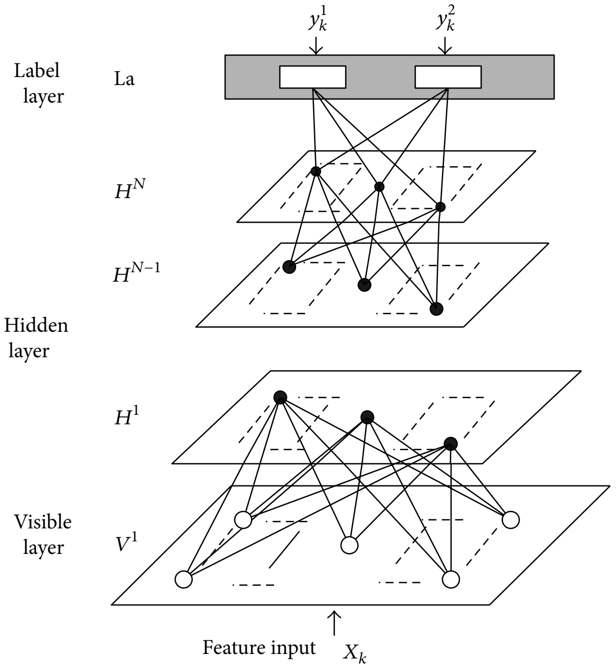


Figure 2: Proposed 2D-DBN for vehicle detection.

The main learning process of the proposed 2D-DBN has three steps.

(1) The bilinear projection is utilized to map the lower layer output data onto subspace and optimized to optimum dimension as well as to retain discriminative information. The size of the upper layer will be determined by this optimum dimension.

(2) When the size of the upper layer is determined, the parameters of the two adjacent layers will be refined with the greedy-wise reconstruction method. Repeat step (1) and step (2) till all the parameters of hidden layers are fixed. Here, step (1) and step (2) are so called pretraining process.

(3) Finally, the whole 2D-DBN will be fine-tuned with the La layer information based on back propagation. Here, step (3) can be viewed as supervised training step.

3.2. Bilinear Projection for Upper Layer Size Determination. In this section, followed with Zhong's contribution [15], bilinear projection is used in order to determine the size of every upper layer in adjacent layer groups. As mentioned in Section 3.1, with the labeled training data $\mathbf{X}_k \in \mathbf{R}^{I\times J}$ as the output of the visible layer V^1, bilinear projection maps the original data $\mathbf{X}_k$ onto a subspace and is represented by its latent form $\mathbf{LX}_k$. The bilinear projection is written as follows:

$$\mathbf{LX}_k = \mathbf{U}^T\mathbf{X}_k\mathbf{V}, \quad k = 1, 2, \ldots, K. \tag{3}$$

Here, $\mathbf{U} \in R^{M\times P}$ and $\mathbf{V} \in R^{N\times Q}$ are projection matrices that map the original data $\mathbf{X}_k$ by its latent form $\mathbf{LX}_k$ with the constraint that $\mathbf{U}^T\mathbf{U} = \mathbf{I}$ and $\mathbf{V}^T\mathbf{V} = \mathbf{I}$.

How to determine the value of $\mathbf{U}$ and $\mathbf{V}$ so that the discriminative information of $\mathbf{X}_k$ can be preserved is the issue that needs to be solved. For this, a specific objective function is built as follows:

$$\begin{aligned}\arg\max_{\mathbf{U},\mathbf{V}} J(\mathbf{U},\mathbf{V}) = \sum_{i,j}\left\|\mathbf{U}^T\left(\mathbf{X}_i - \mathbf{X}_j\right)\mathbf{V}\right\|^2 \\ \times\left(\alpha B_{ij} - (1-\alpha)W_{ij}\right),\end{aligned} \tag{4}$$

in which B_{ij} is between-class weights, W_{ij} is within class weights, and $\alpha \in [0, 1]$ is the balance parameter. B_{ij} and W_{ij} are calculated as follows [16]:

$$\begin{aligned} B_{ij} &= \begin{cases} -\dfrac{n_{nc}}{\left(n_{nc}+n_c\right)n_c}, & \text{if } y_i^c = y_j^c = 1 \\ \dfrac{1}{n_{nc}+n_c}, & \text{else,} \end{cases} \\ W_{ij} &= \begin{cases} \dfrac{1}{n_c}, & \text{if } y_i^c = y_j^c = 1 \\ 0, & \text{else.} \end{cases} \end{aligned} \tag{5}$$

Here, y_i^c is the class label of sample data $\mathbf{X}_i$, which is either $(1, 0)$ or $(0, 1)$. n_c is the number of samples that belong to class c and n_c is those not belonging to class c. Since vehicle detection is a binary classification problem, $c \in \{1, 2\}$ in this application.

It can be seen that the purpose of the objective function is to simultaneously maximize the between-class distances and minimize the within-class distances. In other words, the objective function focuses on maximizing the discriminative information of all the sample data. However, optimizing $J(\mathbf{U}, \mathbf{V})$ is a nonconvex optimization problem with two matrices $\mathbf{U}$ and $\mathbf{V}$. To deal with this trouble, a strategy called alternative fixing (AF) is used, which is fixing $\mathbf{U}$ (or $\mathbf{V}$) and optimizing the objective function $J(\mathbf{U}, \mathbf{V})$ with just variable matrix $\mathbf{V}$ (or $\mathbf{U}$) and then fixing $\mathbf{V}$ (or $\mathbf{U}$) and optimizing $J(\mathbf{U}, \mathbf{V})$ with just $\mathbf{U}$ (or $\mathbf{V}$). AF will be implemented alternatively till $J(\mathbf{U}, \mathbf{V})$ reaches its upper bound.

After the optimum process, new $\mathbf{U}^*$ and $\mathbf{V}^*$ that maximize $J(\mathbf{U}, \mathbf{V})$ are got and preserve the discriminative information of original sample data X. Based on this, then, the size of the upper layer can be determined by the number of positive eigenvalues of $\mathbf{U}^*$ and $\mathbf{V}^*$, which is P and Q, respectively.

3.3. Pretraining with Greedy Layer-Wise Reconstruction Method. In last subsection, the size of the upper layer is determined to be $P \times Q$. In this subsection, the parameters of the two adjacent layers will be refined with the greedy-wise reconstruction method proposed by Hinton et al. [11]. To illustrate this pretraining process, we take the visible input layer V^1 and the first hidden layer H^1 for example.

The visible input layer V^1 and the first hidden layer H^1 contract a restrict Boltzmann machine (RBM). $I \times J$ is the neural number in V^1 and $P \times Q$ is that of H^1. The energy of the state (v^1, h^1) in this RBM is

$$\begin{aligned} E\left(\mathbf{v}^1, \mathbf{h}^1, \theta^1\right) &= -\left(\mathbf{v}^1\mathbf{A}\mathbf{h}^1 + \mathbf{b}^1\mathbf{v}^1 + \mathbf{c}^1\mathbf{h}^1\right) \\ &= -\sum_{i=1,j=1}^{i\le I,j\le J}\sum_{p=1,q=1}^{p\le P,q\le Q} v_{ij}^1 A_{ij,pq}^1 h_{pq}^1 \\ &\quad - \sum_{i=1,j=1}^{i\le I,j\le J} b_{ij}^1 v_{ij}^1 - \sum_{p=1,q=1}^{p\le P,q\le Q} c_{pq}^1 h_{pq}^1, \end{aligned} \tag{6}$$

in which $\theta^1 = (\mathbf{A}^1, \mathbf{b}^1, \mathbf{c}^1)$ are the parameters between the visible input layer V^1 and the first hidden layer H^1. $A_{ij,pq}^1$ is the symmetric weights from input neural (i, j) in V^1 to the hidden neural (p, q) in H^1. b_{ij}^1 and c_{pq}^1 are the $(i, j)^{th}$ and $(p, q)^{th}$ bias of V^1 and H^1. So this RBM is with the joint distribution as follows:

$$\begin{aligned} P\left(\mathbf{v}^1, \mathbf{h}^1; \theta^1\right) &= \frac{1}{Z} e^{-E(\mathbf{v}^1,\mathbf{h}^1;\theta^1)} \\ &= \frac{e^{-E(\mathbf{v}^1,\mathbf{h}^1;\theta^1)}}{\sum_{v^1}\sum_{h^1} e^{-E(\mathbf{v}^1,\mathbf{h}^1;\theta^1)}}. \end{aligned} \tag{7}$$

Here, Z is the normalization parameter and the probability that $\mathbf{v}^1$ is assigned to V^1 of this modal is

$$P\left(\mathbf{v}^1\right) = \frac{1}{Z}\sum_{h^1} e^{-E(\mathbf{v}^1,\mathbf{h}^1;\theta^1)} = \frac{\sum_{h^1} e^{-E(\mathbf{v}^1,\mathbf{h}^1;\theta^1)}}{\sum_{v^1}\sum_{h^1} e^{-E(\mathbf{v}^1,\mathbf{h}^1;\theta^1)}}. \tag{8}$$

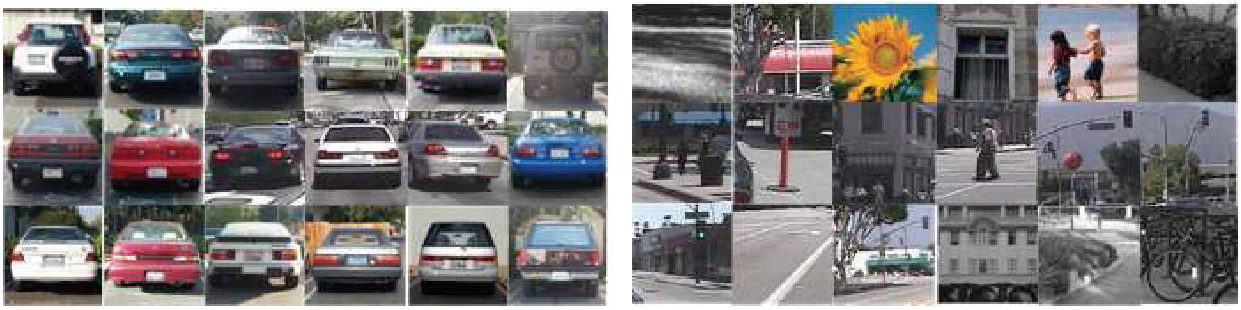
After that, the conditional distributions over visible input state $\mathbf{v}^1$ in layer V^1 and hidden state h^1 in H^1 are able to be given by the logistic function, respectively,

$$\begin{aligned} p\left(\mathbf{h}^1 \mid \mathbf{v}^1\right) &= \prod_{p,q} p\left(h_{pq}^1 \mid \mathbf{v}^1\right), p\left(h_{pq}^1 \mid \mathbf{v}^1\right) \\ &= \sigma\left(\sum_{i=1,j=1}^{i\le I,j\le J} v_{ij}^1 A_{ij,pq}^1 + c_{pq}^1\right), \end{aligned} \tag{9}$$

$$\begin{aligned} p\left(\mathbf{v}^1 \mid \mathbf{h}^1\right) &= \prod_{i,j} p\left(\mathbf{v}_{ij}^1 \mid \mathbf{h}^1\right), p\left(\mathbf{v}_{ij}^1 \mid \mathbf{h}^1\right) \\ &= \sigma\left(\sum_{p=1,q=1}^{p\le P,q\le Q} h_{pq}^1 A_{ij,pq}^1 + b_{ij}^1\right). \end{aligned} \tag{10}$$

Here, $\sigma(x) = 1/(1 + \exp(-x))$.

At last, the weights and biases are able to be updated step by step from random Gaussian distribution values



(a)

(b)

Figure 3: Some positive and negative training samples. (a) Positive samples. (b) Negative samples.

$A^1_{ij,pq}(0)$, $b^1_{ij}(0)$, and $c^1_{pq}(0)$ with Contrastive Divergence algorithm [17], and the updating formulations are

$$\begin{aligned} A^1_{ij,pq} &= \vartheta A^1_{ij,pq} \\ &\quad + \varepsilon_A \left(\left\langle v^1_{ij}(0)\, h^1_{ij}(0) \right\rangle_{\text{data}} - \left\langle v^1_{ij}(t)\, h^1_{ij}(t) \right\rangle_{\text{recon}} \right), \\ b^1_{ij} &= \vartheta b^1_{ij} + \varepsilon_b \left(v^1_{ij}(0) - v^1_{ij}(t) \right), \\ c^1_{pq} &= \vartheta c^1_{pq} + \varepsilon_c \left(h^1_{pq}(0) - h^1_{pq}(t) \right), \end{aligned} \tag{11}$$

in which $\langle\cdot\rangle_{\text{data}}$ means the expectation with respect to the data distribution and $\langle\cdot\rangle_{\text{recon}}$ means the reconstruction distribution after one step. Meanwhile, t is step size which is set to $t = 1$ typically.

Above, the pretraining process is demonstrated by taking the visible input layer V^1 and the first hidden layer H^1 for example. Indeed, the whole pretraining process will be taken from low layer groups $(V^1,\ H^1)$ to up layer groups $(H^{n-1},\ H^n)$ one by one.

3.4. Global Fine-Tuning. In the above unsurprised pretraining process, the greedy layer-wise algorithm is used to learn the 2D-DBN parameters with the information added from bilinear projection. In this subsection, a traditional back propagation algorithm will be used to fine-tune the parameters $\theta = [\mathbf{A}, \mathbf{b}, \mathbf{c}]$ with the information of label layer La.

Since good parameters initiation has been maintained in the pretraining process, back propagation is just utilized to finely adjust the parameters so that local optimum parameters $\theta^* = [\mathbf{A}^*, \mathbf{b}^*, \mathbf{c}^*]$ can be got. In this stage, the learning objection is to minimize the classification error $[-\sum_t \mathbf{y}_t \log \widehat{\mathbf{y}}_t]$, where $\mathbf{y}_t$ and $\widehat{\mathbf{y}}_t$ are the real label and output label of data $\mathbf{X}_t$ in layer N.

4. Experiment and Analysis

This section will take experiments on vehicle datasets to demonstrate the performance of the proposed 2D-DBN. The datasets for training are Caltech1999 database which includes images containing 126 rear view vehicles. Besides, another 600 vehicles in images are collected by our groups in recorded road videos for training. Meanwhile, the negative samples are chosen from 500 images not containing vehicles and the number of negative samples for training is 5000. Figure 3 shows some of these positive and negative training samples. The testing datasets are recorded road videos with 735 manual marked vehicles.

Table 1: Detection results of three different architectures of 2D-DBN.

Classifier types	Correct labeling	Correct rate
2D-DBN (1H)	689/735	93.74%
2D-DBN (2H)	**706/735**	**96.05%**
2D-DBN (3H)	695/735	94.56%

By using the proposed method, three different architectures of 2D-DBN are applied. They all contain one visible layer and one label layer, but with one, two, and three hidden layers, respectively. In training, the critical parameters of the proposed 2D-DBN in experiments are set as $\alpha = 0.5$ and $\vartheta = 0.8$ and image samples for training are all resized to 32×32.

The detection results of these three architectures of 2D-DBN are shown in Table 1. It can be seen that 2D-DBN with two hidden layers maintains the highest detection rate.

The learned weights of hidden layers are shown in Figure 4.

Then, we compared the performance of our 2D-DBN with many other state-of-the-art classifiers, including support vector machine (SVM), k-nearest neighbor (KNN), neural networks, 1D-DBN, and deep convoluted neural network (DCNN).

The detection results of these methods are shown in Table 2.

From the compared results, it can be concluded that classification methods with deep architecture, for example, 1D-DBN, DCNN, and 2D-DBN are significantly better than those of shallow architecture, for example, SVM, KNN, and NN. Moreover, our proposed 2D-DBN is better than 1D-DBN and DCNN due to 2D feature input and the bilinear projection.

Finally, this 2D-DBN vehicle detection method is utilized on road vehicle detection system and some of the vehicle

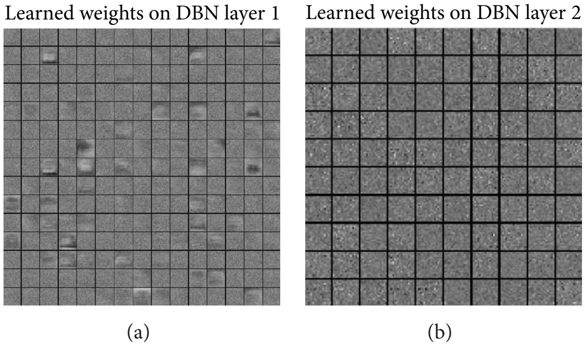

Learned weights on DBN layer 1
Learned weights on DBN layer 2
(a)
(b)

Figure 4: Learned weights of first hidden layer and second hidden layer on 2D-DBN: (a) weights of first hidden layer and (b) weights of second hidden layer.

Table 2: Detection results of multiple methods.

Classifier types	Correct labeling	Correct rate
SVM	658/735	89.52%
KNN	642/735	87.35%
NN	619/735	84.21%
1D-DBN	684/735	93.06%
DCNN	697/735	94.83%
2D-DBN (2H)	**706/735**	**96.05%**

sensing results in real road situation are shown in Figure 5. The four rows of images are picked in daylight highway, raining day highway, daylight urban, and night highway with road lamp, respectively. The solid green box means detected vehicles, and the dotted red box means undetected vehicles or false detected vehicles. The average vehicle detection time for one frame with 640 × 480 resolution is around 53 ms in our Advantech industrial computer.

Overall, most of the on-road vehicles can be sensed successfully while misdetection and false detection sometimes occurred during adverse situations such as partial occlusion and bad weather.

5. Conclusion

In this work, a novel vehicle detection algorithm based on 2D-DBN is proposed. In the algorithm, the proposed 2D-DBN architecture uses second-order planes instead of first-order vector as input and uses bilinear projection for retaining discriminative information so as to determine the size of the deep architecture which enhances the success rate of vehicle detection. On-road experimental results demonstrate that the system works well in different roads, weather, and lighting conditions.

The future work of our research will focus on the situation when a vehicle is partially occluded with deep architecture framework.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

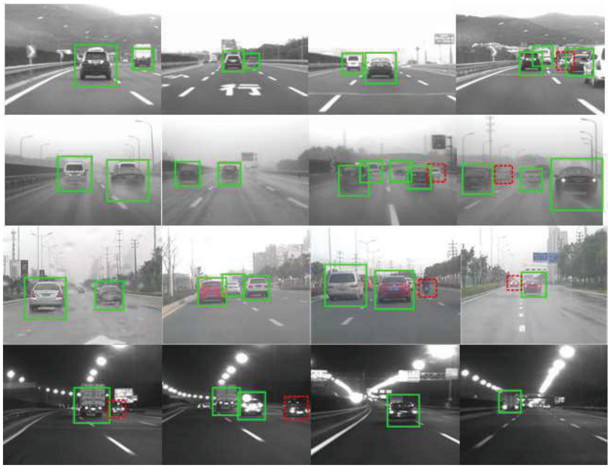


Figure 5: Some of the real road vehicle sensing results. First row: daylight highway situation; second row: raining day highway situation; third row: daylight urban situation; forth row: night highway with road lamp.

Acknowledgments

This research was funded partly and was supported in part by the National Natural Science Foundation of China under Grant no. 51305167, Information Technology Research Program of Transport Ministry of China under Grant no. 2013364836900, and Jiangsu University Scientific Research Foundation for Senior Professionals under Grant no. 12JDG010.

References

[1] W. Liu, X. Wen, B. Duan, H. Yuan, and N. Wang, “Rear vehicle detection and tracking for lane change assist,” in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV ’07)*, pp. 252–257, June 2007.

[2] R. Miller, Z. Sun, and G. Bebis, “Monocular precrash vehicle detection: features and classifiers,” *IEEE Transactions on Image Processing*, vol. 15, no. 7, pp. 2019–2034, 2006.

[3] S. Sivaraman and M. M. Trivedi, "Active learning for on-road vehicle detection: a comparative study," *Machine Vision and Applications*, pp. 1–13, 2011.

[4] S. Teoh and T. Brunl, "Symmetry-based monocular vehicle detection system," *Machine Vision and Applications*, vol. 23, pp. 831–842, 2012.

[5] R. Sindoori, K. Ravichandran, and B. Santhi, "Adaboost technique for vehicle detection in aerial surveillance," *International Journal of Engineering & Technology*, vol. 5, no. 2, 2013.

[6] J. Cui, F. Liu, Z. Li, and Z. Jia, "Vehicle localisation using a single camera," in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV '10)*, pp. 871–876, June 2010.

[7] T. T. Son and S. Mita, "Car detection using multi-feature selection for varying poses," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 507–512, June 2009.

[8] D. Acunzo, Y. Zhu, B. Xie, and G. Baratoff, "Context-adaptive approach for vehicle detection under varying lighting conditions," in *Proceedings of the 10th International IEEE Conference on Intelligent Transportation Systems (ITSC '07)*, pp. 654–660, October 2007.

[9] C. T. Lin, S. C. Hsu, J. F. Lee et al., "Boosted vehicle detection using local and global features," *Journal of Signal & Information Processing*, vol. 4, no. 3, 2013.

[10] O. Ludwig Jr. and U. Nunes, "Improving the generalization properties of neural networks: an application to vehicle detection," in *Proceedings of the 11th International IEEE Conference on Intelligent Transportation Systems (ITSC '08)*, pp. 310–315, December 2008.

[11] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[12] V. Nair and G. E. Hinton, "3D object recognition with deep belief nets," in *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems (NIPS '09)*, pp. 1339–1347, December 2009.

[13] G. Taylor, R. Fergus, Y. L. Cun, and C. Bregler, "Convolutional learning of spatio-temporal features," in *Computer Vision—ECCV 2010*, vol. 6316 of *Lecture Notes in Computer Science*, pp. 140–153, 2010.

[14] C. X. Zhang, J. S. Zhang, N. N. Ji et al., "Learning ensemble classifiers via restricted Boltzmann machines," *Pattern Recognition Letters*, vol. 36, pp. 161–170, 2014.

[15] S.-H. Zhong, Y. Liu, and Y. Liu, "Bilinear deep learning for image classification," in *Proceedings of the 19th ACM International Conference on Multimedia ACM Multimedia (SIG MM '11)*, December 2011.

[16] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: a general framework for dimensionality reduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 40–51, 2007.

[17] F. Wood and G. E. Hinton, "Training products of experts by minimizing contrastive divergence," Tech. Rep. 2012. 53, 143, Brown University, 2012.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 650147, 8 pages
http://dx.doi.org/10.1155/2014/650147

Research Article

Improving Causality Induction with Category Learning

Yi Guo,[1,2] Zhihong Wang,[1] and Zhiqing Shao[1]

[1] *Department of Computer Science and Engineering, East China University of Science and Technology, P.O. Box 408, Shanghai 200237, China*
[2] *School of Information Science and Technology, Shihezi University, Shihezi 832003, China*

Correspondence should be addressed to Yi Guo; yguo71625@foxmail.com

Received 7 April 2014; Accepted 14 April 2014; Published 30 April 2014

Academic Editor: Shan Zhao

Causal relations are of fundamental importance for human perception and reasoning. According to the nature of causality, causality has explicit and implicit forms. In the case of explicit form, causal-effect relations exist at either clausal or discourse levels. The implicit causal-effect relations heavily rely on empirical analysis and evidence accumulation. This paper proposes a comprehensive causality extraction system (CL-CIS) integrated with the means of category-learning. CL-CIS considers cause-effect relations in both explicit and implicit forms and especially practices the relation between category and causality in computation. In elaborately designed experiments, CL-CIS is evaluated together with general causality analysis system (GCAS) and general causality analysis system with learning (GCAS-L), and it testified to its own capability and performance in construction of cause-effect relations. This paper confirms the expectation that the precision and coverage of causality induction can be remarkably improved by means of causal and category learning.

1. Introduction

A general philosophical definition of causality states that, the philosophical concept of causality refers to the set of all particular causal or cause-and-effect relations [1]. Causal relations are of fundamental importance for human perception and reasoning. Since ignoring causal relationships may have fatal consequences, their knowledge plays a crucial role in daily life to ensure survival in an ever changing environment.

In many research works, the causality generally refers to the existence of causality in mathematics and physics. Causalities are often investigated in situations influenced by uncertainty involving several variables. Thus, causalities, which can be presented in terms of flows among processes or events, are generally expressed in mathematical languages and analyzed in a mathematical manner. Therefore, statistics and probability theories seem to be the most popular mathematical languages for modeling causality in most scientific disciplines. There is an extensive range of literature on causality modeling, applying and combining mathematical logic, graph theory, Markov models, Bayesian probability, and so forth [2]. But, they seem not to be able to predominate all relevant issues or questions.

In recent years, clarification and extraction of cause-effect relationships among texts (e.g., objects or events), causality extraction, is elevated to a prominent research topic in text mining, knowledge engineering, and knowledge management.

A variety of research works testify that causalities in texts can be extracted at three technology levels. The first is the clausal level (CL), which includes cue phrases and lexical clues [3] and semantic similarity [4]. The second is the discourse level (DL), which implements connective markers [5] and constructs discourse relations [6]. The third is the mode level (ML), which extracts causal relations from a QA system [7, 8], applies commonsense rules [9], associative memory [10], and Chain Event Graphs [11].

According to the nature of causality in texts, causality has explicit and implicit forms. In the case of explicit form, causal-effect relations exist at either clausal or discourse levels. The implicit causal-effect relations heavily rely on empirical analysis and evidence accumulation.

Moreover, Waldmann and Hagmayer [12], in their research work of cognitive psychology, state that "categories that have been acquired in previous learning contexts may influence subsequent causal learning," which indicate that

(1) the category information about objects or events in texts is a necessary supplement for causality extraction, and (2) the category information has impact on subsequent learning-based cause-effect identification.

Based on the facts, the task of causality extraction, even induction, in texts could not be accomplished in an arbitrary manner. This paper proposes a comprehensive causality extraction system (CL-CIS) integrated with the means of category-learning. CL-CIS considers cause-effect relations in both explicit and implicit forms and especially practices the relation between category and causality in computation.

The rest of this paper is organized as follows. Section 2 states the causality, category information, and the relationships between them in texts and constructs the theoretical foundation of our research work. Section 3 expatiates the methodology and the technical details of system structure and kernel algorithms. Section 4 focuses on experiment illustration and result analysis of the experimental results. Section 5 concludes this paper and provides future research works.

2. Causality and Category

Traditionally, research about the representation of causal relations and research about the representation of categories were separated. This research strategy rests on the assumption that categories summarize objects or events on the basis of their similarity structure, whereas causality refers to relations between causal objects or events. Literature [12] proves that the relationship between causality and categorization is more dynamic than previously thought.

2.1. Causality. The standard view guiding research on causality presupposes the existence of objective networks of causes and effects that cognitive systems try to mirror. Regardless of whether causal learning is viewed as the attempt to induce causality on the basis of statistical information or on the basis of mechanism information, it is generally assumed that the goal of causal learning is to form adequate representations of the texture of the causal world.

2.2. Causality Rests on Fixed Categories. Studies on causal learning typically investigate trial-by-trial learning tasks which involve learning the contingencies between causes and effects. In a large number of studies which focus on causal contingency learning. A characteristic feature of these tasks is that they present categorized events representing causes and effects which are statistically related. Cause and effect categories are viewed as fixed entities that are already present prior to the learning task. The goal of learning is to estimate causal strength of individual causal links or to induce causal models on the basis of observed covariations. The role of cause and effect categories in the learning process is not the focus of interest in these approaches; they are simply viewed as given.

A similar approach underlies research on the relationship between categories and causality. According to the view that categorization is theory-based, traditional similarity-based accounts of categorization are deficient because they ignore the fact that many categories are grounded in knowledge about causal structures [13]. In natural concepts, features often represent causes or effects with the category label referring to a complex causal model. For example, disease categories frequently refer to common-cause models of diseases with the category features representing causes (e.g., virus) and effects (e.g., symptoms) within this causal model. A number of studies using these and similar materials have shown that the type of causal model connecting otherwise identical cause and effect features influences learning, typicality judgments, or generalization [14–17]. The main goal of these studies was to investigate the effect of different causal relations connecting the causal features. As in contingency learning studies, the cause and effect features within the causal models were treated as fixed, categorized entities, which already existed prior to the learning context.

2.3. Categories Shape Causality. It is certainly true that many interesting insights can be gained from investigating how people learn about causal models on the basis of preexisting cause and effect categories. However, there is also a link between categories and causality in the opposite direction. The categories that have been acquired in previous learning contexts may have a crucial influence on subsequent causal learning.

The basis of the potential influence of categories on causal induction lies in the fact that the acquisition and use of causal knowledge is based on categorized events. Causal laws, such as the fact that smoking causes heart disease, can only be noticed on the basis of events that are categorized (e.g., events of smoking and cases of heart disease).

Without such categories causal laws neither could be detected nor could causal knowledge be applied to new cases. Thus, causal knowledge not only affects the creation of categories, it also presupposes already existing categories for the description of causes and effects. The potential influence of categories is due to the fact that one of the most important cues to causality is statistical covariation between causes and effects.

To study the relation between categories and causal induction, [12] have developed a new paradigm that consists of three phases, the category learning phase, the causal learning phase, and the third test phase. The main goal is to answer the question that under what condition learners will tend to activate the categorical information (Figure 1) from the earlier category learning phase when learning about causal contingencies in the later phase. This effect is entailed by the fact that the alternative categories form different reference classes.

In category learning phase, causes will be classified into the distinct categories (upper left arrow in Figure 1). In causal learning phase, causes in collection are paired with the presence or absence of an effect (lower arrow). In the subsequent test phase, the test causes are rated with the likelihood of the effect. The crucial question is when people would go through the upper route in Figure 1 and assign the test causes to the categories in the category learning phase or

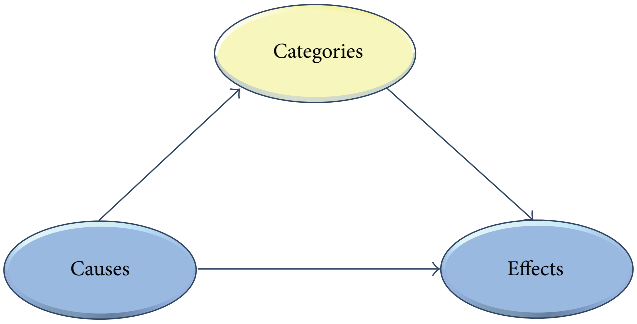


Figure 1: Possible routes of category learning between causes and effects.

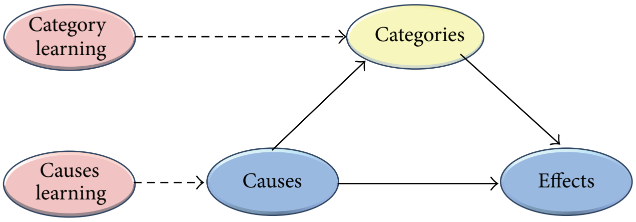


Figure 2: Enhanced strategy of category learning between causes and effects.

whether they would stick to the lower route and induce new categories within causal learning phase.

If the learning strategy opted for the lower route, one possible solution may be to induce new categories that are maximally predictive of the effects. This solution would obviously generate maximally predictive categories. Lien and Cheng [18] reported a research that is consistent with the maximal-contrast hypothesis. Their experiments show that the substances are categorized according to the feature and to the hierarchical level that were maximally predictive for the effect. Thus, the induced substance category was determined by its suitability for predicting the effect. Lien and Cheng [18] interpreted this as evidence for their maximal-contrast hypothesis. In another word, people tend to induce categories that maximize their causal predictability.

In sum, our research addresses the question of which route learners will go. Will they routinely go through the upper road and activate category knowledge when learning about novel effects? A simple connectionist one-layer network that may be used to understand the task we are going to explore in our experiments, or will they go the lower road and learn a new set of categories on the basis of causal information in causal learning phase that is maximally predictive within this learning phase? Thus, Figure 1 is enhanced with two learning phases as shown in Figure 2.

3. Research Methodology and System Structure

3.1. Research Methodology. Causality is a fundamental concept in human thinking and reasoning. It is not surprising that most, if not all, languages in the world have a range of lexical expressions specifically designed for communicating causal relations.

Table 1: Statistics for backward causality connectives.

Connectives	For objective reason	For subjective reason	Total percentage in all connectives
Because	83%	17%	50%
For	29%	71%	18%
As	60%	40%	13%
Since	14%	86%	6%
While	14%	86%	6%
Misc	0%	100%	7%
Total	56%	44%	100%

This paper focuses on explicit causality markers and implicit causal relations in text. The explicit causality markers include two grammatically different types of causality markers in English. We investigate the semantic contrasts expressed by different causal auxiliary verbs, marking causal relations expressed *within* one clause, and those expressed by different causal connectives, marking causal relations *between* clauses.

3.1.1. Causality in Verbs. Some instances of causality in verbs are listed below. These verbs include, but are not limited to: "make," "let," "have," "cause," and their synonyms from WordNet [19] which is generally referred to as an online lexical database.

> *[The extreme cold] cause made/caused even [the rivers (to) freeze] effect.*
>
> *[She] cause made/had [her son empty his plate] effect, despite his complaints.*

3.1.2. Causality in Discourse Connectives. For a complex sentence, the predicative or relative clauses of the noun synonyms of "cause" are labeled as "potential cause" of antecedent sentence or main clause. Additionally, each clause inducted with "since," "as," or "because" is also labeled as "potential cause" of its main clause, which is labeled as "potential effect."

Table 1 lists the statistics for above backward causality collected upon Reuters-21578, currently the most widely used test collection for text processing research, and BBC-News2000, collected by a self-developed Web Crawler from http://www.bbc.co.uk/. The forward causality, where in presentation order the cause precedes the effect, is the most frequently used ones, for example, "therefore," "because of that," "that is why," and "so." As the forward causality connectives are one hundred percent strong causality indicators, the backward causality connectives in Table 1 need more specific notation. For example, in all "because" connectives in collections, 83% of them indicate objective reasons, while the other 17% indicate subjective reasons. Meanwhile, the "because" connectives persist 50% of all the connectives including "because," "for," "as," "since," "while," and other miscellaneous connectives. The "for," "as," "since," "while," and

Table 2: Types of implicit causality sentences.

Types	Subtypes	Exemplar sentences
Compound sentences	Cause-effect sentences connected with "and"	Cause-effect: (1) This was the first time I made an international call, and my heart was beating fast. (2) The filter is much more efficient than the primary filter and it removes all remaining solid particles from the fuel. Effect-cause: (3) The crops had failed, and there had not been any rain for months. (4) Aluminum is used as engineering material for planes and spaceships and it is light and tough.
	Cause-effect sentences without connectives	Effect-cause: (5) My heart sank. Some of them did not weigh more than 135 pounds, and the others looked too young to be in trouble. (6) . . .but the Voyager's captain and three crewmen had stayed in board. They had hoped the storm would die out and they could save the ship. Cause-effect: (7) The red distress flare shot up from the sinking ship, the Voyager. Everyman aboard our Coast Guard cutter knew that time had run out.
Relative clauses	SV, SVO, SVC, SVOC, SVOO, SVA, SVOA	(8) To make an atom we have to use uranium, in which the atoms are available for fission. (9) We know that a cat, whose eyes can take in more rays of light than our eyes, can see clearly in the night.
If clauses		(10) This system of subsidies must be maintained if the farmer will suffer considerable losses if it is abolished. (11) If the water will rise above this level, we must warn everybody in the neighborhood.
That clauses		(12) The act was even the bolder that he stood utterly alone. (13) The crying was all the more racking that she was not hysterical but hopeless.
SVO-SVOC		(14) Her falling ill spoiled everything. (15) Timidity and shamefacedness caused her to stand back and looked indifferently away.

other miscellaneous connectives correspondingly hold 18%, 13%, 6%, 6%, and 7% of all the connectives in collections.

3.1.3. Implicit Causality in Texts. Causal effect relations are general connections in the objective world, and the causality sentences exist in languages in a pervasive manner. The explicit causality sentences are those conducted by backward and forward causality connectives listed in Section 3.1.2, while the implicit causality sentences are those connected with other connectives, even without any one. Our research works have testified that there are five types of implicit causality sentences shown in Table 2.

In the practical English texts, there exist a few interpersonal verbs like "praise" and "apologize," thus supplying information about whose behavior or state is the more likely immediate cause of the event at hand. Because it is conveyed implicitly as part of the meaning of the verb, this probabilistic cue is usually referred to as implicit causality.

Such exemplar implicit causality verbs [20] adopted in this paper are listed in Table 3 together with their bias indicating the probabilities as causal cues; for example, 1.00 is the causal baseline; the higher the bias value is, the more likely is the cause.

3.2. System Description. Our causality induction system with assistance of category learning, named CL-CIS, is composed with the following functional modules (shown in Figure 3): category learning, classify exemplars into categories, category and causal mapping, causal learning, building causal-effect relations, and the final testing module. CL-CIS also includes three reference libraries (databases): causality in verbs, causality in discourse connectives, and implicit causality for query and assistance. Table 4 compares the composition difference among general causality analysis system (GCAS), general causality analysis system with learning (GCAS-L), and CL-CIS. As the construction of three libraries is elaborated in Section 3.1 Methodology, this section concentrates on the six functional modules.

3.2.1. Category Learning (CL). The category learning module builds up different distinct and exhaustive categories according to semantic contents (e.g., noun phrases, verb phrases) of sentences in our text collections. This module is a basic preprocessing step and the target is to construct a collection of distinct and exhaustive categories with the text learning methods.

Table 3: Exemplar implicit causality verbs.

NP1 verbs	Bias	NP2 verbs	Bias
Amazed	1.19	*Admire*	2.00
Annoy	1.19	*Adore*	1.86
Apologize	1.00	*Appreciate*	2.00
Be in the way	1.08	*Comfort*	1.91
Beg	1.17	*Compliment on something*	1.96
Bore	1.05	*Congratulate*	1.95
Call	1.19	*Criticize*	2.00
Confess	1.03	*Envy*	2.00
Disappoint	1.03	*Fear*	2.00
Disturb	1.14	*Fire*	1.96
Fascinate	1.00	*Hate*	1.96
Hurt	1.13	*Hold in contempt*	2.00
Inspire	1.23	*Hold responsible*	1.91
Intimidate	1.23	*Loathe*	1.86
Irritate	1.22	*Love*	1.86
Lie to	1.22	*Praise*	1.96
Mislead	1.22	*Press charges against*	1.91
Swindle	1.13	*Punish*	1.95
Win	1.19	*Respect*	1.95
Worry	1.19	*Thank*	1.82

Table 4: System composition comparison of GCAS, GCAS-L, and CL-CIS.

System composition	GCAS	GCAS-L	CL-CIS
Modules			
Category learning			√
Classify exemplars into categories			√
Category and causal mapping			√
Causal learning		√	√
Building causal-effect relations	√	√	√
Testing causal-effect relations	√	√	√
Libraries			
Causality in verbs	√	√	√
Causality in discourse connectives	*Optional*	√	√
Implicit causality		*Optional*	√

3.2.2. Classify Exemplars into Categories (CEC). In the classify exemplars into categories module, each sentence is treated as an individual independent event and parsed into phrases. The classification is based on the comparison of a sentence with features of each category, so as to set up the category background knowledge for each sentence. For example, the concept *bank* can be clarified as a finance organization or a river body boundary with corresponding category background knowledge of its sentence and contexts.

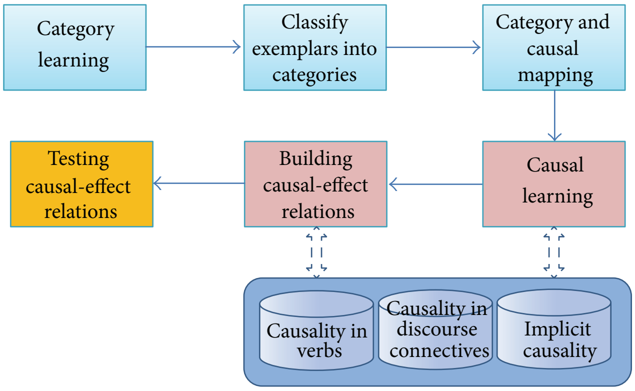

Category learning
Classify exemplars into categories
Category and causal mapping
Testing causal-effect relations
Building causal-effect relations
Causal learning
Causality in verbs
Causality in discourse connectives
Implicit causality

Figure 3: System framework of CL-CIS.

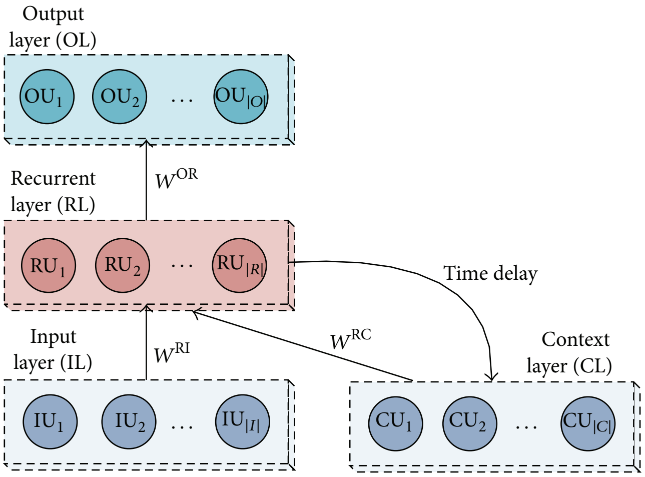


Figure 4: Architecture of SRNN.

3.2.3. Category and Causal Mapping (CCM). This module constructs the category-causal-effect mapping relations; for example, a virus from a predefined category causes specific disease-related symptoms, such as a swelling of the spleen (splenomegaly). The construction mechanism, to a great extent, is based on collection, storage, and indexing of massive cases.

3.2.4. Causal Learning (CauL). As analyzed before, a simple connectionist one-layer network that may be used to understand the task of when to activate category knowledge when learning about novel effects. This module adopts a simple recurrent neural network (SRNN) to simulate the connectionist model.

Symbols in Figure 4 are defined in Table 5: the first order weight matrices $\mathbf{W}^{\mathbf{RI}}$ and $\mathbf{W}^{\mathbf{OR}}$ fully connect the units of the input layer (IL), the recurrent layer (RL), and the output layer (OL), respectively, as in the feed forward multilayer perceptron (MLP). The current activities of recurrent units $\mathrm{RU}^{(t)}$ are fed back through time delay connections to the context layer, which is presented as $\mathrm{CU}^{(t+1)} = \mathrm{RU}^{(t)}$.

Therefore, each unit in recurrent layer is fed by activities of all recurrent units from previous time step through

Table 5: Definition of SRNN Symbols.

Symbols	Definition
IU	A unit of input layer
RU	A unit of recurrent layer
CU	A unit of context layer
OU	A unit of output layer
$\lvert I\rvert$	The number of units in IL
$\lvert R\rvert$	The number of units in RL
$\lvert C\rvert$	The number of units in CL
$\lvert O\rvert$	The number of units in OL
$\mathbf{W}^{\mathbf{RI}}$	The weight vector from IL to RL
$\mathbf{W}^{\mathbf{RC}}$	The weight vector from CL to RL
$\mathbf{W}^{\mathbf{OR}}$	The weight vector from RL to OL

recurrent weight matrix $\mathbf{W}^{\mathbf{RC}}$. The context layer, which is composed of activities of recurrent units from previous time step, can be viewed as an extension of input layer to the recurrent layer. Above working procedure represents the memory of the network via holding contextual information from previous time steps.

The weight matrices W^{RI}, W^{RC}, and W^{OR} are presented as follows:

$$\begin{aligned}
W^{\mathrm{RI}} &= \left[\left(w_1^{\mathrm{RI}}\right)^T, \left(w_2^{\mathrm{RI}}\right)^T, \dots, \left(w_{|R|}^{\mathrm{RI}}\right)^T\right] \\
&= \begin{bmatrix} w_{11}^{\mathrm{ri}} & w_{12}^{\mathrm{ri}} & \cdots & w_{1,|R|}^{\mathrm{ri}} \\ w_{21}^{\mathrm{ri}} & w_{22}^{\mathrm{ri}} & \cdots & w_{1,|R|}^{\mathrm{ri}} \\ \vdots & \vdots & \ddots & \vdots \\ w_{|I|,1}^{\mathrm{ri}} & w_{|I|,2}^{\mathrm{ri}} & \cdots & w_{|I|,|R|}^{\mathrm{ri}} \end{bmatrix}, \\
W^{\mathrm{RC}} &= \left[\left(w_1^{\mathrm{RC}}\right)^T, \left(w_2^{\mathrm{RC}}\right)^T, \dots, \left(w_{|R|}^{\mathrm{RC}}\right)^T\right] \\
&= \begin{bmatrix} w_{11}^{\mathrm{rc}} & w_{12}^{\mathrm{rc}} & \cdots & w_{1,|R|}^{\mathrm{rc}} \\ w_{21}^{\mathrm{rc}} & w_{22}^{\mathrm{rc}} & \cdots & w_{1,|R|}^{\mathrm{rc}} \\ \vdots & \vdots & \ddots & \vdots \\ w_{|C|,1}^{\mathrm{rc}} & w_{|C|,2}^{\mathrm{rc}} & \cdots & w_{|C|,|R|}^{\mathrm{rc}} \end{bmatrix}, \qquad (1) \\
W^{\mathrm{OR}} &= \left[\left(w_1^{\mathrm{OR}}\right)^T, \left(w_2^{\mathrm{OR}}\right)^T, \dots, \left(w_{|O|}^{\mathrm{OR}}\right)^T\right] \\
&= \begin{bmatrix} w_{11}^{\mathrm{or}} & w_{12}^{\mathrm{or}} & \cdots & w_{1,|O|}^{\mathrm{or}} \\ w_{21}^{\mathrm{or}} & w_{22}^{\mathrm{or}} & \cdots & w_{1,|O|}^{\mathrm{or}} \\ \vdots & \vdots & \ddots & \vdots \\ w_{|R|,1}^{\mathrm{or}} & w_{|R|,2}^{\mathrm{or}} & \cdots & w_{|R|,|O|}^{\mathrm{or}} \end{bmatrix}.
\end{aligned}$$

In the above formulations, $(w_k^{\mathrm{RI}})^T$ is the transpose of w_k^{RI} for the instance of W^{RI}, where w_k^{RI} is a row vector and $(w_k^{\mathrm{RI}})^T$ is the column vector of the same elements. The vector $w_k^{\mathrm{RI}} = (w_{1k}^{\mathrm{ri}}, w_{2k}^{\mathrm{ri}}, \dots, w_{|I|,k}^{\mathrm{ri}})$ represents the weights from all the input layer units to the recurrent (hidden) layer unit RU_k. The same conclusion applies with W^{RC} and W^{OR}.

Given an input pattern in time $\mathbf{t}$, $\mathbf{IU}^{(\mathbf{t})} = (\mathrm{IU}_1^{(t)}, \mathrm{IU}_2^{(t)}, \dots, \mathrm{IU}_{|I|}^{(t)})$, and recurrent activities $\mathbf{RU}^{(\mathbf{t})} = (\mathrm{RU}_1^{(t)}, \mathrm{RU}_2^{(t)}, \dots, \mathrm{RU}_{|R|}^{(t)})$ for the ith recurrent unit, the net input $\mathrm{RU}_i'^{(t)}$ and output activity $\mathrm{RU}_i^{(t)}$ are calculated as follows:

$$\begin{aligned}
\mathrm{RU}_i'^{(t)} &= \mathbf{IU}^{(\mathbf{t})} \cdot \left(w_i^{\mathrm{RI}}\right)^T + \mathbf{RU}^{(\mathbf{t-1})} \cdot \left(w_i^{\mathrm{RC}}\right)^T \\
&= \sum_{j=1}^{|I|} \mathrm{IU}_j^{(t)} w_{ji}^{\mathrm{ri}} + \sum_{j=1}^{|R|} \mathrm{RU}_j^{(t-1)} w_{ji}^{\mathrm{rc}}, \qquad (2) \\
\mathrm{RU}_i^{(t)} &= f\left(\mathrm{RU}_i'^{(t)}\right).
\end{aligned}$$

For the kth output unit, its net input $\mathrm{OU}_k'^{(t)}$ and output activity $\mathrm{OU}_k^{(t)}$ are calculated as (3). Consider the following:

$$\begin{aligned}
\mathrm{OU}_k'^{(t)} &= \mathbf{RU}^{(\mathbf{t})} \cdot \left(w_k^{\mathrm{OR}}\right)^T = \sum_{j=1}^{|R|} \mathrm{RU}_j^{(t)} w_{jk}^{\mathrm{or}}, \qquad (3) \\
\mathrm{OU}_k^{(t)} &= f\left(\mathrm{OU}_k'^{(t)}\right).
\end{aligned}$$

Here, the activation function f applies the logistic sigmoid function (4) in this paper. Consider the following:

$$f(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{1 + e^x}. \qquad (4)$$

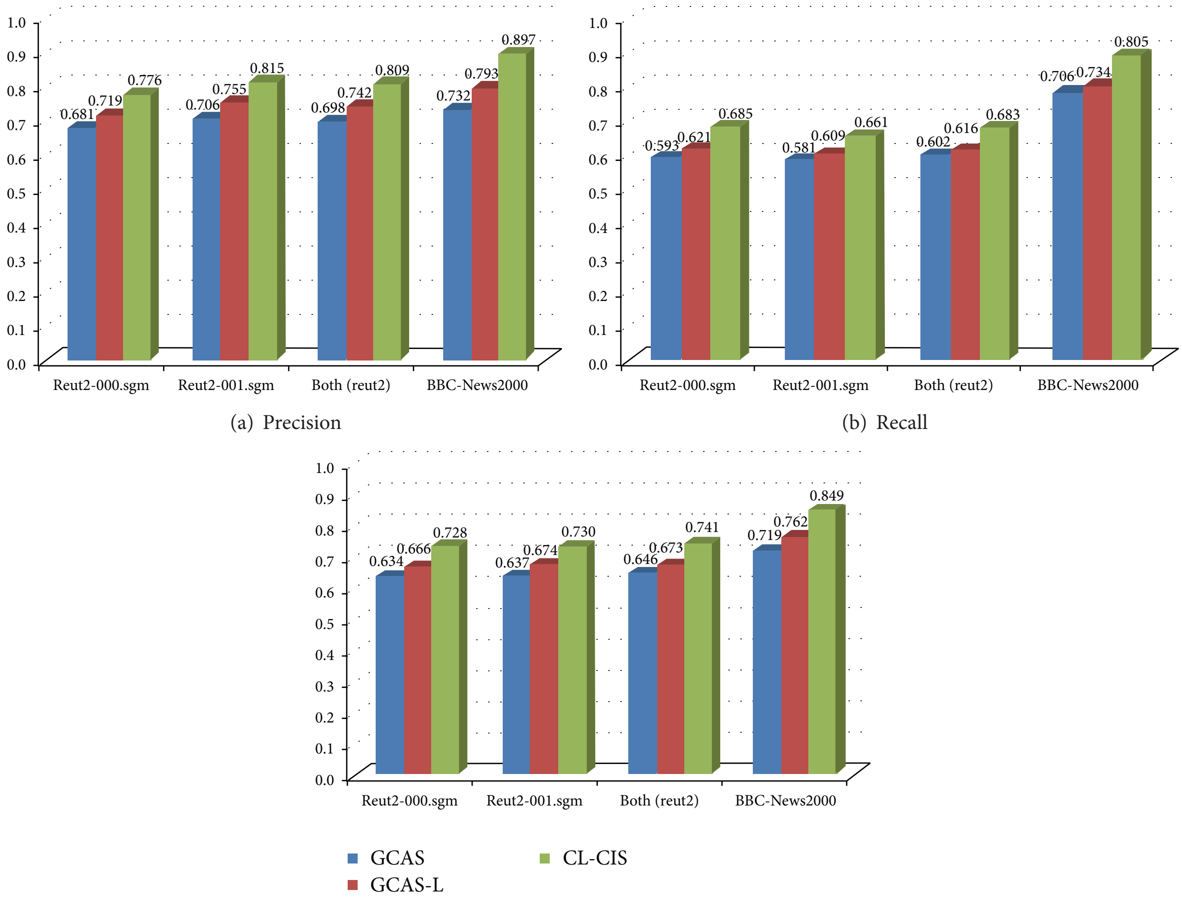
3.2.5. Building Causal-Effect Relations (BCER). This module builds up the "cause-effect" pairs and connections with the inputs of SRNN and corresponding outputs. The causality detection results could include four possible types: (1) single cause-effect pairs in which any two pairs are independent from each other; (2) cause-effect chains, which are formed with more than one cause-effect pairs connected together (an effect is a cause of another effect); (3) one-cause-multiple-effect pairs; (4) multiple-cause-one-effect pairs. All of the causality connections are archived in a database. Both CauL and BCER modules exploit three libraries (databases): causality in verbs, causality in discourse connectives, and implicit causality, for reference.

3.2.6. Testing Causal-Effect Relations (TCER). This module tests and verifies a new processed causal-effect relation with our existing and expanding collection of massive causal-effect relations. The concrete technology includes comparison of triples ⟨Category, Cause, Effect⟩.

4. Experiments and Results

Our experiments test general causality analysis system (GCAS), general causality analysis system with learning (GCAS-L), and CL-CIS together, in order to examine and reveal the assertion that category information is a necessary supplement for causality extraction and has impact on subsequent learning-based cause-effect identification.

In our experiments, we have used two text collections: (1) Reuters-21578, currently the most widely used test collection for text processing research; (2) BBC-News2000, collected by a self-developed Web Crawler from http://www.bbc.co.uk/.



(a) Precision

(b) Recall

(c) *F*-measure (balanced)

Figure 5: Evaluation results of GCAS, GCAS-L, and CL-CIS.

BBC-News2000 collected 2000 news articles which have been pruned off unnecessary Web page elements, such as html tags, images, URLs, and so forth. BBC-News2000 is not categorized and is treated as a whole hybrid.

30 judges are involved to manually search cause-effect pairs and construct standard causality references (SCR), SCR-Reuters for Reuters-21578, and SCR-BBC for BBC-News2000. Due to limited human resources, in current stage, SCR-Reuters only covers 2000 documents, *newid* ranges from 1 to 2000, contained in files "reut2-000.sgm" and "reut2-001.sgm." The rest of the files of Reuters-21578 are still involved in the training phase.

The performances of GCAS, GCAS-L, and CL-CIS are evaluated with precision, recall, and *F*-measure [21], the traditional measures that have been widely applied by most information retrieval systems to analyze and evaluate their performance. The *F*-measure is a harmonic combination of the precision and recall values used in information retrieval. As shown in Tables 6, 7, and 8, the experimental results state that (1) GCAS scores from 0.681 to 0.732 on precision and from 0.581 to 0.706 on recall; (2) GCAS-L scores from 0.719 to 0.793 on precision and from 0.609 to 0.734 on recall; (3) CL-CIS scores from 0.776 to 0.897 on precision and from 0.661 to 0.805 on recall. Figure 5 explicitly states that (1) the general causality analysis system with learning (GCAS-L) performances better than the general causality analysis system (GCAS) on all evaluation measures; (2) the causality analysis system strengthened with causal and category learning (CL-CIS) exceeds GCAS-L in the meantime.

Table 6: Experimental results of GCAS.

Experiment files	Precision	Recall	*F*-measure
reut2-000.sgm (newid: 1–1000)	0.681	0.593	0.634
reut2-001.sgm (newid: 1001–2000)	0.706	0.581	0.637
Both (reut2) (newid: 1–2000)	0.698	0.602	0.646
BBC-News2000	0.732	0.706	0.719

Table 7: Experimental results of GCAS-L.

Experiment files	Precision	Recall	*F*-measure
reut2-000.sgm (newid: 1–1000)	0.719	0.621	0.666
reut2-001.sgm (newid: 1001–2000)	0.755	0.609	0.674
Both (reut2) (newid: 1–2000)	0.742	0.616	0.673
BBC-News2000	0.793	0.734	0.762

Table 8: Experimental results of CL-CIS.

Experiment files	Precision	Recall	F-measure
reut2-000.sgm (newid: 1–1000)	0.776	0.685	0.728
reut2-001.sgm (newid: 1001–2000)	0.815	0.661	0.730
Both (reut2) (newid: 1–2000)	0.809	0.683	0.741
BBC-News2000	0.897	0.805	0.849

5. Concluding Remarks

In recent years, detection and clarification of cause-effect relationships among texts, events, or objects has been elevated to a prominent research topic of natural and social sciences over the human knowledge development history.

This paper demonstrates a novel comprehensive causality extraction system (CL-CIS) integrated with the means of category-learning. CL-CIS considers cause-effect relations in both explicit and implicit forms and especially practices the relation between category and causality in computation.

CL-CIS is inspired with cognitive philosophy in category and causality. In causality extraction and induction tasks, CL-CIS implements a simple recurrent neural network (SRNN) to simulate human associative memory, which has the ability to associate different types of inputs when processing information.

In elaborately designed experimental tasks, CL-CIS has been examined in full with two text collections, Reuters-21578 and BBC-News2000. The experimental results have testified the capability and performance of CL-CIS in construction of cause-effect relations and also confirmed the expectation that the means of causal and category learning will improve the precision and coverage of causality induction in a notable manner.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work is financially supported by the National Natural Science Foundation of China (Grant no. 61003126).

References

[1] http://en.wikipedia.org/wiki/Causality.

[2] J. Pearl, *Causality: Models, Reasoning and Inference*, Cambridge University Press, New York, NY, USA, 2000.

[3] D.-S. Chang and K.-S. Choi, “Incremental cue phrase learning and bootstrapping method for causality extraction using cue phrase and word pair probabilities,” *Information Processing and Management*, vol. 42, no. 3, pp. 662–678, 2006.

[4] S. Kim, R. H. Bracewell, and K. M. Wallace, “A framework for automatic causality extraction using semantic similarity,” in *Proceedings of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE ’07)*, pp. 831–840, Las Vegas, Nev, USA, September 2007.

[5] T. Inui, K. Inui, and Y. Matsumoto, “Acquiring causal knowledge from text using the connective markers,” *Transactions of the Information Processing Society of Japan*, vol. 45, no. 3, pp. 919–932, 2004.

[6] D. Marcu and A. Echihabi, “An unsupervised approach to recognizing discourse relations,” in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics Conference (ACL ’02)*, pp. 368–375, University of Pennsylvania, Philadelphia, Pa, USA, 2002.

[7] C. Pechsiri and A. Kawtraku, “Mining causality from texts for question answering system,” *IEICE Transactions on Information and Systems*, vol. 90, no. 10, pp. 1523–1533, 2007.

[8] R. Girju, “Automatic detection of causal relations for question answering,” in *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, 2003.

[9] K. Torisawa, “Automatic extraction of commonsense inference rules from corpora,” in *Proceedings of the 9th Annual Meeting of the Association for Natural Language Processing*, 2003.

[10] Y. Guo, N. Hua, and Z. Shao, “Cognitive causality detection with associative memory in textual events,” in *Proceedings of the IEEE International Symposium on Information Engineering and Electronic Commerce (IEEC ’09)*, pp. 140–144, Ternopil, Ukraine, May 2009.

[11] P. Thwaites, J. Q. Smith, and E. Riccomagno, “Causal analysis with Chain Event Graphs,” *Artificial Intelligence*, vol. 174, no. 12-13, pp. 889–909, 2010.

[12] M. R. Waldmann and Y. Hagmayer, “Categories and causality: the neglected direction,” *Cognitive Psychology*, vol. 53, no. 1, pp. 27–58, 2006.

[13] G. L. Murphy, *The Big Book of Concepts*, MIT Press, Cambridge, Mass, USA, 2002.

[14] B. Rehder, “A causal-model theory of conceptual representation and categorization,” *Journal of Experimental Psychology: Learning Memory and Cognition*, vol. 29, no. 6, pp. 1141–1159, 2003.

[15] B. Rehder, “Categorization as casual reasoning,” *Cognitive Science*, vol. 27, no. 5, pp. 709–748, 2003.

[16] B. Rehder and R. Hastie, “Causal knowledge and categories: the effects of causal beliefs on categorization, induction, and similarity,” *Journal of Experimental Psychology: General*, vol. 130, no. 3, pp. 323–360, 2001.

[17] B. Rehder and R. Hastie, “Category coherence and category-based property induction,” *Cognition*, vol. 91, no. 2, pp. 113–153, 2004.

[18] Y. Lien and P. W. Cheng, “Distinguishing genuine from spurious causes: a coherence hypothesis,” *Cognitive Psychology*, vol. 40, pp. 87–137, 2000.

[19] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller, “Introduction to WordNet: an on-line lexical database,” *International Journal of Lexicography*, vol. 3, no. 4, pp. 235–312, 1990.

[20] A. W. Koornneef and J. J. A. Van Berkum, “On the use of verb-based implicit causality in sentence comprehension: evidence from self-paced reading and eye tracking,” *Journal of Memory and Language*, vol. 54, no. 4, pp. 445–465, 2006.

[21] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*, The MIT Press, Cambridge, Mass, USA, 4th edition, 2001.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 460593, 6 pages
http://dx.doi.org/10.1155/2014/460593

Research Article

Novel Web Service Selection Model Based on Discrete Group Search

Jie Zhai, Zhiqing Shao, Yi Guo, and Haiteng Zhang

Department of Computer Science and Engineering, East China University of Science and Technology, Shanghai 200237, China

Correspondence should be addressed to Zhiqing Shao; zshao@ecust.edu.cn

Received 2 March 2014; Accepted 11 March 2014; Published 15 April 2014

Academic Editor: Yu-Bo Yuan

In our earlier work, we present a novel formal method for the semiautomatic verification of specifications and for describing web service composition components by using abstract concepts. After verification, the instantiations of components were selected to satisfy the complex service performance constraints. However, selecting an optimal instantiation, which comprises different candidate services for each generic service, from a large number of instantiations is difficult. Therefore, we present a new evolutionary approach on the basis of the discrete group search service (D-GSS) model. With regard to obtaining the optimal multiconstraint instantiation of the complex component, the D-GSS model has competitive performance compared with other service selection models in terms of accuracy, efficiency, and ability to solve high-dimensional service composition component problems. We propose the cost function and the discrete group search optimizer (D-GSO) algorithm and study the convergence of the D-GSS model through verification and test cases.

1. Introduction

We have proposed a novel approach for the verification of service composition with contracts [1]. The approach properties of the generic specification [2] in Tecton [3] are verified by the Violet [4] system. After verification, a global optimum is selected from a number of instantiations of web service composition components with multiple QoS constraints. Compared with other algorithms that evaluate all feasible composition instantiations (e.g., integer programming [5]), evolutionary algorithms (EAs) (e.g., genetic algorithm [6]), which are nature-inspired optimization algorithms, are simple and flexible. Given their characteristics, EAs have been used to solve the service selection problem. We proposed a novel optimization model named discrete group search service (D-GSS) that mainly employs the group search optimizer (GSO) algorithm [7]. The D-GSS model has competitive performance compared with other EAs in terms of accuracy, convergence speed, and ability to solve high-dimensional multimodal problems. On the basis that GSO can solve continuous optimization problems and that service selection can solve discrete instantiations, we present an evolutionary algorithm called discrete group search optimizer (D-GSO) to select the best instantiation that has the lowest cost evaluated by the cost function. The cost function consists of the utility function and the weight for every QoS attribute. We also verify and simulate results to analyze the convergence of the D-GSS model.

The rest of the paper is organized as follows. Section 2 describes the D-GSS model. Section 3 presents a detailed introduction of the cost function, and Section 4 discusses the D-GSO algorithm and applies the algorithm for the problem on searching for the global optimum from discrete instantiations. Section 5 introduces the convergence analysis of the D-GSS model. Finally, Section 6 concludes the paper.

2. Distribute Group Search Optimizer

In this paper, we present a novel algorithm named D-GSS toward the atomic service selection of composing complex services with multiple QoS constraints. The population of the D-GSO algorithm is called a group searching for unknown optima in the services composition problem and each individual in the population is called a member.

In the n-dimensional search space I about composition component, every dimension represents a class of generic service denoted as I_i. The ith member X_i in the space I is denoted as follows:

$$I = \{I_1, I_2, \ldots, I_n\},$$
$$X_i = \{x_i^1, x_i^2, \ldots, x_i^n\}, \qquad (1)$$

where $x_i^j \in I_j$. The ith member X_i at the kth iteration has a current position $X_i^k \in R^n$ and X_i^k is corresponding to an instantiation of services composition component.

A head angle ϕ_i^k is the position of the member; $\phi_i^k = (\phi_{i_1}^k, \phi_{i_2}^k, \ldots, \phi_{i_{(n-1)}}^k) \in R^{n-1}$. The search direction of the ith member, which is a unit vector $D_i^k(\phi_i^k) = (d_{i_1}^k, d_{i_2}^k, \ldots, d_{i_n}^k) \in R^n$ that can be calculated from ϕ_i^k via a polar to Cartesian coordinate transformation [7]:

$$d_{i_1}^k = \prod_{q=1}^{n-1} \cos\left(\phi_{i_q}^k\right),$$
$$d_{i_j}^k = \sin\left(\phi_{i_{(j-1)}}^k\right) \cdot \prod_{q=1}^{n-1} \cos\left(\phi_{i_q}^k\right) \quad (j = 2, \ldots, n-1), \qquad (2)$$
$$d_{i_n}^k = \sin\left(\phi_{i_{(n-1)}}^k\right).$$

In D-GSO based on GSO [7] inspired by animal behavior and animal searching behavior, a group consists of three types of members: only one producer is assumed to have the lowest cost at each searching bout, and the remaining members are assumed to be scroungers and dispersed members. At each iteration, a group member representing the most promising instantiation and conferring the lowest fitness value is chosen as the producer. It then stops and scans the environment to seek optimal instantiation. The scanning field is characterized by maximum pursuit angle $\theta_{\max}$ and maximum pursuit distance $l_{\max}$. The apex is the position of the producer. All scroungers will join the resource found by the producer according to area copying strategy. The rest of the group members will be dispersed from their current positions for randomly distributed better instantiations. To handle the bounded search space, the following strategy is employed: when a member is outside the search space, the member will return into the search space by setting the variables that violated the bounds into their previous values.

The details of D-GSO (see Figure 1) are introduced as follows.

(i) Suppose that n classes of generic services exist in the n-dimensional composition component; each class has N_i $(1 \leq i \leq n)$ candidate services in a special sequence.

(ii) Define the concrete cost function of the specific composition component. The cost function is defined by the QoS attributes of the component services as well as their integration relationships, such as sequential, parallel, conditional, or loop. Generate initial members from all instantiations and evaluate the members according to the cost function.

(iii) Choose a member with the lowest cost as producer. The producer produces on the basis of the discrete GSO algorithm.

(iv) Randomly select 80% of the remaining members to perform scrounging.

(v) The remaining members will be dispersed from their current instantiations to perform ranging.

(vi) Evaluate all members according to the cost function. If no optimal instantiation with multiple QoS constraints is found, reallocate the role of every member on the value of the cost.

3. Cost Function

A "generic service" is a collection of atomic web services with a common functionality, but different nonfunctional properties (e.g., time and quality). Each atomic service may provide a series of QoS parameters, such as service time, cost, reliability, and availability. Users can set the number of QoS values to be considered and can set the weights of the QoS values according to their requirements. In our study, each user has k QoS attribute constraints in their QoS requirements: $Q_c = [Q^1, \ldots, Q^k]$. We focus on the QoS service selection problem, in which multiple QoS constraints must be satisfied. We present the cost function to help in the selection of the best services. The following steps are involved in the creation of the cost function.

(i) Each QoS attribute must be quantitative. Service functionalities can be evaluated by several QoS properties. Some QoS attributes, for example, security and reliability, are difficult to measure quantitatively. For these criteria, we employ the linguistic expression set $L1 = \{\mathrm{VP}, \mathrm{MP}, \mathrm{P}, \mathrm{M}, \mathrm{G}, \mathrm{MG}, \mathrm{VG}\}$, where VP is very poor, MP is medium poor, P is poor, M is medium, G is good, MG is medium good, and VG is very good. When calculating the cost function, set $L1$ is transformed into the corresponding quantitative set $P1 = \{0.15, 0.3, 0.45, 0.6, 0.75, 0.9, 1\}$.

(ii) Global QoS attributes ($q_c = [q^1, \ldots, q^k]$) are needed to describe the performance of an instantiation of service composition component. Every global QoS attribute is aggregated by the QoS attributes of all atomic services considering the integration relationships of the global QoS attribute. Each service has four main basic structures: (1) the sequential structure, which represents n services that are invoked one by one; (2) the loop structure, which represents one service that is repeated p times; (3) the conditional structure, which represents only one branch that is selected to be invoked from n branches; (4) the parallel structure, which represents n branches that are invoked simultaneously. The complete structure of the service composition component consists of the above four basis structures. Every global QoS

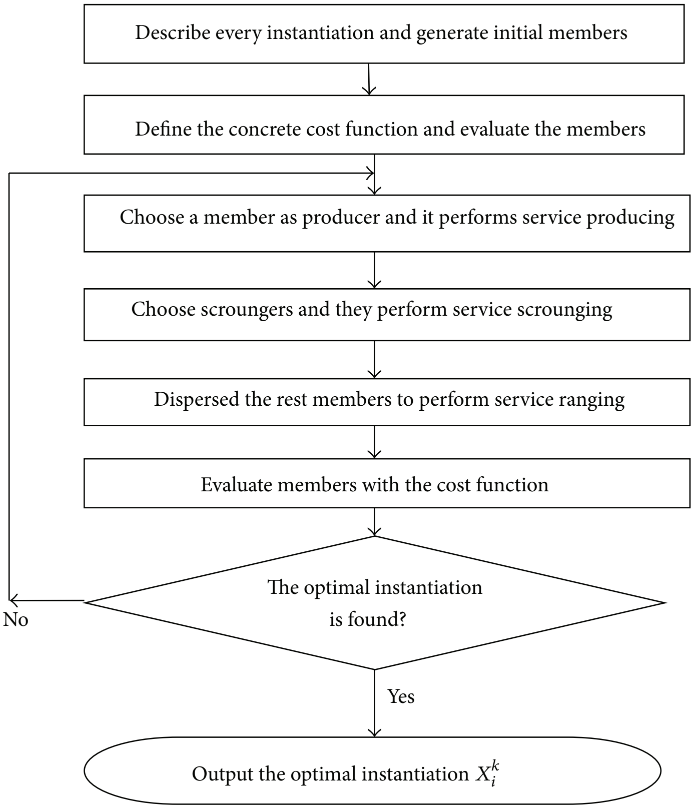

Describe every instantiation and generate initial members
Define the concrete cost function and evaluate the members
Choose a member as producer and it performs service producing
Choose scroungers and they perform service scrounging
Dispersed the rest members to perform service ranging
Evaluate members with the cost function
The optimal instantiation is found?
No
Yes
Output the optimal instantiation X_i^k

Figure 1: Flowchart of the D-GSS model.

Table 1: Aggregated methods for global QoS attributes.

Method	Sequential	Loop	Choice	Parallel
Summation	$\sum_{i=1}^{n} q^i$	pq^i	$\sum_{i=1}^{n} c^i q^i$	$\sum_{i=1}^{n} q^i$
Continued multiplication	$\prod_{i=1}^{n} q^i$	q^i	$\sum_{i=1}^{n} c^i q^i$	$\prod_{i=1}^{n} q^i$
Average	$\frac{1}{n}\sum_{i=1}^{n} q^i$	q^i	$\sum_{i=1}^{n} c^i q^i$	$\frac{1}{n}\sum_{i=1}^{n} q^i$

attribute has its own aggregated method. We sort the QoS aggregated methods into three types: (1) the summation method (e.g., cost), in which the fees must be accumulated by the user to pay for invoking the services; (2) the continued multiplication method (e.g., availability), in which global availability can be computed as the product of the ratios of all atomic service availability; (3) the average method (e.g., reputation), in which global reputation is the average value of the related service reputation. We present all particulars (see Table 1) of these three methods with sequential, parallel, conditional, or loop structures. In Table 1, c^i is a 0-1 variable. If condition c^i is satisfied, then we define $c^i = 1$; otherwise, $c^i = 0$.

(iii) After the values of $[q^1, \ldots, q^k]$ and $[Q^1, \ldots, Q^k]$ are evaluated, we present a utility function to describe the relationship between q^i and Q^i. Two types of QoS criteria are available, that is, cost and benefit. In the cost criterion, variables (e.g., response time) with higher values have lower qualities. In the benefit criterion, variables (e.g., availability) with higher values have higher qualities. The utility function synthesizes the cost and benefit criteria.

Definition 1 (utility function). Suppose that a global QoS attribute q^i $(1 \leq i \leq k)$ and its constraint Q^i of an instantiation S^j exist, the utility function is defined as follows:

$$U\left(q_i^j, Q_i^j\right) = \begin{cases} \dfrac{q_i^j}{Q_i^j}, & \text{if } q_i^j \text{ is the cost criterion,} \\ 2 - \dfrac{q_i^j}{Q_i^j}, & \text{if } q_i^j \text{ is the benefit criterion.} \end{cases} \quad (3)$$

If the global QoS attribute q^i satisfies the requirement of the QoS constraint Q^i, then $U(q_i^j, Q_i^j) \leq 1$; otherwise $U(q_i^j, Q_i^j) > 1$.

(iv) The cost function is based on the values of the utility function and the weights the user defined. The better the instantiation is, the lower the quality of the cost function result becomes.

Definition 2 (cost function). Suppose that an instantiation S^j exists in the QoS attributes $q_c = [q_1^j, \ldots, q_k^j]$, QoS constraints $Q_c = [Q_1^j, \ldots, Q_k^j]$, and the weights for each QoS attribute; then the cost function is defined as follows:

$$F\left(X_j, q_c, Q_c\right) = \sum_{i=1}^{k} w_i^j U\left(q_i^j, Q_i^j\right), \tag{4}$$

where $\sum_{i=1}^{k} w_i^j = 1$ and $U(q_i^j, Q_i^j) \leq 1$ $(1 \leq i \leq k)$.

The objective of this paper is to employ D-GSO to get the optimal solution of the following model:

$$\min\left(F\left(X_j\right)\right) = \sum_{i=1}^{k} w_i^j U\left(q_i^j, Q_i^j\right), \tag{5}$$

where $X_j \in R^n$.

4. D-GSO Algorithm

The GSO algorithm [7] designs optimum searching strategies to solve continuous optimization problems. However, service selection is a discrete problem. Therefore, we present an evolutionary algorithm named D-GSO that can handle composition components with discrete atomic services. The steps of the D-GSO algorithm are described in Algorithm 1. In the D-GSO algorithm, round(x) represents a round function for half adjust result. Suppose that subX_i^h. represents $[1_i^h, 2_i^h, \ldots, n_i^h]$, which are the subscripts of atomic services composing an instantiation X_i^h about the ith member X_i at the hth iteration. At the $(h+1)$ iteration, the transformation of the subscripts by the following formulas is $[1_i^{h+1}, 2_i^{h+1}, \ldots, n_i^{h+1}]$ relating to a new instantiation (see Algorithm 1).

5. Convergence Analysis of the D-GSS Model

5.1. Convergence Verification. In this section, we verified the convergence of the D-GSS model. After n iterations, the best instantiation with the lowest cost can be determined with the cooperation of the producer and some scroungers and rangers.

Lemma 3. *If X represents the space of all instantiations X_i^k and P represents the space of the producer, then $X = P$.*

Proof. (1) $l_{\max}$ denotes the maximum distance between two points in space X. By using (3) to (6), we can equate space P to a sphere that has center X_h^p possessing sub(X_h^p) and radius $l_{\max}$. Thus, $X \subset P$.

(2) The following strategy is employed by using the D-GSS model: when a member in space P is outside space X, the member will return into space X by setting the variables that violated the bounds to their previous values. Therefore, $P \subset X$.

(3) Thus, we conclude that $X = P$. □

Theorem 4. *The costs of instantiations in the group will converge to the global optimum that corresponds to the best instantiation with the lowest cost.*

Proof. In the D-GSS model at the hth iteration,

(1) the producer S^p behaves according to (ii)–(iv) in Algorithm 1. By applying the D-GSO algorithm, we can derive the following:

$$\begin{aligned} &\operatorname{cost}\left(X_{h+1}^p\right) \\ &\quad = \min\left(\operatorname{cost}\left(X_h^p\right), \operatorname{cost}\left(X_z\right), \operatorname{cost}\left(X_r\right), \operatorname{cost}\left(X_l\right)\right), \end{aligned} \tag{6}$$

(2) the scroungers X_{h+1}^s will approach the producer through (vii) in Algorithm 1,

(3) the rangers X_{h+1}^r will disperse from a group to perform random walks via (viii) and (ix) in Algorithm 1 to avoid entrapments in the local minima,

(4) finally, we calculate the costs of all instantiations in the group and reallocate their roles. The cost of the new producer is shown as follows:

$$\operatorname{cost}\left(X_{h+1}^p\right) = \min\left(\operatorname{cost}\left(X_{h+1}^p\right), \operatorname{cost}\left(X_{h+1}^s\right), \operatorname{cost}\left(X_{h+1}^r\right)\right). \tag{7}$$

We conclude that $\operatorname{cost}(X_{h+1}^p) \leq \operatorname{cost}(X_h^p)$ by using (6) and (7), which means that the cost of the producer is monotonically decreasing. A global optimum, which has the lowest cost in all instantiations, exists. As stated in the proof of Lemma 3, $X = P$. Therefore, the infimum of $\operatorname{cost}(X^p)$ is cost (global optimum); that is, after n iterations, the instantiation X^p converges to the global optimum. □

5.2. Simulation Convergence Results. The parameter setting of the D-GSS model is summarized as follows. M classes of generic services are present in the complex composition component, in which each class has 50 candidate services that has 10 QoS attributes. The service requestor provides 10 QoS attribute constraints as well as the weights for each QoS attribute. Overall, 51 initial instantiations X^i with $U(q_t^i, Q_t^i) \leq 1$ $(1 \leq t \leq 10)$ are selected at random in all instantiations. The initial head angle ϕ^0 of each individual is set to $(\pi/4, \ldots, \pi/4)$. The constant a is given by round$(\sqrt{n+1})$. The maximum pursuit angle $\theta_{\max}$ is π/a^2. The maximum turning angle $\alpha_{\max}$ is set to $\theta_{\max}/2$. Suppose $n = 10, 100$; the relations between the cost of the producer and the iteration times within 500 runs are shown in Figure 2. The experimental results show that the cost of the producer always converges to the optimum of the low- or high-dimensional service composition component.

```
Algorithm. D-GSO
Step 1. $N$ classes of generic services are present in the composition component, where each class has $N_i$ $(1 \le i \le n)$
candidate services. The maximum pursuit distance $l_{\max}$ is calculated from the following equation:
$l_{\max} = \sqrt{\sum_{i=1}^{n} N_i^2},$    (i)
Each candidate service has $k$ QoS attributes, which are rearranged into $[q_{\text{des}}^1, \ldots, q_{\text{des}}^k]$ according to the weights in
descending sequence. The candidate services of generic service $I_i$ $(1 \le i \le n)$ are reordered into a set $I_i^{\text{order}} = [x_i^0, \ldots, x_i^{N_i}]$
with reference to $q_{\text{des}}^1, \ldots, q_{\text{des}}^k$ in turn.

Step 2. Set $h := 0$;
Randomly initialize $r$ instantiations $X_i\ [x_i^1, x_i^2, \ldots, x_i^n]$ $(1 \le i \le r)$ with $U(q_t^i, Q_t^i) \le 1$ $(1 \le t \le k)$ of services composition
component and head angle $\phi_i$ of all initial instantiations;
Calculate the values of initial instantiations according to the cost function;
WHILE (the best instantiation is not found)
    FOR (each instantiation $X^i$ where $U(q_t^i, Q_t^i) \le 1$ $(1 \le t \le k)$ in the group)
Choose the producer:
        Find the producer $X^p$ with the lowest cost in the group;
Perform producing:
    (a) The producer will scan at zero degree and then scan laterally by randomly sampling three instantiations in the scanning
        field: one instantiation at zero degree, one instantiation in the right-hand side of the hypercube, and one instantiation in the
        left-hand side of the hypercube. $r_1 \in R^1$ is a normally distributed random number with mean 0 and standard deviation 1,
        where as $r_2 \in R^{n-1}$ is a uni-formly distributed random sequence in the range (0, 1);
        $\text{sub}\left(X_z\right) = \text{sub}\left(X_h^p\right) + \text{round}(r_1 l_{\max} D_h^p(\phi_h)),$    (ii)
        $\text{sub}\left(X_r\right) = \text{sub}\left(X_h^p\right) + \text{round}\left(r_1 l_{\max} D_h^p\left(\phi_h + r_2 \frac{\theta_{\max}}{2}\right)\right),$    (iii)
        $\text{sub}\left(X_l\right) = \text{sub}\left(X_h^p\right) + \text{round}\left(r_1 l_{\max} D_h^p\left(\phi_h - r_2 \frac{\theta_{\max}}{2}\right)\right),$    (iv)
    (b) The producer will find the best instantiation $X^i$ where $U(q_t^i, Q_t^i) \le 1$ $(1 \le t \le k)$ with the lowest cost.
        If the best instantiation has a lower cost compared with the current instantiation, then the best instantiation
        will be chosen; otherwise, the current instantiation will remain and turn its head to a new randomly generated angle.
        $\alpha_{\max} \in R^1$ is the maximum turning angle;
        $\phi^{h+1} = \phi^h + r_2 \alpha_{\max},$    (v)
    (c) If the producer cannot find a better instantiation after $a$ iterations, then the producer will turn its head back to zero degree;
        $\phi^{h+a} = \phi^h,$    (vi)
Perform scrounging:
        Randomly select 80% members from the rest of the instantiations to perform scrounging.
        The area copying behavior of the $i$th scrounger can be modeled as a random walk toward the producer.
        In (vii), $r_3 \in R^n$ is a uniform random sequence in the range (0, 1);
        $\text{sub}\left(X_i^{h+1}\right) = \text{sub}\left(X_i^h\right) + \text{round}\left(r_3 \circ \left(\text{sub}\left(X_p^h\right) - \text{sub}\left(X_i^h\right)\right)\right),$    (vii)
Perform dispersion:
        The rest of the instantiations will be dispersed to perform ranging: (1) generate a random head angle by using (v); (2) choose
        a random distance $l_i$ from the Gauss distribution by using (viii); transform into the new instantiation by using (ix);
        $l_i = a \cdot r_1 l_{\max},$    (viii)
        $\text{sub}\left(X_i^{h+1}\right) = \text{sub}\left(X_i^h\right) + \text{round}\left(l_i D_i^h\left(\phi^{h+1}\right)\right).$    (ix)

Calculate fitness:
        Calculate the values of the current instantiations according to the cost function;
    END FOR
    Set $h := h + 1$;
END WHILE
```

Algorithm 1: Procedure for the D-GSO algorithm.

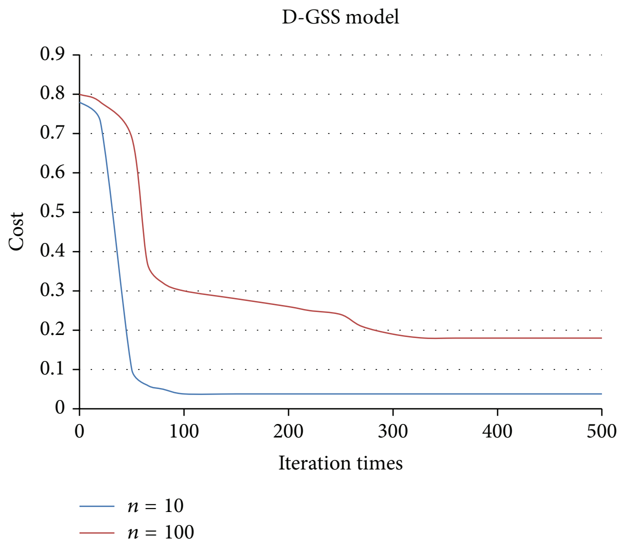


FIGURE 2: Convergence for $n = 10, 100$.

The experiments were conducted on a PC with 2.50 GHz Intel Processor and 8.0 GB RAM. All programs were written and executed in Java. The operating system was Microsoft Windows 7.

6. Conclusion

In this paper, we describe a new evolutionary approach for multiconstraints service selection on the basis of the D-GSS model. We propose the cost function and the D-GSO algorithm for searching the global optimum from discrete instantiations of the service composition component. The convergence of the D-GSS model is verified via several formal proofs and simulations. This model has an outstanding advantage in terms of solving high-dimensional service composition problems. In the future, we hope to search for the global optimum under a dynamic heterogeneous environment by using the D-GSS model.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work was supported by of the National Natural Science Foundation of China under Grant no. 61003126 and no. 61300133.

References

[1] Z. Jie, S. Zhiqing, G. Yi, and Z. Haiteng, "Generic contract-regulated web service composition specification and verification," in *Proceedings of the International Conference on Information Technology and Software Engineering (ITSE '12)*, vol. 3, pp. 137–145, Beijing, China, 2012.

[2] Z. Jie and S. Zhiqing, "Specification and verification of generic web service composition," *Computer Application and Software*, vol. 28, no. 11, pp. 64–68, 2011.

[3] D. R. Musser and S. Zhiqing, "Concept use or concept refinement: an important distinction in building generic specifications," in *Proceedings of the 4th International Conference on Formal Engineering Methods (ICFEM '02)*, pp. 132–143, Shanghai, China, 2002.

[4] Z. Jie and S. Zhiqing, "The proof system based on tecton—violet," *Journal of East China University of Science and Technology*, vol. 31, no. 2, pp. 198–202, 2005.

[5] L. Z. Zeng, B. Benatallah, A. H. H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang, "QoS-aware middleware for Web services composition," *IEEE Transactions on Software Engineering*, vol. 30, no. 5, pp. 311–327, 2004.

[6] G. Canfora, M. D. Penta, R. Esposito, and M. L. Villani, "An approach for QoS-aware service composition on algorithms," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '05)*, pp. 1069–1075, Washington, DC, USA, June 2005.

[7] S. He, Q. H. Wu, and J. R. Saunders, "Group search optimizer: an optimization algorithm inspired by animal searching behavior," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 973–990, 2009.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 373902, 7 pages
http://dx.doi.org/10.1155/2014/373902

Research Article

Web Service Reputation Evaluation Based on QoS Measurement

Haiteng Zhang, Zhiqing Shao, Hong Zheng, and Jie Zhai

Department of Computer Science and Engineering, East China University of Science and Technology, Shanghai 200237, China

Correspondence should be addressed to Zhiqing Shao; zshao@ecust.edu.cn

Received 9 March 2014; Accepted 13 March 2014; Published 13 April 2014

Academic Editor: Yu-Bo Yuan

In the early service transactions, quality of service (QoS) information was published by service provider which was not always true and credible. For better verification the trust of the QoS information was provided by the Web service. In this paper, the factual QoS running data are collected by our WS-QoS measurement tool; based on these objectivity data, an algorithm compares the difference of the offered and measured quality data of the service and gives the similarity, and then a reputation evaluation method computes the reputation level of the Web service based on the similarity. The initial implementation and experiment with three Web services' example show that this approach is feasible and these values can act as the references for subsequent consumers to select the service.

1. Introduction

Nowadays, Web services are one of the important innovations in software which bring many benefits in software design and implementation. With the fast growth of Web services, a large number of Web services with the same or similar function are developed and released. How to select a suitable and best service has become an important research topic. The Web service selection technology based on QoS has been referred to for solving this problem, which considers distinguishing those Web services with the same function using a set of different QoS levels [1].

The existing QoS-based services selection approaches always assume that the QoS data coming from service providers are effective and trustworthy. However, the values of QoS attributes which are provided by service providers may be incredible, since service providers sometimes may advertise higher QoS data than the factual level of the service in order to attract more users to use their services and so gain better benefits [2]. For example, the maximum response time of these services may be increased, while the invocation rate remains under a certain threshold during runtime. Therefore, how to give the objective and effective evaluation to service provider's reputation to help the consumer to reference and choose the appropriate service becomes a problem to solve [3].

To ensure the impartiality and objectivity of a Web service reputation evaluation, this paper proposes a trust Web service reputation evaluation framework based on QoS similarity of the factual values and the advertised values. Firstly, a Web services QoS measurement tool which is independent of service providers or consumers was developed, which provides an automatic approach on measuring and storing QoS values of the service. Secondly, Web service reputation evaluating component computes the similarity of advertised QoS values and factual values, and then the similarity is used to evaluate the reputation level of the Web service. Lastly, a set of experiments are given, which show that this approach can effectively evaluate the reputation of the service provider and thus can strengthen the effectiveness of the service selection.

The rest of this paper is organized as follows. In Section 2, we give an overview of our Web service reputation evaluation framework. Section 3 illustrates the core component of our WS-QoS measurement tool. The similarity algorithm and QoS reputation evaluation method are given in Section 4. In Section 5 we present the main implementation and the experiment to verify the efficiency of our method. This is followed by an overview of the related work in Section 6. Section 7 concludes our paper and presents further research directions.

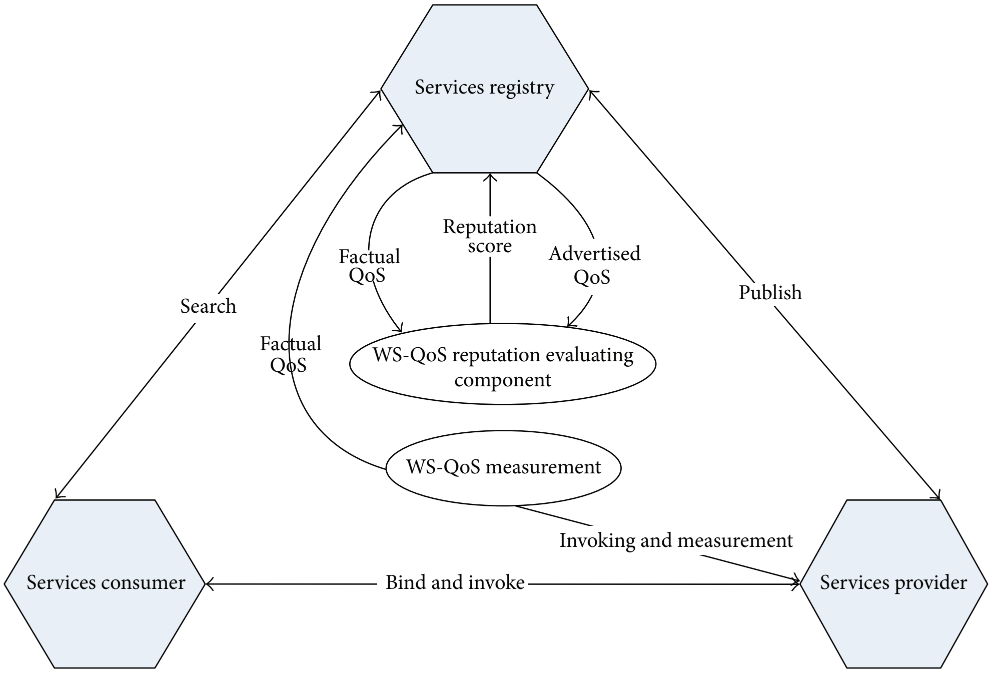


Figure 1: The Web service reputation evaluation framework.

2. The Web Service Reputation Evaluation Framework

The Web service reputation evaluation framework is shown in Figure 1. The framework consists of the basic Web service model components like the Web service provider, Web service consumer, and the Web service registry. Two major components are introduced into traditional Web services architecture to realize the reputation evaluation of the service provider.

WS-QoS measurement tool is a client side technique which works completely on Web service consumer and provider independently. It measures the performance related QoS values which are achieved by dynamic invoking Web services together with aspect-oriented programming (AOP). So that factual QoS values are provided and stored by this component.

WS-QoS reputation evaluating component supports service reputation measurement based on QoS similarity. Service providers issue the advertised values of the QoS information into Service Registry Center. WS-QoS measurement also gives feedback of the factual values of QoS to Service Registry Center after invoking the service. QoS similarity is computed firstly according to the differences between advertised QoS and factual QoS values, and then the Web service reputation score was given based on these similarities.

3. The WS-QoS Measurement Tool

To objectively measure service related quality information, the WS-QoS measurement tool is designed to acquire QoS attribute values for a given set of Web services. The main processes of the WS-QoS measurement tool are depicted in Figure 2. In the first phase, Web services description language (WSDL) file is acquired from UDDI. The WSDL file is parsed to get service related information, and the test data are generated for each input element of the operations. As a next step, the Web service stub classes are generated as Java files by using the WSDL2Java tool from Axis, which gives the service invoker all the exposed methods and parameters' types by the Web service. In the third step, the Web service invoker assembles the generated test date to stub code to cause the Web service to be invoked and its response results and status to be collected which can be used to compute the QoS typical parameters such as availability, reliability, and accessibility. In the last step, timeAspect code weaves time measurement codes before and after the byte code of the Web service invoking method; then the start time and end time of Web service call are acquired, and the Web service response time is computed.

3.1. *Test Case Generation Based on WSDL.*

In the distributed environment, the service provider exposes the functionality of the service in the form of a Web services description language. WSDL describes Web services by using the following major elements: portType, message, parts, types, binding, port, and service [4]. For Web services dynamical invocation, WSDL parser is first needed to get service related information such as service name, description, operations, and the data type of the input arguments and the output arguments. WSDL4J has been used to parse the WSDL files by many Web services underlying technology implementations [5]. To obtain a complete Web service information needed to invoke the service, this technology is also used in our component. The parser reads the WSDL specification and extracts the operation and the message tags that are exposed

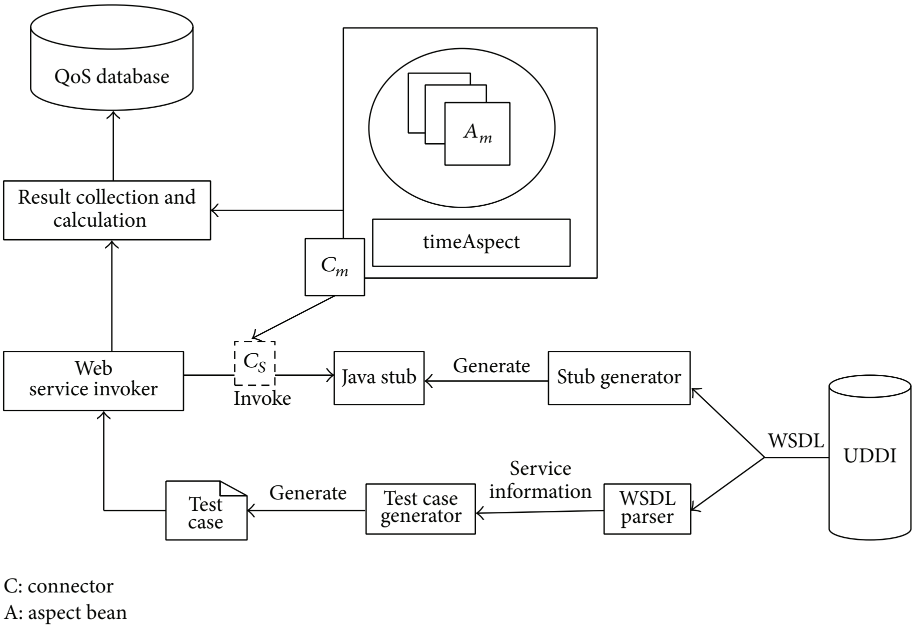


FIGURE 2: The architecture of the WS-QoS measurement tool.

by a particular Web service from the WSDL; in this way the methods and their input arguments' and return arguments' types of the service are acquired which will later on help the Web service invoker in invoking the required method of that service.

Then a test case knowledge base is established based on these pieces of information, as described in [6, 7], where each simple data type is associated with default facets definition and sets of candidate values based on the test strategies such as random values and boundary values. Complex data type defines a composition of simple and/or complex data types. To generate test data of complex data types, the generator recursively analyzes the structure of the data type until it reaches the simple type. The generated service information and test cases are documented in XML-based test files, which can be easily used by service invoker.

3.2. Web Service Stubs Generation Based on WSDL. Stubs are client side programs which act as a proxy for the server. Stubs are used to make calls to the Web services. Using stubs simplifies our applications considerably. We do not have to write a complex client program that dynamically generates SOAP requests and interprets SOAP responses. We can simply concentrate on writing the Web service invoking client code and leave all the other work to the stubs. The stubs are generally generated only once and then we can reuse the stubs as many times as we want. WSDL2Java from Axis is a tool that generates Java classes from an existing WSDL document. Generated classes represent a service and port combination with operations as methods. A data type class represents an input or output message part [8].

WSDL2Java generates the following stub and skeleton classes from existing WSDL documents: (1) the data type class that represents the input message part defined in the WSDL document; (2) the data type class that represents the output message part defined in the WSDL document; (3) the stub class that represents a combination of service and port defined in the WSDL document; (4) the default constructor method to create a stub object with information defined in the WSDL document; (5) the stub method that represents an operation defined in the WSDL document.

3.3. The Web Service Invoker. WSDL2Java analyzes WSDL file of Web service and creates the stub program and some interface programs. However we have to create the client program to execute the Web service by composing those stub and interface programs. Therefore, the Web service invoker is developed which tries to invoke a service operation just by "probing" arbitrary test values for the input parameters for an operation. Firstly the Web service invoker analyzes the Java code using "Class" and "Method" API in Java reflection and we can get the getter method and its return type. Secondly, the information from the test case file is acquired and some parameters required for the dynamic invocation of Web service are set in our system. Thirdly, Java reflection is used to dynamically instantiate these complex classes and Web service stubs. By dealing with the transactions described above, the Web service's operation is executed. Lastly, responses' results and status of the Web service are analyzed and collected by Result Collector and Calculation Component and stored in the QoS database. ome computation models of QoS properties are given in [9], which can be computed on the basis of the Web service invoker results and status.

3.4. The timeAspect. Response time is the time needed to process a query, from the moment of sending a request until receiving the response [9]. For measuring Web service response time, before sending the request, the current date and time are saved, and after receiving the response from Web service, the date and time are saved again. The response time of the Web method is calculated by subtracting the sending request time from the receiving response time. For keeping flexibility, this paper proposes using aspect-oriented programming technology to measure the response time of Web services. AOP approaches introduce a new concept to modularize crosscutting concerns, called an aspect. An aspect defines a set of join points in the target application where the normal execution is altered. Aspect weavers are used to weave the aspect logic into the target application [10]. The goal of AOP is to achieve a better separation of concerns, which makes it particularly suitable for solving the problems of Web services response time collection. This is because time record part is such a crosscutting concern since it spans over each service we have to invoke. Before the service invoking method starts execution, timeAspect points to the codes and records the start time; after the service invoking method is performed, timeAspect also weaves the codes and records the end time. The response time is equal to subtracting the start time from end time.

4. WS-QoS Reputation Evaluation

In order to provide better evaluation of the service provider's reputation, this section gives the computing model of the Web service's QoS similarity. The values of similarity can be used to represent the reputation level; higher values of similarity represent a better reputation level.

4.1. The Calculation of the Global QoS. For each call to the service by WS-QoS measurement tool, the collected Web services QoS attribute values may be different. In order to be able to reflect the real property of the dynamic changes of QoS, global QoS value of the service must be recomputed and stored based on the historical data and current data.

Given a Web service, it has m QoS attributes can be expressed as A: $\{A_j, 1 \le j \le m\}$. The Web service QoS attribute values set is defined as Q: $\{q_j, 1 \le j \le m\}$; q_j is the attribute value of the attribute A_j. The nth current factual QoS data collected by the WS-QoS measurement tool is defined by the set $\text{fa\_}Q_n$: $\{\text{fa\_}q_{nj}, 1 \le j \le m\}$; $\text{fa\_}q_{nj}$ is the nth actual value of the attribute A_j. Then the global QoS value of the service can be computed by using two-time average method. We randomly sample $p(p \le n)$ numeric from the n values $(p \le n)$; the lth QoS value $\text{fa\_}q_{lj}$ can be computed as in formula (1), so sample k times can get k sampling value, and then the global QoS value of the service $\text{fa\_}Q_g = \{\text{fa\_}q_j, 1 \le j \le m\}$,$\text{fa\_}q_j$ can be computed as in formula (2):

$$\text{fa\_}q_{lj} = \frac{1}{p}\sum_{i=1}^{p} \text{fa\_}q_{ij}, \quad p \neq 0, \tag{1}$$

$$\text{fa\_}q_j = \frac{1}{k}\sum_{l=1}^{k} \text{fa\_}q_{lj}, \quad k \neq 0. \tag{2}$$

4.2. The Calculation of the QoS Similarity. Similarity is acquired by calculating the accumulation average of the comparative result between the advertising quality values and the factual global quality values. QoS attributes hold two different directions or tendencies of their values; if the tendency of the attribute is positive, it means that a bigger value is better. On the contrary if the tendency is referred to as negative, it means that smaller values are preferred. For example, for attribute "response time" the smaller value is usually preferred, so the tendency of this parameter is negative, whereas for attribute "availability" the bigger value indicates a better quality for the specified parameter, so the tendency is positive. Based on the direction of the attribute, the similarity can be computed by the following formulas (3)–(5). As described in Section 4.1, the global QoS value of the service is $\text{fa\_}Q_g$ *and* $\text{fa\_}Q_g = \{\text{fa\_}q_j, 1 \le j \le m\}$; the advertised QoS which reflects the quality offered by the Web service provider is defined by the set $\text{ad\_}Q = \{(\text{ad\_min\_}q_j, \text{ad\_max\_}q_j), 1 \le j \le m\}$. $\text{ad\_min\_}q_j$ and $\text{ad\_max\_}q_j$ refer to the minimum value and the maximum value of the advertised quality attribute A_j, respectively. Consider

$$\text{sim} = \frac{\sum_{j=1}^{m} c}{m}. \tag{3}$$

If the tendency of attribute is negative,

$$c = \begin{cases} 0 & \text{if } \text{fa\_}q_j \le \text{ad\_max\_}q_j \\ 1 & \text{if } \text{fa\_}q_j > \text{ad\_ max \_}q_j. \end{cases} \tag{4}$$

If the tendency of attribute is positive,

$$c = \begin{cases} 0 & \text{if } \text{fa\_}q_j \le \text{ad\_ min \_}q_j \\ 1 & \text{if } \text{fa\_}q_j > \text{ad\_ min \_}q_j. \end{cases} \tag{5}$$

4.3. The Evaluation of the QoS Reputation Level. The QoS similarity obtained by using the aforementioned methods is between 0 and 1 ($0 \le \text{sim} \le 1$). The interval is divided into 5 stages, that is, $[0, 0.2]$, $[0.2, 0.4]$, $[0.4, 0.6]$, $[0.6, 0.8]$, and $[0.8, 1]$; each stage can correspond to a reputation level. According to the rank, the reputation level is ordered from low to high, respectively, that is, $1, 2, 3, 4, 5$, which represent reputation scores of the service. It is shown that if sim is higher, then the difference between the factual value and the advertised value of QoS is smaller and the reputation score is higher and vice versa.

5. Implementation and Experiment

We choose Java-based open source platforms and tools to implement the measurement tool. Axis provides better support to call Java and Java-based service, so we use the Axis to develop the client invoker part and deploy the simulation Web service in Axis. For parsing and analyzing the WSDL files we use the WSDL4J library from SourceForge [5]. The transformation from WSDL to Java classes is handled by the Axis WSDL2Java tool. timeAspect code is implemented with AspectJ.

To demonstrate the validity of our approach, the following three Web services are used as a sample in our experiment:

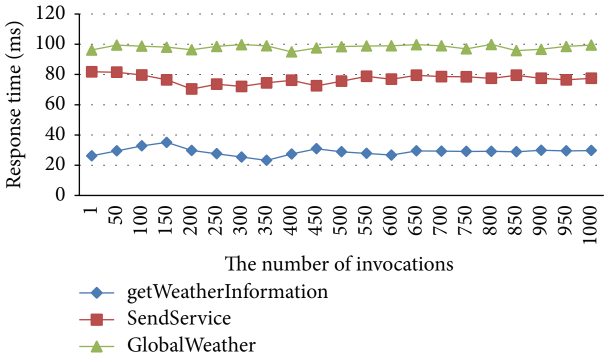


Figure 3: Global response time.

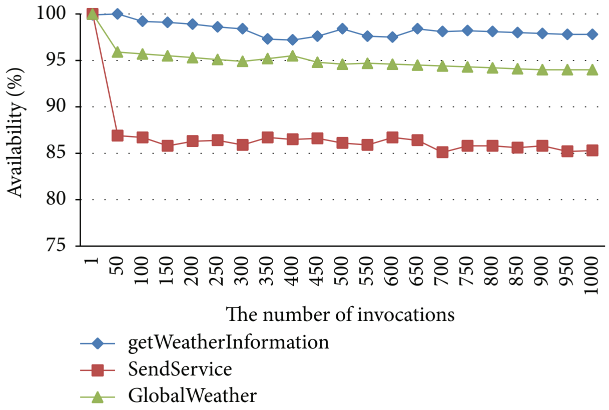


Figure 4: Global availability.

(1) getWeatherInformation: which allows you to get your city forecast over the next 7 days and is updated hourly; WSDL address is http://wsf.cdyne.com/WeatherWS/Weather.asmx?WSDL; (2) SendService: methods to send SMS messages individually and in batches; WSDL address is http://www.esendex.com/secure/messenger/soap/SendService.asmx?wsdl; (3) GlobalWeather: which gets country weather information; WSDL address is http://www.webservicex.com/globalweather.asmx?WSDL.

We invoked and monitored the mentioned services for 1000 times by using our tool, and then the factual QoS values such as response time, availability, and accessibility can be acquired. Figures 3-4 give changes of the global response time and availability with the difference of the invoking times. From the figure we can see that the global QoS values are more stable by using two-time average method with the increasing of the invoking times. In addition, Even if only 50 calls, we can still achieve good results. The experiment proved that our QoS measurement tool is useful and the calculation method of the global QoS is feasible; the global QoS value can fully represent the factual QoS value.

Table 1 gives the similarity between the factual values and the advertised values and shows the reputation level of the three Web services. From this table, we can see that not only our approach gives the reputation level objectively, but also similarity can be used as a rank of Web service, which can be useful to help service consumer select service.

6. Related Works

Artaiam and Senivongse [11] review a QoS model which covers various dimensions of service quality (i.e., availability, accessibility, performance, reliability, security, and regulatory) and propose metrics to enhance QoS measurement on the service side. A monitoring tool is designed and developed as an extension to Web services monitoring feature of Java system application server under Sun's Glass Fish project. Chen et al. [12] propose a novel trustable mechanism to monitor and evaluate SLA compliance based on the AOP paradigm. Authoritative monitoring aspects are supplied by a trustable SLA manager and by weaving the aspects into susceptible service runtime; service providers are ensured to monitor and report their service status obligatorily and accurately. In contrast, our WS-QoS measurement tool is a client tool and measures QoS values standing in the position of customers.

Michlmayr et al. [9] present a framework that combines the advantages of client and server side QoS monitoring. It builds on event processing to inform interested subscribers of current QoS values and possible violations of service level agreements. Rosenberg et al. [13] present an evaluation approach for QoS attributes of Web services, which works completely on service and provider independently; it also assesses performance of specific values (such as latency or service processing time) that usually require access to the server which hosts the service. Their approach is similar to ours, but they omit a way to specify how QoS test parameters' values are generated.

Nonintrusive monitoring [14–16] requires the establishment of mechanisms for capturing runtime information on service execution, for example, service operation calls and responses. In this way, monitoring logic is responsible for evaluation service QoS. This paper also employs aspect-oriented programming to ensure monitoring aspect codes separated from the service code. References [17–19] focus on the provision of a QoS monitoring architecture and measure QoS compliance in SOA infrastructures. Compared with our work, it is not specified how QoS attributes are actually measured.

Kalepu et al. [3] consider that the reputation of Web service consists of user ranking, compliance, and verity. They measure the consistency in service providers to deliver the QoS level specified in their contracts, which has been proposed as a metric to evaluate the reputation of Web services. According to the paper's proposal, we do an in-depth study and provide the concrete implementation. Fu et al. [2] design corresponding upper and lower QoS ontology for computing QoS consistency of factual value with advertised value automatically. The QoS consistency computing algorithm supports hierarchical QoS item consistency computing. Compared with our work, it is not specified how QoS values are actually measured. Nianhua et al. [20] propose a reputation evaluation algorithm for the new added Web service based on the similarity theory. Similarities and trust

Table 1: Similarity and reputation level.

QoS attribute/service name	getWeatherInformation	SendService	GlobalWeather
Adv_Response (ms)	(30, 40)	(40, 60)	(80, 95)
Fac_Response (ms)	38.9	76.9	98.1 ms
Adv_Availability	(98%, 100%)	(80%, 90%)	(90%, 100%)
Fac_Availability	97.2%	87.5%	94.8%
Adv_accessibility	(80%, 100%)	(85%, 90%)	(90%, 100%)
Fac_accessibility	88.6%	82.4%	93%
Similarity	1	0.33	0.67
Reputation level	5	2	4

are used as weights for computing reputations from different recommenders. Zhao et al. [21] propose a gradually adjusting reputation evaluation method of Web services based on eliminating the collusive behaviors of consumers step by step, and a reputation-aware model for service selection is designed. Unlike us, the reputation score is computed based on subjective judgment of service users but not objective measurement. Shao et al. [22] propose a similarity computing algorithm for Web services and their consumers based on Euclidean distance theory. Consumers' similarities are used as weights of indirect experiences. However, their similarity computing algorithm is different from us and mainly used in the QoS comparison between service providers and service consumers. Jøsang et al. [23] combine Bayesian reputation systems with a trust model for evaluating the quality of service in a single framework. Nepal et al. [24] propose a fuzzy trust evaluation approach for Web services. Both of them pay attention to propose a trust and reputation management framework for Web service selection.

7. Conclusions

This paper gives the factual QoS values by using our QoS measurement tool, compares the similarity of the factual QoS values and advertising QoS values, and completes the impartiality and objective Web service reputation evaluation. WS-QoS measurement tool is implemented by dynamically invoking the Web services and weaving aspects code into the Web service invoking code. Similarity is acquired by comparing the advertising quality values and the global quality values. According to the similarity, the reputation level is ordered from low to high. By a set of experiments, we prove the effectiveness and feasibility of the method. In the future, we will consider improving the QoS measurement tool, supporting more runtime data acquisition; furthermore, we plan to research on the updating algorithms for trust and reputations, making trustworthiness information reflect the latest changes in service.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors are pleased to acknowledge the support of the National Natural Science Foundation of China under Grant 61103115 and the Key Project of Guangdong and Hong Kong under Grant 2007A090601001. The authors are also grateful to the anonymous referees for their insightful and valuable comments and suggestions.

References

[1] K. Yue, X.-L. Wang, and A.-Y. Zhou, "Underlying techniques for web services: a survey," *Journal of Software*, vol. 15, no. 3, pp. 428–442, 2004.

[2] X. Fu, P. Zou, Y. Jiang, and Z. Shang, "QoS consistency as basis of reputation measurement of web service," in *Proceedings of the 1st International Symposium on Data, Privacy, and E-Commerce (ISDPE '07)*, pp. 391–396, IEEE, November 2007.

[3] S. Kalepu, S. Krishnaswamy, and S. W. Loke, "Reputation = f(user ranking, compliance, verity)," in *Proceedings of the IEEE International Conference on Web Services (ICWS '04)*, pp. 200–207, July 2004.

[4] Web Services Description Language[EB/OL], http://www.w3.org/TR/2001/NOTE-wsdl-20010315.

[5] Web Services Description Language for Java [EB/OL], http://sourceforge.net/projects/wsdl4j/files/WSDL4J.

[6] X. Bai, W. Dong, W.-T. Tsai, and Y. Chen, "WSDL-based automatic test case generation for Web Services testing," in *Proceedings of the IEEE International Workshop on Service-Oriented System Engineering (SOSE '05)*, pp. 207–212, October 2005.

[7] S. Hanna and M. Munro, "An approach for specification-based test case generation for Web services," in *Proceedings of the IEEE/ACS International Conference on Computer Systems and Applications (AICCSA '07)*, pp. 16–23, May 2007.

[8] WSDL2Java (Apache-Axis) [EB/OL], http://cxf.apache.org/docs/wsdl-to-java.html.

[9] A. Michlmayr, F. Rosenberg, P. Leitner, and S. Dustdar, "Comprehensive QoS monitoring of Web services and event-based SLA violation detection," in *Proceedings of the 4th Workshop on Middleware for Service Oriented Computing (MW4SOC '09)*, pp. 1–6, December 2009.

[10] Eclipse AspectJ [EB/OL], http://www.eclipse.org/aspectj/.

[11] N. Artaiam and T. Senivongse, "Enhancing service-side QoS monitoring for Web services," in *Proceedings of the 9th ACIS International Conference on Software Engineering, Artificial*

Intelligence, Networking and Parallel/Distributed Computing (SNPD '08), pp. 765–770, IEEE, August 2008.

[12] C. Chen, L. Li, and J. Wei, "AOP based trustable SLA compliance monitoring for web services," in *Proceedings of the 7th International Conference on Quality Software (QSIC '07)*, pp. 225–230, October 2007.

[13] F. Rosenberg, C. Platzer, and S. Dustdar, "Bootstrapping performance and dependability attributes of Web services," in *Proceedings of the IEEE International Conference on Web Services (ICWS '06)*, pp. 205–212, September 2006.

[14] H. Foster and G. Spanoudakis, "Advanced service monitoring configurations with SLA decomposition and selection," in *Proceedings of the 26th Annual ACM Symposium on Applied Computing (SAC '11)*, pp. 1582–1589, ACM, March 2011.

[15] R. Kazhamiakin, M. Pistore, and A. Zengin, "Cross-layer adaptation and monitoring of service-based applications," in *Service-Oriented Computing*, vol. 6275 of *Lecture Notes in Computer Science*, pp. 325–334, 2010.

[16] W. M. P. Van der Aalst, M. Dumas, C. Ouyang, A. Rozinat, and E. Verbeek, "Conformance checking of service behavior," *ACM Transactions on Internet Technology*, vol. 8, no. 3, article 13, 2008.

[17] A. Wahl, A. Al-Moayed, and B. Hollunder, "An architecture to measure QoS compliance in SOA infrastructures," in *Proceedings of the 2nd International Conferences on Advanced Service Computing*, pp. 27–33, ThinkMind, November 2010.

[18] M. Comuzzi, C. Kotsokalis, G. Spanoudakis, and R. Yahyapour, "Establishing and monitoring slas in complex service based systems," in *Proceedings of the IEEE International Conference on Web Services (ICWS '09)*, pp. 783–790, July 2009.

[19] C. Muller, M. Oriol, M. Rodriguez, X. Franch, J. Marco, and A. Ruiz-Cortes, "SALMonADA: a platform for monitoring and explaining violations of WS-agreement-compliant documents," in *Proceedings of the ICSE Workshop on Principles of Engineering Service Oriented Systems (PESOS '12)*, pp. 43–49, IEEE, June 2012.

[20] Y. Nianhua, C. Xin, and Y. Huiqun, "A reputation evaluation technique for web services," *International Journal of Security and Its Applications*, vol. 6, no. 2, pp. 329–334, 2012.

[21] S. Zhao, G. Wu, G. Chen, and H. Chen, "Reputation-aware service selection based on QOS similarity," *Journal of Networks*, vol. 6, no. 7, pp. 950–957, 2011.

[22] L.-S. Shao, L. Zhou, J.-F. Zhao, B. Xie, and H. Mei, "Web service QoS prediction approach," *Journal of Software*, vol. 20, no. 8, pp. 2062–2073, 2009.

[23] A. Jøsang, T. Bhuiyan, Y. Xu, and C. Cox, "Combining trust and reputation management for Web-based services," in *Trust, Privacy and Security in Digital Business*, vol. 5185 of *Lecture Notes in Computer Science*, pp. 90–99, 2008.

[24] S. Nepal, W. Sherchan, J. Hunklinger, and A. Bouguettaya, "A fuzzy trust management framework for Service Web," in *Proceedings of the 8th IEEE International Conference on Web Services (ICWS '10)*, pp. 321–328, July 2010.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 459636, 6 pages
http://dx.doi.org/10.1155/2014/459636

Research Article

Classifying Normal and Abnormal Status Based on Video Recordings of Epileptic Patients

Jing Li,[1] Xiantong Zhen,[2] Xianzeng Liu,[3] and Gaoxiang Ouyang[4,5]

[1] *Department of Electrical and Automatic Engineering, School of Information Engineering, Nanchang University, Nanchang 330031, China*
[2] *Department of Medical Biophysics, University of Western Ontario, Room E5-137, SJHC, 268 Grosvenor Street, London, ON, Canada N6A 4V2*
[3] *The Comprehensive Epilepsy Center, Departments of Neurology and Neurosurgery, Peking University People's Hospital, Beijing 100044, China*
[4] *State Key Laboratory of Cognitive Neuroscience and Learning & IDG/McGovern Institute for Brain Research, Beijing Normal University, Beijing 100875, China*
[5] *Center for Collaboration and Innovation in Brain and Learning Sciences, Beijing Normal University, Beijing 100875, China*

Correspondence should be addressed to Xianzeng Liu; xianzeng.liu@gmail.com and Gaoxiang Ouyang; ouyang@bnu.edu.cn

Received 27 February 2014; Accepted 27 March 2014; Published 8 April 2014

Academic Editors: Y.-B. Yuan and S. Zhao

Based on video recordings of the movement of the patients with epilepsy, this paper proposed a human action recognition scheme to detect distinct motion patterns and to distinguish the normal status from the abnormal status of epileptic patients. The scheme first extracts local features and holistic features, which are complementary to each other. Afterwards, a support vector machine is applied to classification. Based on the experimental results, this scheme obtains a satisfactory classification result and provides a fundamental analysis towards the human-robot interaction with socially assistive robots in caring the patients with epilepsy (or other patients with brain disorders) in order to protect them from injury.

1. Introduction

Epilepsy, one of the most common neurologic disorders, is a chronic disease of brain sudden paradoxical discharge of cortical neurons. It is characterized by the spontaneous and unforeseeable occurrence of seizures [1] with transient signs and/or symptoms due to abnormal, excessive, or synchronous neuronal activity in the brain [2]. It is often accompanied with disturbances in behaviour, short-term brain dysfunction, and cognitive impairment. According to the World Health Organization, the incidence of epilepsy has affected more than 50 million individuals worldwide—about 0.6–1% of the world's population. Because patients with epilepsy have poor ability of independent living, they have lower rates of employment and marriage than others. This not only affects the patients themselves, but also causes fear and inconvenience to their family.

Since the first robot was created in the 1960s, robots have been increasingly used in industrial and entertainment, and more recently the research on socially assistive robots (SARs) for domestic use has received much attention. A socially assistive robot is an intelligent system that is capable of providing assistance for healthy adults or enhancing existing care for persons with cognitive disabilities/impairment, for example, stroke, Alzheimer, and autism spectrum disorder. The question then arises: can socially assistive robots replace nurses or patients' family members for monitoring and caring the epilepsy patients, that is, sounding the alarm, or taking some effective actions to alleviate seizure symptoms?

To address this, human-robot interaction (HRI) [3, 4] is an important issue. Based on the information collected from multiple sensors, HRI works toward smooth interactions between a human user and a socially assistive robot via the use of speech, vision, haptic control, and so forth, for implementing a task. It is a broad area, including a wide variety of research topics, that is, robotics, computer vision, human-computer interaction, modern artificial intelligent, natural language processing, and cognitive science. In the viewpoint of computer vision, the aim of HRI is to understand the patients' behaviors. The first thing is that the robot needs to analyze their motions through action recognition to determine whether the patients are at the seizure status. Otherwise, it may generate unnecessary interaction between the robots and the patients with epilepsy.

According to the International League Against Epilepsy (ILAE), seizure types are organized according to whether the source of the seizure within the brain is localized (partial or focal onset seizures) or distributed (generalized seizures) [5]. Partial seizures, having a focal origin, are further divided based on the extent to which consciousness is affected (simple partial seizures and complex partial seizures) [6]. Generalized seizures affect both cerebral hemispheres (sides of the brain) from the beginning of seizures, such as absence seizures [7] which are short in duration (typically lasting from a few seconds up to around a minute) and may recur over 100 times a day [8]. The seizures may bring the patients with sudden accidents and injuries. Moreover, the sudden and abrupt seizures can appear at any age and may cause serious problems to the patients' body, mind, and intelligence with long-term repeated seizure onset. This supports the importance of detecting seizures as early as possible such that clinicians can prescribe necessary medication for the patients to stop the progression of the chronic disease.

Initially, epilepsy is diagnosed by experienced experts via observing patients' actions, behavioral changes, and mental health history in the family. However, it is not practical in clinical use due to high cost of the manpower and financial resources. During the past few decades, it was confirmed that electroencephalogram (EEG) signals, recording the spontaneous brain electrical activity by means of electrodes located on the scalp, can provide evidence for the existence of a preseizure phase in partial epilepsy [9, 10]. In this paper, we focus on the research of epilepsy and investigate whether the detection/classification of seizure status of patients with epilepsy can be explored by analyzing the movement of epileptics at the video level through human action recognition. As the absence or anomalies of such movement is a highly predictive indicator for epilepsy, accurate classification about the patients' status through video recordings is a fundamental step in detecting different seizure status in the epilepsy. Here, we use computer vision-based techniques to extract the information of movement from video recordings of patients.

Human action recognition [11, 12] is one of the most active topics in computer vision and has been widely applied in video surveillance, video annotation, and retrieval. Current action recognition systems are mainly based on local and holistic representations. Local representations [12] sparsely detect spatiotemporal interest points (STIPs) and have dominated in human action recognition due to their attractive advantages, such as being less sensitive to partial occlusions and clutter and requiring no background subtraction or target tracking as in holistic representations. Nevertheless, local methods suffer from some limitations, one of which is the inability to capture adequate spatial and temporal structure information of actions. On the other hand, holistic representations [13] directly extract spatiotemporal features from raw video sequences and are able to provide entire spatial and temporal structural information of human actions in a sequence. However, they are highly sensitive to partial occlusions and background variations and often require computationally expensive preprocessing steps such as background subtraction, segmentation, and tracking.

To this end, in the task of recognizing normal and abnormal status of epileptics, we propose a simple classification scheme on video recordings by combining local representation with holistic representation, which is able to deal with their shortcomings while integrating their merits. For local representation, we use 3D Gabor filters [14, 15] as they are biologically relevant to human image understanding and recognition. Afterwards, holistic representation is obtained by applying gist features [16] over each filtered volume. Finally, the classification is implemented by a support vector machine (SVM) [17, 18].

To determine whether an epileptic is at the seizure status, the scheme is implemented on the video recordings of epileptic patients at different ages. The movement of affected patients is characterized by more abrupt motion direction changes with periods of no movement. Moreover, the classification task entails lots of challenges: (1) different types of actions: epilepsy appears in different actions, for example, abruptly falling down and continuously vibrating; (2) multiple persons: some video recordings not only contain the patient himself/herself, but also contain the family members or nurses; (3) turning on and turning off the lights in the ward lead to different lighting conditions; and (4) the persons in the videos sometimes wear different-color clothes. Due to the above-mentioned difficulties, the literature that addresses the epilepsy area using video data is limited. Although we only obtain primary results, it is the first time that only video recordings are analyzed for the classification problem in epilepsy, aiming at shedding the light for future research on the prediction of seizure onset. The rest of the paper is organized as follows. In Section 2, we present the newly proposed scheme in detail, including feature extraction based on 3D Gabor features and gist features. In Section 3, we describe the video dataset applied in evaluating the performance of the scheme and report the experimental results. Section 4 concludes the paper and points out some future works.

2. Methods

The proposed scheme (as shown in Figure 1) consists of the following two main steps: (i) feature extraction by 3D Gabor features and gist features and (ii) classification by SVM. Each step will be described in detail in the next subsections.

2.1. 3D Gabor Filters. Research findings from cognitive psychology and psychophysics suggest that Gabor filters [14, 15] based on image decomposition are biologically relevant to

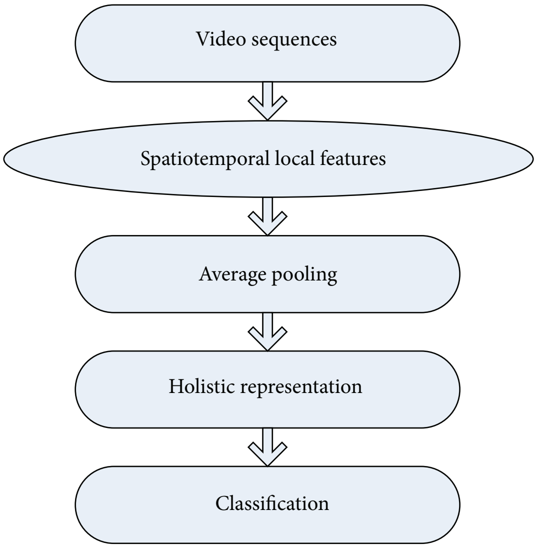


Figure 1: The proposed scheme.

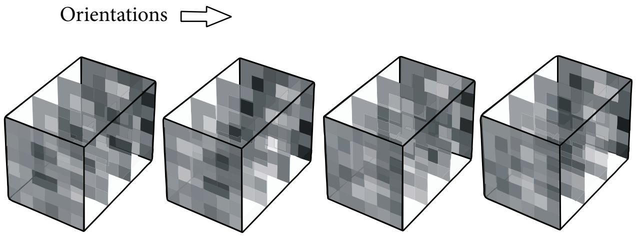


Figure 2: 3D Gabor filters on intensity volume.

human image understanding and recognition. Consequently, Gabor filters are appropriate for orientation information extraction within a purely computer vision context.

Here, we apply a bank of 3D Gabor filters with one scale and four orientations to localizing salient features in spatiotemporal dimensions, making a total of four Gabor functions. In a 3D space, Gabor filters are defined as

$$G(x,y,t) = \exp\left(-\left(\frac{X^2}{2\sigma_x}+\frac{Y^2}{2\sigma_y}+\frac{T^2}{2\sigma_t}\right)\right) \times \cos\left(\frac{2\pi}{\lambda_x}X\right)\cos\left(\frac{2\pi}{\lambda_y}Y\right) \tag{1}$$

with

$$\begin{pmatrix} X \\ Y \\ T \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{pmatrix} \times \begin{pmatrix} \cos(\omega) & 0 & \sin(\omega) \\ 0 & 1 & 0 \\ -\sin(\omega) & 0 & \cos(\omega) \end{pmatrix} \begin{pmatrix} x \\ y \\ t \end{pmatrix}, \tag{2}$$

where θ and ω are the spatial and temporal orientations. Figure 2 shows the 3D Gabor filters on intensity volume in horizontal direction. Since Gabor filters are differential algorithms, the extracted visual features are robust to illumination changes.

2.2. Spatiotemporal Gist Feature Extraction. To make compact representation and achieve invariance to small shifts in position and changes in lighting conditions, we use average pooling [19, 20] over the filtered volumes to extract gist features. Similar to the gist feature extraction in scene recognition [16], a gist feature is generated from each filter volume by dividing the volume into a $4 \times 4 \times 4$ grid and then averaging the responses of pixels within each spatiotemporal subregion, resulting in a 256-dimensional feature vector. In this way, the extracted gist features can preserve discriminative information and are tolerant to spatial and temporal shifts and insensitive to noise.

2.3. Classification. A support vector machine (SVM) [17, 18] is a binary classifier, which maximizes the margin between positive examples and negative examples, as shown in Figure 3. Because of its good generalization ability and no requirement for prior knowledge about the data, it has been universally utilized as one of the most popular classifiers in various research areas, for example, face recognition, texture classification, content-based image retrieval (CBIR), and so forth.

Hard-margin SVM and soft-margin SVM are two different forms of an SVM. On one hand, hard-margin SVM solves a quadratic programming problem to deal with linearly separable data. It is effective and requires no parameters. However, it cannot deal with linearly nonseparable examples. On the other hand, soft-margin SVM, the standard solution of a SVM, allows some misclassifications or outliers by adding a regularization term to handle linearly nonseparable data. The methodology of soft-margin SVM is reviewed as follows.

Consider a problem of classifying a set of linearly separable training examples $\{(\vec{x}_i, y_i)\}_{i=1}^{N}$ with $\vec{x}_i \in \Re^L$ and their associated class labels $y_i \in \{+1, -1\}$, an SVM separates these two classes by a hyperplane

$$\vec{w}^T \cdot \vec{x} + b = 0, \tag{3}$$

where $\vec{x}$ is an input vector, $\vec{w}$ is an adaptive weight vector, and the scalar b is a bias. The optimal hyperplane, which maximizes the geometric margin $2/\|\vec{w}\|$ between two classes, can be obtained by

$$\begin{aligned} &\min_{\vec{w},b,\vec{\xi}} \quad \frac{\|\vec{w}\|^2}{2} + C\sum_{i=1}^{N}\xi_i \\ &\text{s.t.} \quad y_i\left(\vec{w}^T \cdot \vec{x}_i + b\right) \geq 1 - \xi_i, \\ &\qquad 1 \leq i \leq N \quad \vec{\xi} \geq 0, \end{aligned} \tag{4}$$

where C is a constant determined by cross-validation and $\vec{\xi} = [\vec{\xi}_1, \vec{\xi}_2, \ldots, \vec{\xi}_N]^T$ is the vector of all slack variables to deal with the linearly nonseparable problem by giving each misclassified example an individual penalty. For linearly

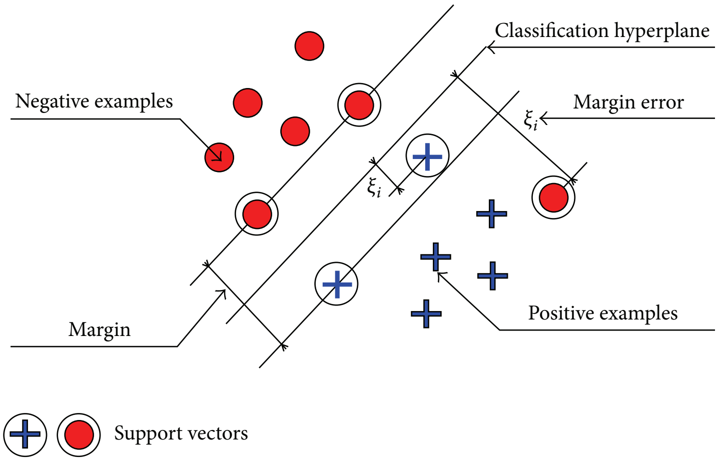


FIGURE 3: An SVM maximizes the margin between positive examples and negative examples.

separable training examples, we can set $\vec{\xi} = 0$. By introducing a Lagrange multiplier α_i, the Lagrangian is

$$L\left(\vec{w}, b, \vec{\xi}, \vec{\alpha}, \vec{\kappa}\right) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i\left(y_i\left(\vec{w}^T \cdot \vec{x}_i + b\right) - 1 + \xi_i\right) - \sum_{i=1}^{N}\kappa_i\xi_i, \tag{5}$$

and the solution is determined by

$$\max_{\vec{\alpha},\vec{\kappa}} \min_{\vec{w},b,\vec{\xi}} L\left(\vec{w}, b, \alpha\right), \tag{6}$$

which can be achieved by the Karush-Kuhn-Tucker (KKT) conditions

$$\begin{gathered}\frac{\partial L}{\partial \vec{w}} = 0 \Longrightarrow \vec{w} = \sum_{i=1}^{N}\alpha_i y_i \vec{x}_i, \\ \frac{\partial L}{\partial b} = 0 \Longrightarrow \vec{\alpha}^T\vec{y} = 0, \\ \frac{\partial L}{\partial \xi} = 0 \Longrightarrow C - \vec{\alpha} - \vec{\kappa} = 0.\end{gathered} \tag{7}$$

Therefore, the parameters $\vec{w}$ and b can be obtained using the Wolfe dual problem

$$\begin{aligned}\max_{\vec{\alpha}} \quad & Q\left(\alpha\right) = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j\left(\vec{x}_i^T \cdot \vec{x}_j\right) \\ \text{s.t.} \quad & 0 \le \alpha_i \le C \\ & \vec{\alpha}^T\vec{y} = 0.\end{aligned} \tag{8}$$

Most of α_i are zeros, and $\vec{x}_i$ corresponding to $\alpha_i > 0$ are referred to the support vectors. In Figure 3, they are expressed as the examples close to the decision boundary or at the wrong side of the margin.

In the dual format, data points only appear in the inner product. To solve the nonlinearly separable problem, the data points from the low-dimensional input space L are mapped onto a higher dimensional feature space H (the Hilbert inner product space) by the replacement

$$\vec{x}_i \cdot \vec{x}_j \longrightarrow \phi\left(\vec{x}_i\right) \cdot \phi\left(\vec{x}_j\right) = K\left(\vec{x}_i, \vec{x}_j\right), \tag{9}$$

where $K(\vec{x}_i, \vec{x}_j)$ is a kernel function with entries $\phi(\vec{x}_i), \phi(\vec{x}_j) \in \mathfrak{R}^H$. A lot of standard kernel functions can be embedded in SVMs, such as linear kernels $K(\vec{x}_i, \vec{x}_j) = \vec{x}_i^T \cdot \vec{x}_j$, polynomial kernels $K(\vec{x}_i, \vec{x}_j) = (\vec{x}_i \cdot \vec{x}_j + c)^d$, Gaussian radial basis function (RBF) $K(\vec{x}_i, \vec{x}_j) = \exp\{-\|\vec{x}_i - \vec{x}_j\|^2/2\sigma^2\}$. Then, the kernel version of the Wolfe dual problem is

$$Q\left(\alpha\right) = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j K\left(\vec{x}_i \cdot \vec{x}_j\right). \tag{10}$$

Finally, for a given kernel function, the SVM classifier is given as

$$F\left(\vec{x}\right) = \operatorname{sgn}\left(f\left(\vec{x}\right)\right), \tag{11}$$

where $f(\vec{x}) = \sum_{i=1}^{l}\alpha_i y_i K(\vec{x}_i, \vec{x}_j) + b$ is the output hyperplane decision function of SVM.

In traditional SVM-based RF algorithms, $f(\vec{x})$ is used for measuring the dissimilarity between the query image and an example image in the database. For a given example, a high $f(\vec{x})$ indicates that it is far away from the decision boundary and thus has high prediction confidence while a low $f(\vec{x})$ shows that it is close to the boundary and its corresponding prediction confidence is low.

3. Results and Discussion

In this section, we describe the process of video data acquisition and experimental setup. Afterwards, the experimental results that evaluate the effectiveness of our proposed scheme are reported.

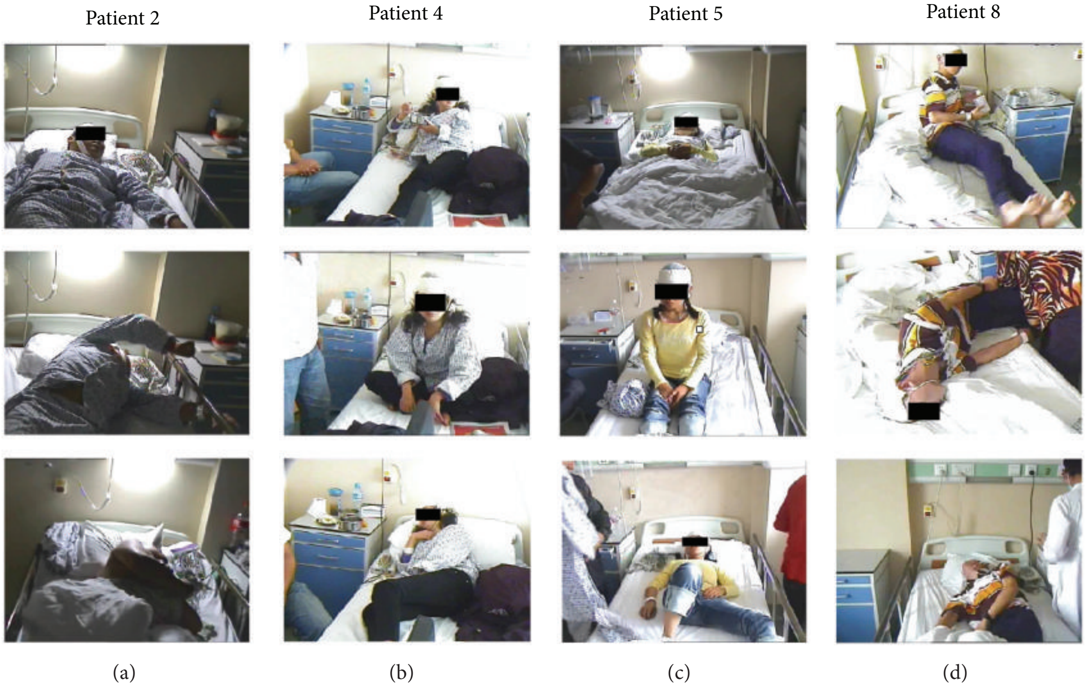


Figure 4: Sample video frames from Patient 2, Patient 4, Patient 5, and Patient 8, respectively.

3.1. Video Data Acquisition and Experimental Setup. We collected 41 video recordings of 9 epilepsy patients at a resolution of 640 × 480. The videos were recorded with a frame rate of 25 frames/s in the AVI video format. Each patient was lying on a standard hospital bed, and a stationary digital video camera was placed at a distance above the patients to record the movements. This resulted in an experimental setup where a similar camera position was assured for all recordings. Please note that not every video clip recorded the seizure status; that is, some videos were selected to ensure an awake and comfortable state. The original 41 video recordings are of different length, ranging from less than one minute to nearly ten minutes. Considering this, we segment each video into several ones that are no longer than one minute. Afterwards, each of them is converted into the AVI video format at a resolution of 160 × 128. The study protocol had previously been approved by the ethics committee of Peking University People's Hospital and the patients had signed informed consent that their clinical data might be used and published for research purposes.

Figure 4 shows sample video frames from Patient 2, Patient 4, Patient 5, and Patient 8, respectively, where frames in a column belong to the same patient. As we can see, with this dataset, the classification task entails lots of challenges: (1) different types of actions: epilepsy appears in different actions, for example, abruptly falling down, continuously shaking, and so forth; (2) multiple persons: some video recordings not only contain the patient himself/herself, but also contain the family members or nurses; (3) the patients are usually in different lighting conditions; and (4) the persons in the videos sometimes wear different-color clothes. All of the four points mentioned above bring about some difficulty in distinguishing normal status from abnormal status.

To evaluate the classification performance, we manually annotated each video recording with a bounding box to locate the epilepsy child. Moreover, a label, that is, "normal" or "abnormal," is assigned to each video.

For classification, we make use of an SVM classifier with a linear kernel due to its good generalization ability and efficiency. Classification consists of the training phase and the testing phase. We randomly select half of the video clips for each patient for training and use the rest for testing. This step is conducted for 5 times and the average accuracy is 65.22%. After training the classifier based on the extracted features of the training set, the classifier is trained and able for classifying the examples in the testing set. We define two categories labeled as "normal" and as "abnormal," which denote different states of the tested epileptics.

Although the accuracy is not very high, but with such a complex video dataset, the classification performance is satisfactory and can serve as a tool for automatically predicting the seizure status of the epilepsy patients. The current video data include epilepsy patients of different ages. An interesting question is how the movement changes with an increasing age of the patients and whether this can be used as a feature value.

4. Conclusions

This paper explores whether the normal status and the abnormal status of epileptic patients can be distinguished based on video recordings rather than traditional EEG recordings. Combining local representation and holistic representation, the extracted features are effective for the subsequent classification by an SVM. Our future goal is to capture the characteristic abrupt movements of epileptics by developing new features that are effective in detecting abnormal actions of the epileptic patients.

As pointed in [21], higher frame rates of video recordings could increase the accuracy of the motion estimation and result in a high quality of motion tracking. Another promising option that can be explored in the future would be to collect Kinect videos using RGB-D camera. In this way, depth information of 3D points can be included for motion tracking. What is more, we would collect more video data and classify the actions into different types in our future work. Last, we intend to extend this work into the research of other medical areas, such as autism and Alzheimer.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (61105027 and 81341042), Beijing Natural Science Foundation (4143063), Peking University People's Hospital Research and Development Funds (RDB2010-16), and the Fundamental Research Funds for the Central Universities.

References

[1] B. Litt and J. Echauz, "Prediction of epileptic seizures," *Lancet Neurology*, vol. 1, no. 1, pp. 22–30, 2002.

[2] R. S. Fisher, W. van Emde Boas, W. Blume et al., "Epileptic seizures and epilepsy: definitions proposed by the International League Against Epilepsy (ILAE) and the International Bureau for Epilepsy (IBE)," *Epilepsia*, vol. 46, no. 4, pp. 470–472, 2005.

[3] H. W. Park and A. M. Howard, "Understanding a child's play for robot interaction by sequencing play primitives using hidden markov models," in *Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA '10)*, M. Rakotondrabe and I. A. Ivan, Eds., pp. 170–177, May 2010.

[4] S. Thrun, "Toward a framework for human-robot interaction," *Human-Computer Interaction*, vol. 19, no. 1-2, pp. 9–24, 2004.

[5] J. Engel Jr., "A proposed diagnostic scheme for people with epileptic seizures and with epilepsy: report of the ILAE task force on classification and terminology," *Epilepsia*, vol. 42, no. 6, pp. 796–803, 2001.

[6] W. S. Anderson, F. Azhar, P. Kudela, G. K. Bergey, and P. J. Franaszczuk, "Epileptic seizures from abnormal networks: why some seizures defy predictability," *Epilepsy Research*, vol. 99, no. 3, pp. 202–213, 2012.

[7] H. K. M. Meeren, J. P. M. Pijn, E. L. J. M. van Luijtelaar, A. M. L. Coenen, and F. H. L. da Silva, "Cortical focus drives widespread corticothalamic networks during spontaneous absence seizures in rats," *Journal of Neuroscience*, vol. 22, no. 4, pp. 1480–1495, 2002.

[8] X. Bai, M. Vestal, R. Berman et al., "Dynamic time course of typical childhood absence seizures: EEG, behavior, and functional magnetic resonance imaging," *Journal of Neuroscience*, vol. 30, no. 17, pp. 5884–5893, 2010.

[9] F. Mormann, R. G. Andrzejak, C. E. Elger, and K. Lehnertz, "Seizure prediction: the long and winding road," *Brain*, vol. 130, no. 2, pp. 314–333, 2007.

[10] F. Mormann, C. E. Elger, and K. Lehnertz, "Seizure anticipation: from algorithms to clinical practice," *Current Opinion in Neurology*, vol. 19, no. 2, pp. 187–193, 2006.

[11] S. Sadanand and J. J. Corso, "Action bank: a high-level representation of activity in video," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 1234–1241, June 2012.

[12] L. Yeffet and L. Wolf, "Local trinary patterns for human action recognition," in *Proceedings of the 12th International Conference on Computer Vision (ICCV '09)*, pp. 492–497, October 2009.

[13] A. Oliva and A. Torralba, "Modeling the shape of the scene: a holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–175, 2001.

[14] J. G. Daugman, "Two-dimensional spectral analysis of cortical receptive field profiles," *Vision Research*, vol. 20, no. 10, pp. 847–856, 1980.

[15] J. G. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *Journal of the Optical Society of America A: Optics and Image Science*, vol. 2, no. 7, pp. 1160–1169, 1985.

[16] C. Siagian and L. Itti, "Rapid biologically-inspired scene classification using features shared with visual attention," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 300–312, 2007.

[17] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, 1995.

[18] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[19] Y.-L. Boureau, J. Ponce, and Y. Lecun, "A theoretical analysis of feature pooling in visual recognition," in *Proceedings of the 27th International Conference on Machine Learning (ICML '10)*, pp. 111–118, Haifa, Israel, June 2010.

[20] A. M. Treisman and G. Gelade, "A feature-integration theory of attention," *Cognitive Psychology*, vol. 12, no. 1, pp. 97–136, 1980.

[21] A. Stahl, C. Schellewald, O. Stavdahl, O. M. Aamo, L. Adde, and H. Kirkerod, "An optical flow-based method to predict infantile cerebral palsy," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 20, pp. 605–614, 2012.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 723595, 9 pages
http://dx.doi.org/10.1155/2014/723595

Research Article

Iris Recognition Using Image Moments and k-Means Algorithm

Yaser Daanial Khan,[1,2] Sher Afzal Khan,[2] Farooq Ahmad,[3] and Saeed Islam[4]

[1] *School of Science and Technology, University of Management and Technology, Lahore 54000, Pakistan*
[2] *Department of Computer Science, AbdulWali Khan University, Mardan 23200, Pakistan*
[3] *Faculty of Information Technology, University of Central Punjab, 1-Khayaban-e-Jinnah Road, Johar Town, Lahore 54000, Pakistan*
[4] *Department of Mathematics, AbdulWali Khan University, Mardan 23200, Pakistan*

Correspondence should be addressed to Yaser Daanial Khan; ydk@ucp.edu.pk

Received 11 January 2014; Accepted 19 February 2014; Published 1 April 2014

Academic Editors: Y.-B. Yuan and S. Zhao

This paper presents a biometric technique for identification of a person using the iris image. The iris is first segmented from the acquired image of an eye using an edge detection algorithm. The disk shaped area of the iris is transformed into a rectangular form. Described moments are extracted from the grayscale image which yields a feature vector containing scale, rotation, and translation invariant moments. Images are clustered using the k-means algorithm and centroids for each cluster are computed. An arbitrary image is assumed to belong to the cluster whose centroid is the nearest to the feature vector in terms of Euclidean distance computed. The described model exhibits an accuracy of 98.5%.

1. Introduction

Identification of individuals has been an important need over the ages. Conventionally identification documents like an identity card, passport, or driving license have been utilized for the purpose. Such identification methods have been evaded several times by use of forged documents. In the digital world a login and a password or a PIN code is used for identification. Besides shoulder surfing and sniffing several other techniques have evolved which are used to crack such codes and breach security. Undoubtedly a robust identification technique is essential for a safe and well supervised environment. This situation thrives the need of an identification technique using some biological inimitable features of a person. Numerous biological human features are peculiar and unique such as fingerprints, suture patterns, iris patterns, gait, and ear shapes. The patterns found in these structures are unique for every human; hence they can be used as an identification tool. In the recent past, use of iris image of a person for his identification has gained popularity. The radial and longitudinal muscles in the iris of an eye are responsible for the constrictions and dilation of the pupil. The pupil changes its size depending upon the light intensity the eye is exposed to. The muscles of iris form the texture of the iris while the presence or absence of a pigment forms the color of the iris. The color of the iris is genetically dependent, whereas the texture is not. The texture of iris forms random unique patterns for each human. A close observation of an iris may reveal pustules, rings, stripes, and undulations forming a unique pattern.

In the recent past, researchers have developed several mechanisms for matching the pattern that lies within the iris. In [1] the author employs a bank of Gabor filters to form a fixed length vector from the local and global iris characteristics. Iris matching is established based on the weighted Euclidean distance between the two iris images being compared. In another article by Monro et al., a technique is devised using discrete cosine transform. Iris coding is based on the differences of discrete cosine transform coefficients of overlapped angular patches from normalized iris images [2]. Certain researchers have employed various statistical models for the purpose. A nonparametric statistical model, namely, neural networks (NN), is used for pattern matching and data compression in [3–5]. The image processing technique using specially designed kernels for iris recognition is used to capture local characteristics so as to produce discriminating texture features in [6]. Several sorts of transformations also prove helpful in extracting useful features from an iris image. This feature vector is further used to form a classification approach for identifying a person based on his iris image [7, 8]. In the groundbreaking work by Daugman the iris recognition principle is based on the failure of statistical independence tests on iris phase structure encoded by multiscale quadrature wavelets. The combinatorial complexity of this phase information across different persons generates discriminating entropy enabling the most probable decision about a person's identity [9].

Most of the techniques based on feature extraction are designed for image of a certain fixed resolution. They fail to provide the desired result for the same images with different resolution. This characteristic implies that the model is not scale invariant. Techniques making use of NN incorporate a time taking training procedure. At times this training process may prove to be tricky rendering the model unable to yield quick results. On the other hand, some techniques that make use of certain filters may produce undesired results if the image is rotated which implies that such models are not rotation invariant. In this underlying paper a scale and rotation invariant technique for the same purpose is described. The proposed technique requires little training after which results are produced instantly. It is based on the use of image moments. Moments are properties that describe the characteristics of a certain distribution of data. Image moments (namely, Hu moments) are a quantitative measure of the shape of distribution formed by data collected as image pixel intensities and their locations [10].

In the proposed work the iris is segmented from an eye image. Image moments are computed from the segmented grayscale image. Classification of an iris is performed by the k-means algorithm. The composition of the paper is as described here. Section 2 gives an overview of iris recognition process. Section 3 explains the method used for iris segmentation. Section 4 gives a method for transforming the radial information of the iris into a rectangular form. Section 5 explains how this method can be further optimized. Image moments and method of computation of moments are described in Section 6. Section 7 describes the adoption of k-means algorithm for clustering and classification using moments information. Some of the results are discussed in Sections 8 and 9 presents some conclusions.

2. Iris Recognition

Initially the image of an eye is acquired by a device called iriscope specifically designed for eye image acquisition at a high resolution. A large database of such images is collected having several classes. The iris within the image is segmented using an accurate and sufficiently fast technique. The iris image is of radial nature, rather than rectangular, which makes it unsuitable to be processed by any mathematical or statistical model of linear nature. There are two approaches to resolve this problem. The first approach is to adapt a model capable of processing data in its inherent radial form. Other approaches require transformation of the radial data into multidimensional linear form such that the information pertaining to iris texture is retained. In this piece of work the latter approach is adopted.

The information within the texture of the rectangular image may be used to form a probability density function. The image moments quantify the characteristics of this distribution. Using these raw moments translation, scale and rotation invariant moments are computed. Accumulated, these moments describe the characteristics of the pattern of the iris. This forms a feature vector which is later used for classification of iris images.

3. Iris Segmentation

Each image in the database contains the iris pattern which is of interest; the rest of the image is of no use and therefore is not processed. The iris is extracted from the image using the segmentation process described in [5]. The iris is modeled as a disk-like structure consisting of two concentric circles (see Figure 2). The noise in the eye image is suppressed using numerous iterations of median filtering [11]. The image with reduced noise is filtered to extract edges using an edge detection algorithm like the Canny [12] or the Sobel [13] filter as shown in Figure 1(a). Now using the resultant image the iris outline is extracted. The image is scanned top to bottom, left to right line by line. Each point on the outer and the inner edge is stored in two separate arrays. These points are further used to determine the center and the radii of the concentric circles forming the iris and the pupil as shown in Figure 1(b). Assuming that the outline of the iris is a circle, a point (x, y) on the circle with the center at $(-g, -f)$ satisfies the equation

$$x^2 + y^2 + 2gx + 2fx + c = 0. \tag{1}$$

And the radius of the circle is given as

$$r = \sqrt{g^2 + f^2 - c}. \tag{2}$$

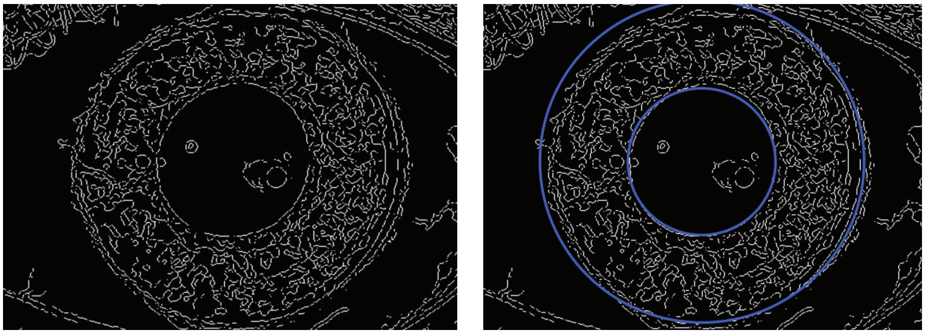
Choosing any three arbitrary points from the array containing points of the circle, a system of simultaneous equations is formed. The solution to c, f, and g in terms of the selected three points are derived from the system and is given as

$$\begin{aligned} g &= \left(x_1^2 - x_3^2 + y_1^2 - y_3^2\right)\left(y_2 - y_1\right) \\ &\quad - \left(x_1^2 - x_2^2 + y_1^2 - y_2^2\right)\left(y_3 - y_1\right)\Big) \\ &\quad \times \left(2\left[\left(x_3 - x_1\right)\left(y_2 - y_1\right)\right.\right. \\ &\quad \left.\left. + \left(x_1 - x_2\right)\left(y_3 - y_1\right)\right]\right)^{-1}, \\ f &= \frac{x_1^2 - x_2^2 + y_1^2 - y_2^2 + 2g\left(x_1 - x_2\right)}{2\left(y_2 - y_1\right)}, \end{aligned} \tag{3}$$

where (x_1, y_1), (x_2, y_2), and (x_3, y_3) are the three arbitrary points.

Putting the values of f and g the value of c is determined from the following equation:

$$c = -x_1^2 - y_1^2 - 2gx_1 - 2fy_1. \tag{4}$$



(a) The figure shows an iris image after edge detection

(b) The figure shows the disk shaped characteristics of iris. Note the circles overlapping the inner and outer circular edges

Figure 1: The figure depicts iris image after edge detection making disk shaped edges apparent.

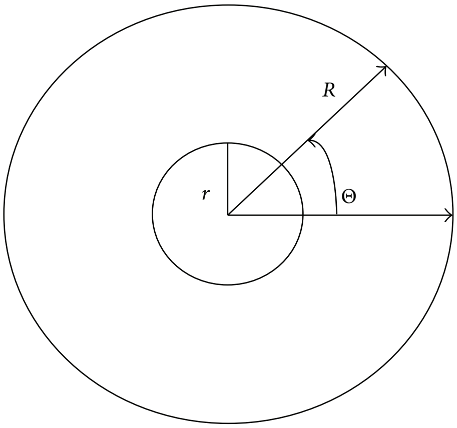


Figure 2: Transforming the radial iris into rectangular form.

Moreover the radius r is determined using (2). The center and the radii of both the concentric circles are determined in the described manner. The information within the inner circle is left out as it encompasses the pupil while the information bound in between the inner and outer circle contains the significant and unique iris pattern. Several triplets of circle points are used to compute the center of each circle. The best estimation of center is achieved by discarding extreme center points and then taking the mean of the rest of the points. For each center point (x_i, y_i) of inner circle and for each center point (u_i, v_i) for outer circle the mean (x_c, y_c) is computed as

$$x_c = \frac{\sum_{i=1}^{n} x_i + \sum_{i=1}^{n} u_i}{2n}, \quad y_c = \frac{\sum_{i=1}^{n} y_i + \sum_{i=1}^{n} v_i}{2n}. \tag{5}$$

Similarly, averages for the radius of the inner circle r_m and the radius for outer circle R_m are computed as

$$r_m = \frac{\sum_{i=1}^{n} r_i}{n}, \quad R_m = \frac{\sum_{i=1}^{n} R_i}{n}. \tag{6}$$

The pattern formed by the radial and longitudinal muscles of the iris is of interest. Further, this pattern is extracted to form a rectangular image. A computationally moderate solution to the problem must provide a faster transformation method. One such method is discussed in [5] which transforms the radial image into rectangular form and further makes use of the midpoint algorithm to optimize it as described in the next section.

4. Radial to Linear Transformation

An arbitrary Cartesian point (x', y') anywhere on the disk-like structure having parametric coordinates (r, θ) is given as

$$x' = r \cdot \cos(\theta), \quad y' = r \cdot \sin(\theta). \tag{7}$$

Cartesian points along the line starting at parametric point (r_m, θ) and ending at (R_m, θ) are discretized at appropriate intervals and are placed in a column of a two-dimensional array. A number of columns are collected starting from $\theta = 0^\circ$ and incrementing it in small steps up to $\theta = 360^\circ$. The collection of these columns forms a rectangular canvas containing the required iris pattern, (see Figure 3).

The computations required to compute each point within the disk shaped structures are reduced by exploiting the symmetric properties of a circle. A circular shape exhibits eight-way symmetry [14]. This means for any computed point (a, b) seven more points are determined that lie on the same circle using its symmetric properties. These seven points are described as $(-a, b)$, $(-b, a)$, $(a, -b)$, $(b, -a)$, $(-a, -b)$, and $(-b, -a)$ given that the center lies at the origin. In case the center lies at an arbitrary point, then these points are translated accordingly. Use of this symmetric property reduces the computations eightfold. Each point on the line is determined by incrementing the x-coordinate in discrete steps and calculating the corresponding value of y-coordinate using the line equation.

The x-coordinate and the y-coordinate of the pixels along a single line making an arbitrary angle θ can be determined

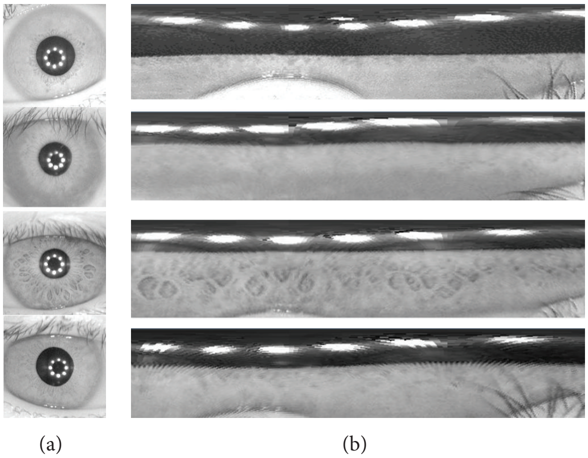


Figure 3: The figure illustrates iris images before and after radial to rectangular transformation.

incrementally while the starting pixel coordinates are (x', y') and the coordinates of the endpoint pixel are (x, y). Based upon the value of x-coordinate of previous pixel, the value of x-coordinate for next pixel is calculated by incrementing the previous value of x. This value of x-coordinate, say x_m, is put into the following equation:

$$y_m = \frac{y - y'}{x - x'} \left(x_m - x'\right) + y, \tag{8}$$

which yields the corresponding y-coordinate.

5. Optimizing the Algorithm

The determination of points along a line is further optimized by the use of the midpoint method [14]. The computations required to yield a point along a line are reduced to mere addition of a small incremental value. The gradient m is computed for this purpose as given in the following equation:

$$m = \frac{y - y'}{x - x'}. \tag{9}$$

Let

$$\begin{aligned} dy &= y - y', \\ dx &= x - x', \end{aligned} \tag{10}$$

where the line end points are (x', y') and (x, y). In accordance with the midpoint method for straight line with tangent between 0 and 1 the value of $\Delta E = 2(dy{-}dx)$ and $\Delta NE = 2dy$. Initially the control variable $d = 2dy - dx$. If the value of d is positive then East Pixel is chosen and if it is negative then North East Pixel is chosen. At each step d is updated by adding ΔE or ΔNE accordingly [5, 14].

6. Pattern Recognition Using Image Moments

A perceptive action performed on intricate structures needs to quantify its attributes. The state of any structure is quantifiable into data. Diversification of this data represents interaction or changes in the state. All such quantification methods generate finite data. Data by itself is insignificant, but the information implanted within the data is useful. Information is either extracted directly from the data itself or from the patterns formed by the arrangement of data. Researchers have devised various models for extracting information from data embedded in an image. Applications based on such models do not add to the contents of data rather they find hidden data patterns in order to extract interesting and useful information. A probability density can be formed for any data set. The parameters of the probability density function inform us about the general manner in which data is distributed. Moments are the characteristics of the probability density function which are based on the kurtosis and skewedness of the probability density function. Image moments describe the properties of a distribution formed using the pixel data of the image along its axes. The moments are typically chosen to depict a certain interesting property of the image. Such moment proves beneficial in extracting and summarizing the properties of the image in order to produce useful results. Properties of an image such as centroid, area, and orientation are quantified by this process. Another dividend of image moments is that they bring together the local and global geometric details of a grayscale image [15].

6.1. Extracting Moments from an Image. An image in the real world is modeled using a Cartesian distribution function $f(x, y)$ in its analog form. This function is used to provide moments of the order of $(p + q)$ over the image plane P and is generalized as

$$\begin{aligned} &M_{pq} \\ &= \int\int_{\mathrm{P}} \psi_{pq}(x, y) \cdot f(x, y)\, dx\, dy; \quad p, q = 0, 1, 2, \ldots, \infty, \end{aligned} \tag{11}$$

where ψ_{pq} is the basis function and P is the image plane. Equation (11) yields a weighted average over the plane P.

The basis function is designed such that it represents some invariant features of the image. Furthermore the properties of the basis function are passed onto moments. An image is of discrete nature; thus it is divided into pixels each having a discrete intensity level. Equation (11) is adopted for the digital image as

$$M_{pq} = \sum_{x}\sum_{y} \psi_{pq}(x, y) \cdot I(x, y); \quad p, q = 0, 1, 2, \ldots, \infty, \tag{12}$$

where $I(x, y)$ is the intensity of a pixel in the digital image at the xth row and yth column.

In [10, 16] the authors prove that the two-dimensional continuous $(p + q)$th order moments are defined using the integral

$$M_{pq} = \iint_{-\infty}^{\infty} x^{p} y^{q} I(x, y)\, dx\, dy, \tag{13}$$

where $f(x, y)$ lies within some finite region of the xy plane. In case of digital image the integrals are replaced by summations, which is formulated as

$$M_{pq} = \sum_{x=1}^{K}\sum_{y=1}^{L} x^{p} y^{q} I(x, y), \tag{14}$$

where $x^{p} y^{q}$ is the basis function, K and L are the dimensions of the image, and the M_{pq} is the cartesian moment for the two-dimensional image. Note that this basis function is highly correlated, that is, nonorthogonal. The moment M_{00} represents the image, whereas the first order moments are used to find the center of the mass or the centroid of the image is given as

$$\begin{aligned} \overline{x} &= \frac{M_{10}}{M_{00}}, \\ \overline{y} &= \frac{M_{01}}{M_{00}}, \end{aligned} \tag{15}$$

where $(\overline{x}, \overline{y})$ is the centroid.

6.2. Centralized Moments. Once the centroid is determined, it is used to compute the centralized moments. In [15] the central moments for two-dimensional data are given as

$$\mu_{pq} = \sum_{x}\sum_{y} (x - \overline{x})^{p} (y - \overline{y})^{q} I(x, y), \tag{16}$$

where μ_{pq} are the central moments. Note that these are similar to Cartesian moments translated to the centroid. This depicts the translation invariant property of the centralized moments which are always akin to the centroid of the segmented object.

Further simplification of (16) up to order 3 generates the following moments:

$$\begin{aligned} \mu_{00} &= M_{00}, \\ \mu_{01} &= 0, \\ \mu_{10} &= 0, \\ \mu_{11} &= M_{11} - \overline{x}M_{01} = M_{11} - \overline{y}M_{10}, \\ \mu_{20} &= M_{20} - \overline{x}M_{10}, \\ \mu_{02} &= M_{02} - \overline{y}M_{01}, \\ \mu_{21} &= M_{21} - 2\overline{x}M_{11} - \overline{y}M_{20} + 2\overline{x}^{2}M_{01}, \\ \mu_{12} &= M_{12} - 2\overline{y}M_{11} - \overline{x}M_{02} + 2\overline{y}^{2}M_{10}, \\ \mu_{30} &= M_{30} - 3\overline{x}M_{20} + 2\overline{x}^{2}M_{10}, \\ \mu_{03} &= M_{03} - 3\overline{y}M_{02} + 2\overline{y}^{2}M_{01}. \end{aligned} \tag{17}$$

6.3. Scale Invariant Moments. These moments are further made scale invariant as explained in [4, 5] and are given as

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}}, \tag{18}$$

where η_{pq} are scale normalized central moments and $\gamma = (p + q)/2 + 1$ where $(p + q) \geq 2$.

6.4. Image Orientation. The second order central moments contain information about the orientation of the image. Using these moments a covariance matrix is derived. Let

$$\begin{aligned} \mu_{20}' &= \frac{\mu_{20}}{\mu_{00}} = \frac{M_{20}}{M_{00}} - \overline{x}^{2}, \\ \mu_{02}' &= \frac{\mu_{02}}{\mu_{00}} = \frac{M_{02}}{M_{00}} - \overline{y}^{2}, \\ \mu_{11}' &= \frac{\mu_{11}}{\mu_{00}} = \frac{M_{11}}{M_{00}} - \overline{xy}, \end{aligned} \tag{19}$$

and then the covariance matrix is given as

$$\operatorname{cov}[I(x, y)] = \begin{pmatrix} \mu_{20}' & \mu_{11}' \\ \mu_{11}' & \mu_{02}' \end{pmatrix}. \tag{20}$$

The major and minor axes of the image intensity correlate with the eigenvectors of the given covariance matrix. The orientation of the image is described by the eigenvector of the highest eigenvalue. In [4] it is shown that the angle Θ is computed by the following equation:

$$\Theta = \frac{1}{2}\tan^{-1}\left(\frac{2\mu_{11}'}{\mu_{20}' - \mu_{02}'}\right), \tag{21}$$

where

$$\mu' \neq 0. \tag{22}$$

Using (20) the eigenvalues of the covariance matrix are easily obtained and are given as

$$\lambda_i = \frac{\mu'_{20} + \mu'_{02}}{2} \pm \frac{\sqrt{4\mu'^2_{11} + (\mu'_{20} - \mu'_{02})^2}}{2}. \tag{23}$$

Notice that these values are proportional to the square of the length of the eigenvector axes. The difference between the eigenvalues marks yet another important characteristic. It shows how elongated the image is. This property is termed eccentricity and is computed as

$$\sqrt{1 - \frac{\lambda_2}{\lambda_1}}. \tag{24}$$

6.5. Rotation Invariant Moments. Previously we have discussed translation and scale invariant moments. In [16] rotation invariant moments are derived which are usually termed as a Hu set of invariant moments. These are given as follows:

$$\begin{aligned}
I_1 &= \eta_{20} + \eta_{02}, \\
I_2 &= (\eta_{20} - \eta_{02})^2 + (2\eta_{11})^2, \\
I_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2, \\
I_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2, \\
I_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12}) \\
&\quad \times \left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] \\
&\quad + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \\
&\quad \times \left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right], \\
I_6 &= (\eta_{20} - \eta_{02}) \\
&\quad \times \left[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] \\
&\quad + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}), \\
I_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12}) \\
&\quad \times \left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] \\
&\quad - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03}) \\
&\quad \times \left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right], \\
I_8 &= \eta_{11}\left[(\eta_{30} + \eta_{12})^2 - (\eta_{03} + \eta_{21})^2\right] \\
&\quad - (\eta_{20} - \eta_{02})(\eta_{30} + \eta_{12})(\eta_{03} + \eta_{21}).
\end{aligned} \tag{25}$$

Every one of the rotation invariant moments extracts a characteristic attribute of the image. For example I_1 represents the moment of inertia along the centroid while I_7 extracts skew invariant properties which are useful in differentiating between images which are mirror reflections of each other [10, 15–17].

7. Clustering for Classification

By now the iris image has been segmented and transformed into a rectangular canvas. All described moments are applied and a feature vector is extracted, namely, $\bar{I}$. This vector contains translation, scale, and rotation invariant and orientation related moments. This vector corresponds to various features of the image; hence it is used for classification. An unsupervised approach is adopted for classification using k-means clustering algorithm. Using a set of multidimensional observations $(x_1, x_2, \ldots, x_n)$, the k-means algorithm partitions the observations into K sets such that $K \le n$ generating the set $S = S_1, S_2, \ldots, S_k$ so as to minimize the following objective function:

$$\underset{s}{\arg\min} \sum_{i=1}^{K} \sum_{x_j \in S_i} \left|x_j - \mu_i\right|^2, \tag{26}$$

where μ_i is mean of all the observations in S_i. Initially extracted moment vectors for an iris image sample are considered to be the initial mean.

The k-means Algorithm has two major steps, namely, the assignment step and the update step. The mean is used to assign each observation to a cluster in the assignment step. An observation is assigned a cluster whose mean makes the closest match. Formally this step generated the set S_i such that

$$S_i = x_r : \left|x_r - m_i\right| \le \left|x_r - m_j\right| \quad \forall 1 \le j \le K. \tag{27}$$

Also an observation x_r should be associated with exactly one S_i even if two or more differences are found comparable. The next step is based on the identification of a cluster for the observation established in the previous step. The mean for the cluster is recalculated as the centroid of the observations as given in following equation:

$$m_i = \frac{1}{\left|S_i\right|} \sum_{x_j \in S_i} x_j. \tag{28}$$

Both steps are iterated and the centroid is readjusted. This process continues until there is no appreciable change in the means. At this stage the means have converged and no further training is required [18, 19].

8. Results

The CASIA database containing thousands of images belonging to hundreds of different people is used to gather test results. Nearly one-fourth of the iris images from each class are retained as test case while the rest are used for training. The distorted images within the database are rejected. Iris portion of the image is marked out using the segmentation algorithm and is later transformed into a rectangular canvas. Further the grey scale rectangular canvas of iris is used

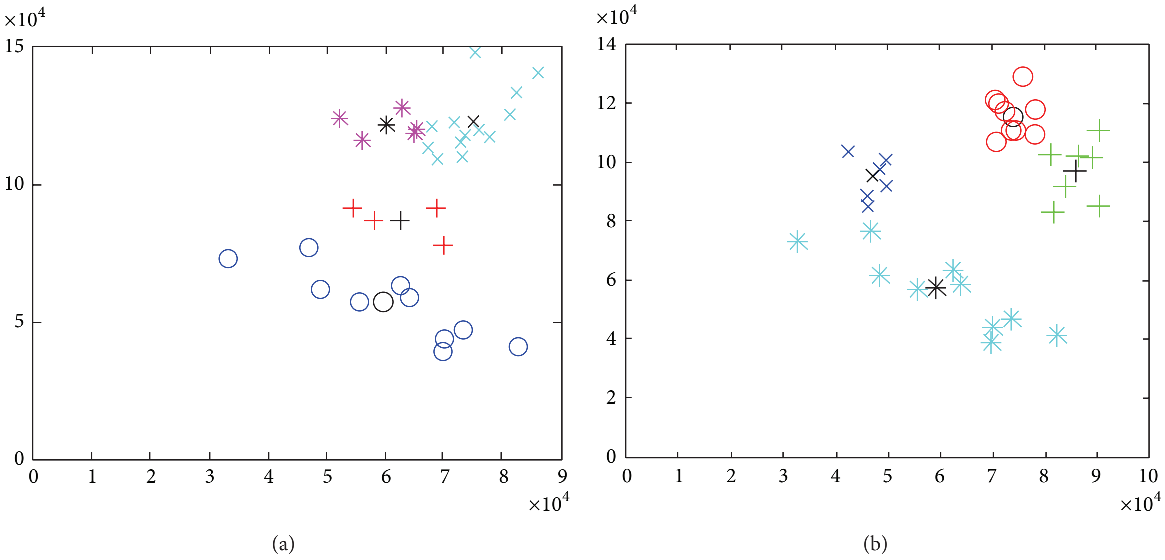


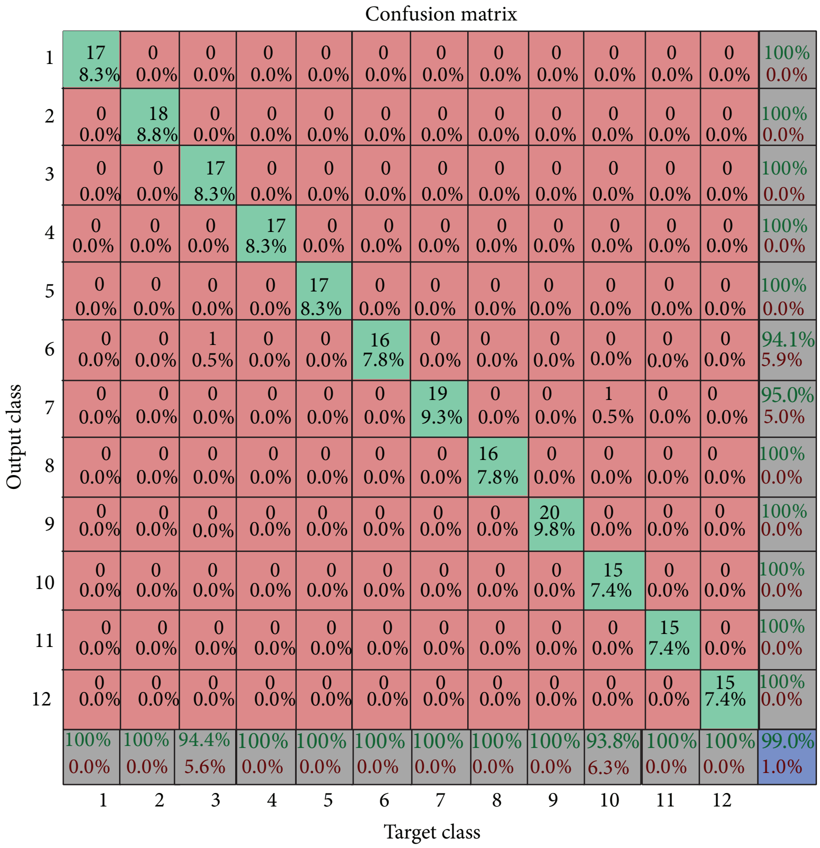
Figure 4: Each of (a) and (b) shows different clusters. Notice that all the clusters are linearly separable and can be distinguished by their Euclidean distance from the Centroid.

to compute image moment vector. This vector contains information which is translation, scale, and rotation invariant and provides orientation information. Using the k-means algorithm each image is assigned to a cluster. The k-means algorithm is iterated until convergence is achieved and the centroid of each cluster is determined. Once the system is fully trained it is ready to accept an arbitrary input and provide a match. The model responds with the correlation of an arbitrary image moments vector to a cluster, if the image belongs to a known class. In Figure 4 various clusters formed are depicted; it also shows how the class of a sample is distinguished based upon the Euclidean distance of the feature vector of the sample from the centroid of an arbitrary cluster. Moreover Figure 5 shows a confusion matrix depicting the accuracy of the model. The confusion matrix shows that the accuracy of the model for certain arbitrary classes is 99.0% while the overall accuracy of the model for all the images in the database is estimated to be 98.5%. Moreover it also reports the level of confidence of match based on Euclidean distance of the sample from the centroid of the identified cluster. Level 0 is the highest which means that the Euclidean distance of the sample from the centroid of the cluster is low and level 4 is the lowest which indicates that the Euclidean distance of the sample from the centroid of any cluster does not lie within a stipulated threshold to confidently indicate a match. Figure 4 shows the clusters formed using the k-means algorithm.

Furthermore a number of experiments were carried out to determine the accuracy and efficiency of the proposed model in comparison with other competitive models. In [20] the authors present a technique which extracts the features of iris using fine-to-coarse approximation at different resolution levels determined through discrete dyadic wavelet transform zero crossing representation. The resultant one-dimensional feature vector is used to find a match by computing various distances with arbitrary feature vectors. Ma et al. present yet another iris recognition technique using Gabor filters [1, 6]. The authors use a bank of Gabor filters to extract a fixed length feature vector signifying the global and local characteristics of the iris image. A match is established by computing the weighted Euclidean distance between feature vectors of arbitrary iris images. Daugman in [9] relies on the morphogenetic randomness of texture in the trabecular meshwork of iris. A failure of statistical independence test on two coded patterns from same iris indicates a match. This method extracts the visible texture of iris from a real time video image. The image is later encoded into a compact sequence of multiscale quadrature 2D Gabor wavelet coefficients. The most significant 256 bytes form the iris code. An exclusive OR operation is performed to generate a decision. All the above-mentioned techniques including the proposed technique are executed in order to obtain results. The genuine acceptance rate (GAR) and false acceptance rate (FAR) are observed for each technique. A receiver operating characteristics (ROC) distribution is plotted for each technique based on the results as shown in Figure 6. The ROC distribution comparatively highlights the accuracy along with the frequency of occurrence of errors of the proposed and other current state-of-art models. The following section briefly provides some discussion about the proposed system along with an interpretation of the ROC distribution formed.

9. Conclusion

Through analysis of data obtained after moments extraction a number of conclusions are inferred. Images of a certain iris differing in orientation yielded varying eigenvalues and eccentricity. However, a change in orientation of an image barely affects the values of rotation invariant moments while raw and scale invariant moments are affected. Change in orientation of an image affects the Euclidean distance of the moment vectors from the centroid. Despite this there still remains a great probability of the image to be classified



Confusion matrix

Output class \ Target class	1	2	3	4	5	6	7	8	9	10	11	12	
1	17 8.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	18 8.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	17 8.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	17 8.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	17 8.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
6	0 0.0%	0 0.0%	1 0.5%	0 0.0%	0 0.0%	16 7.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	94.1% 5.9%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	19 9.3%	0 0.0%	0 0.0%	1 0.5%	0 0.0%	0 0.0%	95.0% 5.0%
8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	16 7.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
9	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	20 9.8%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
10	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	15 7.4%	0 0.0%	0 0.0%	100% 0.0%
11	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	15 7.4%	0 0.0%	100% 0.0%
12	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	15 7.4%	100% 0.0%
	100% 0.0%	100% 0.0%	94.4% 5.6%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	93.8% 6.3%	100% 0.0%	100% 0.0%	99.0% 1.0%

FIGURE 5: The figure shows the confusion matrix for some arbitrary classes, while the accuracy of the model for these classes is 99.0%.

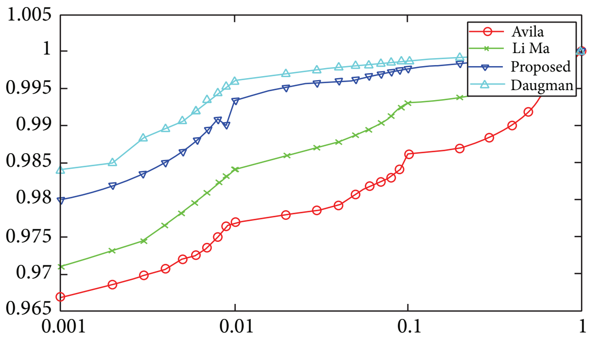


FIGURE 6: The figure illustrates the receiver operating characteristics distributions for different competitive techniques including the proposed one.

correctly because of coherence in scale invariant moments. Although the model exhibits scale and rotation invariant attributes but some impairment is offered by luminosity of the image. Two arbitrary images of the same objects yield comparable moments if the luminosity is the same but they may yield differing moments in case luminosity is altered. In the underlying research work it is assumed that the luminosity level will be the same for all the images as each image is obtained by an iriscope working in similar conditions. The model provides resilience towards variation of scale and rotation as compared to other techniques which requires coherence of phase and size. The model can be further improved by incorporation of a technique that will process each image to provide uniform luminosity. Furthermore, the ROC distribution obtained (shown in Figure 6) from all the test cases shows that the performance of proposed model is comparable with Daugman method, while it yields a better performance than the methods described in [1, 6, 20].

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] L. Ma, T. Tan, Y. Wang, and D. Zhang, "Efficient iris recognition by characterizing key local variations," *IEEE Transactions on Image Processing*, vol. 13, no. 6, pp. 739–750, 2004.

[2] D. M. Monro, S. Rakshit, and D. Zhang, "DCT-bsed iris recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 4, pp. 586–595, 2007.

[3] K. Roy, P. Bhattacharya, and R. C. Debnath, "Multi-class SVM based iris recognition," in *Proceedings of the 10th International Conference on Computer and Information Technology (ICCIT '07)*, pp. 1–6, Dhaka, Bangladesh, December 2007.

[4] R. H. Abiyev and K. Altunkaya, "Personal iris recognition using neural network," *International Journal of Security and Its Applications*, vol. 2, no. 2, pp. 41–50, 2008.

[5] Y. D. Khan, F. Ahmad, and M. W. Anwar, "A neuro-cognitive approach for iris recognition using back propagation," *World Applied Sciences Journal*, vol. 16, no. 5, pp. 678–685, 2012.

[6] L. Ma, Y. Wang, and T. Tan, "Iris recognition based on multichannel Gabor filtering," in *Proceedings of the 5th Asian Conference on Computer Vision*, pp. 279–283, Melbourne, Australia, 2002.

[7] W. W. Boles and B. Boashash, "A human identification technique using images of the iris and wavelet transform," *IEEE Transactions on Signal Processing*, vol. 46, no. 4, pp. 1185–1188, 1998.

[8] K. Lee, S. Lim, O. Byeon, and T. Kim, "Efficient iris recognition through improvement of feature vector and classifier," *ETRI Journal*, vol. 23, no. 2, pp. 61–70, 2001.

[9] J. Daugman, "How iris recognition works," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 21–30, 2004.

[10] M.-K. Hu, "Visual pattern recognition by moment invariants," *IRE Tranactions on Information Theory*, vol. 8, no. 2, pp. 179–187, 1962.

[11] E. Arias-Castro and D. L. Donoho, "Does median filtering truly preserve edges better than linear filtering?" *Annals of Statistics*, vol. 37, no. 3, pp. 1172–1206, 2009.

[12] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.

[13] N. Kanopoulos, N. Vasanthavada, and R. L. Baker, "Design of an image edge detection filter using the Sobel operator," *IEEE Journal of Solid-State Circuits*, vol. 23, no. 2, pp. 358–367, 1988.

[14] J. D. Foley, *Computer Graphics: Principles and Practice*, Addison-Wesley, Reading, Mass, USA, 1996.

[15] S. O. Belkasim, M. Shridhar, and M. Ahmadi, "Pattern recognition with moment invariants: a comparative study and new results," *Pattern Recognition*, vol. 24, no. 12, pp. 1117–1138, 1991.

[16] J. Flusser, T. Suk, and B. Zitová, *Moments and Moment Invariants in Pattern Recognition*, John Wiley & Sons, Chichester, UK, 2009.

[17] W. T. Freeman, D. B. Anderson, P. A. Beardsley et al., "Computer vision for interactive computer graphics," *IEEE Computer Graphics and Applications*, vol. 18, no. 3, pp. 42–52, 1998.

[18] D. Aloise, A. Deshpande, P. Hansen, and P. Popat, "NP-hardness of Euclidean sum-of-squares clustering," *Machine Learning*, vol. 75, no. 2, pp. 245–248, 2009.

[19] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: a k-means clustering algorithm," *Journal of the Royal Statistical Society C: Applied Statistics*, vol. 28, no. 1, pp. 100–108, 1979.

[20] C. Sanchez-Avila and R. Sanchez-Reillo, "Two different approaches for iris recognition using Gabor filters and multiscale zero-crossing representation," *Pattern Recognition*, vol. 38, no. 2, pp. 231–240, 2005.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 736106, 8 pages
http://dx.doi.org/10.1155/2014/736106

Research Article

Pain Expression Recognition Based on pLSA Model

Shaoping Zhu

Department of Information Management, Hunan University of Finance and Economics, Changsha 410205, China

Correspondence should be addressed to Shaoping Zhu; zenglegen@163.com

Received 9 January 2014; Accepted 19 February 2014; Published 27 March 2014

Academic Editors: Y.-B. Yuan and S. Zhao

We present a new approach to automatically recognize the pain expression from video sequences, which categorize pain as 4 levels: "no pain," "slight pain," "moderate pain," and " severe pain." First of all, facial velocity information, which is used to characterize pain, is determined using optical flow technique. Then visual words based on facial velocity are used to represent pain expression using bag of words. Final pLSA model is used for pain expression recognition, in order to improve the recognition accuracy, the class label information was used for the learning of the pLSA model. Experiments were performed on a pain expression dataset built by ourselves to test and evaluate the proposed method, the experiment results show that the average recognition accuracy is over 92%, which validates its effectiveness.

1. Introduction

In recent years, tremendous amounts of researches have been carried out in the field of automatic expressions (such as pain, anger, and sadness) recognition from video sequence. Pain is a subjective and personal experience, and pain recognition is still difficult. There are numerous potential applications for pain recognition. Doctors can recognize pain when patients are experiencing genuine pain so that their pains are taken seriously, like young children who could not self-report pain measures, or many patients in postoperative care or transient states of consciousness, and with severe disorders requiring assisted breathing, among other conditions [1, 2]. Real-time automatic system can be trained which could potentially provide significant advantage in patient care and cost reduction.

Measuring or monitoring pain is normally conducted via self-report as it is convenient and requires no special skill or staffing. However, self-report measures cannot be used when patients cannot communicate verbally. Many researchers have pursued the goal of obtaining a continuous objective measure of pain through analyses of tissue pathology, neurological "signatures," imaging procedures, testing of muscle strength, and so on [3]. These approaches have been fraught with difficulty because they are often inconsistent with other evidence of pain [3], in addition to being highly invasive and constraining to the patient.

The experience of pain is often represented by changes in facial expression. So, facial expression is considered to be the most reliable source of information when judging the pain intensity experienced by another. In the past several years, significant efforts have been made to identify reliable and valid facial indicators of pain [4–14]. In [4, 5], an approach was developed to automatically recognize acute pain; active appearance models (AAM) were used to decouple shape and appearance parameters from face images; based on AAM, three pain representations were derived. And then SVM were used to classify pain. In [6–10], Prkachin and Solomon validated a facial action coding system (FACS) based measure of pain that can be applied on a frame-by-frame basis. But these methods require manual labeling of facial action units or other observational measurements by highly trained observers [15, 16], which is both timely and costly. Most must be performed offline, which makes them ill-suited for real-time applications in clinical settings. In [11], a robust approach for pain expression recognition was presented using video sequences. An automatic face detector is employed which uses skin color modeling to detect human face in the video sequence. The pain affected portions of the face are obtained by using a mask image.

The obtained face images are then projected onto a feature space, defined by Eigenfaces, to produce the biometric template. Pain recognition is performed by projecting a new image onto the feature spaces spanned by the Eigenfaces and then classifying the painful face by comparing its position in the feature spaces with the positions of known individuals. Zhang and Xia [12] used supervised locality preserving projections (SLPP) to extract feature of pain expression, and multiple kernels support vector machines (MKSVM) are employed for recognizing pain expression. Methods described above used static features to character pain expression, but these static features cannot fully represent pain.

In this paper, we propose a method for automatically inferring pain form video sequences. This approach includes two steps: extracting feature of pain expression and classifying pain expression. In the extracting feature, features of pain expression are extracted by motion descriptor based on optical flow. Then we convert facial velocity information to visual words using "bag-of-words" models, and pain expression is represented by a number of visual words; final pLSA model is used for pain expression recognition. In addition, in order to improve the recognition accuracy, the class label information was used for the learning of the pLSA model.

The paper is structured as follows. After reviewing related work in this section, we describe the pain feature extraction based on optical flow technique and "bag-of-words" models in Section 2. Section 3 gives details of pLSA model for recognizing pain expression. Section 4 shows experiment result, also comparing our approach with three state-of-the-art methods, and the conclusions are given in the final section.

2. Pain Expression Representation

2.1. Facial Velocity Feature. According to the physiology, the experience of pain is often represented by changes in facial expression and the expression is a dynamic event; it is must be represented by the motion information of the face. So, we use facial velocity features to characterize pain. The facial velocity features (optical flow vector) are estimated by optical flow model, and each pain expression was coded on a 4-level intensity dimension (A–D): "no pain," "slight pain," "moderate pain," and "severe pain."

Given a stabilized video sequence in which the face of a person appears in the center of the field of view, we compute the facial velocity (optical flow vector) $\mathbf{u} = (u_x, u_y)$ at each frame using optical flow equation, which is expressed as

$$I_x u_x + I_y u_y + I_t = 0, \tag{1}$$

where

$$\begin{gathered} I_x = \frac{\partial I}{\partial x}, \quad I_y = \frac{\partial I}{\partial y}, \quad I_t = \frac{\partial I}{\partial t}, \\ u_x = \frac{dx}{dt}, \quad u_y = \frac{dy}{dt}, \end{gathered} \tag{2}$$

where (x, y, t) is the image in pixel (x, y) at time t, where $I(x, y, t)$ is the intensity at pixel (x, y) and time t, u_x, u_y are the horizontal and vertical velocities in pixel (x, y).

We can obtain $\mathbf{u} = (u_x, u_y)$ by minimizing the objective function:

$$C = \int_D \left[\lambda^2 \|\nabla u\|^2 + \left(\nabla I \cdot u + I_t\right)^2\right] dx\, dy. \tag{3}$$

There are many methods to solve the optical flow equation. We use the iterative algorithm [17] to compute the optical flow velocity:

$$\begin{gathered} u_x^{k+1} = \overline{u}_x^k - \frac{I_x\left[I_x \overline{u}_x^k + I_y \overline{u}_y^k + I_t\right]}{\lambda + I_x^2 + I_y^2}, \\ u_y^{k+1} = \overline{u}_y^k - \frac{I_y\left[I_x \overline{u}_x^k + I_y \overline{u}_y^k + I_t\right]}{\lambda + I_x^2 + I_y^2}, \end{gathered} \tag{4}$$

where k is the number of iterations, initial value of velocity $u_x^0 = u_y^0 = 0$,and $\overline{u}_x^k, \overline{u}_y^k$ is the average velocity of the neighborhood of point (x, y).

The optical flow vector field $\mathbf{u}$ is then split into two scalar fields u_x and u_y, corresponding to the x and y components of $\mathbf{u}$ [18]. u_x and u_y are further half-wave rectified into four nonnegative channels u_x^+, u_x^-, u_y^+, and u_y^-, so that $u_x = u_x^+ - u_x^-$ and $u_x = u_x^+ - u_x^-$. These four nonnegative channels are then blurred with a Gaussian kernel and normalized to obtain the final four channels $ub_{x^+}, ub_{x^-}, ub_{x^-}$, and ub_{x^-}.

Facial pain expressionisrepresented by velocity features that are composed of the channels $ub_{x^+}, ub_{x^-}, ub_{y^+}$, and ub_{y^-} of all pixels in facial image. Because pain expression can be regarded as facial motion, the velocity features can describe pain effectively, in addition to the velocity features having been shown to perform reliably with noisy image sequences [18], and have been applied in various tasks, such as action classification and motion synthesis. But the dimension of these velocity features is too high ($4 \times N \times N$, where $N \times N$ is image size) to be used directly for recognition and, so, we convert these velocity features into visual words using "bag of words" [19, 20].

2.2. Visual Words for Characterizing Pain. The "bag-of-words" model was originally proposed for analyzing text documents, where a document is represented as a histogram over word counts.

In this paper, each facial image is divided into blocks whose size is $L \times L$, and each image block is represented by optical flow vector of all pixels in the block. On this basis, pain is represented by visual words using the method of BoW (bag of words).

To construct the codebook, we randomly select a subset from all image blocks; then, we use k-means clustering algorithms to obtain V clusters. Codewords are then defined as the centers of the obtained clusters, namely, visual words. In the end, each face image is converted to the "bag-of-words" representation by appearance times of each codeword in the image that is used to represent the image, namely, BoW histogram.

The step for characterizing pain is as follows.

Step 1. Optical flow channels $ub_{x^+}, ub_{x^-}, ub_{y^+}$, and ub_{y^-} are computed.

Step 2. Each facial image is divided into $n \times n$ blocks, which is represented by optical flow vector of all pixels in the block.

Step 3. Vision words are obtained using k-means clustering algorithms.

Step 4. Pain expression is represented by BoW histogram d:

$$d = \left\{ n(I, w_1), \ldots, n\left(I, w_j\right), \ldots, n(I, w_M) \right\}, \quad (5)$$

where $n(I, w_j)$ is the number of visual word w_j included in image and M is the number of vision words in word sets.

Figure 1 shows an example of our "bag-of-words" representation.

3. pLSA-Based Pain Expression Recognition

We use the pLSA models [21] to learn and recognize human pain. Our approach is directly inspired by a body of work on using generative topic models for visual recognition based on the "bag-of-words" paradigm. The pLSA models have been applied to various computer vision applications, such as scene recognition, object recognition, action recognition, and human detection [22–26].

3.1. Probabilistic Latent Semantic Analysis (pLSA). pLSA is a statistical generative model that associates documents and words via the latent topic variables, which represents each document as a mixture of topics. We briefly outline the principle of the pLSA in this subsection. The model of pLSA is shown in Figure 2.

Suppose document, word, and topic are represented by d_i, w_j, and z_k, respectively. The joint probability of document d_i, topic z_k, and word w_j can be expressed as

$$p\left(d_i, z_k, \omega_j\right) = p\left(\omega_j \mid z_k\right) p\left(z_k \mid d_i\right) p\left(d_i\right), \quad (6)$$

where $p(\omega_j \mid z_k)$ is the probability of word ω_j occurring in pain category z_k, $p(z_k \mid d_i)$ is the probability of topic z_k occurring in image d_i, and $p(d_i)$ can be considered as the prior probability of d_i. The conditional probability of $p(\omega_j \mid d_i)$ can be obtained by marginalizing over all the topic variables z_k:

$$p\left(\omega_j \mid d_i\right) = \sum_k p\left(z_k \mid d_i\right) p\left(\omega_j \mid z_k\right). \quad (7)$$

Denote $n(d_i, \omega_j)$ as the occurrence of word ω_j in image d_i; the prior probability $p(d_i)$ can be modeled as

$$p\left(d_i\right) \propto \sum_j n\left(d_i, \omega_j\right). \quad (8)$$

A maximum likelihood estimation of $p(\omega_j \mid z_k)$ and $p(z_k \mid d_i)$ is obtained by maximizing the function using the expectation maximization (EM) algorithm. The objective likelihood function of the EM algorithm is

$$l = \prod_i \prod_j p\left(\omega_j \mid d_i\right)^{n(w_j, d_i)} \quad (9)$$

or

$$l = \sum_i \sum_j n\left(d_i, w_j\right) \log p\left(\omega_j \mid d_i\right). \quad (10)$$

The EM algorithm consists of two steps: an expectation (E) step computes the posterior probability of the latent variables, and a maximization (M) step maximizes the completed data likelihood computed based on the posterior probabilities obtained from E-step. Both steps of the EM algorithm for pLSA parameter estimate are listed below.

E-Step. Given $p(\omega_j \mid z_k)$ and $p(z_k \mid d_i)$, estimate $p(z_k \mid d_i, w_j)$:

$$p\left(z_k \mid d_i, w_j\right) = \frac{p\left(\omega_j \mid z_k\right) p\left(z_k \mid d_i\right)}{\sum_l p\left(\omega_j \mid z_l\right) p\left(z_l \mid d_i\right)}. \quad (11)$$

M-Step. Given the estimated $p(z_k \mid d_i, w_j)$ in E-Step and $n(d_i, w_j)$, estimate $p(w_j \mid z_k)$ and $p(z_k \mid d_i)$:

$$p\left(\omega_j \mid z_k\right) = \frac{\sum_i n\left(d_i, w_j\right) p\left(z_k \mid d_i, w_j\right)}{\sum_i \sum_h n\left(d_i, w_h\right) p\left(z_k \mid d_i, w_h\right)}, \quad (12)$$

$$p\left(z_k \mid d_i\right) = \frac{\sum_j n\left(d_i, w_j\right) p\left(z_k \mid d_i, w_j\right)}{n\left(d_i\right)}, \quad (13)$$

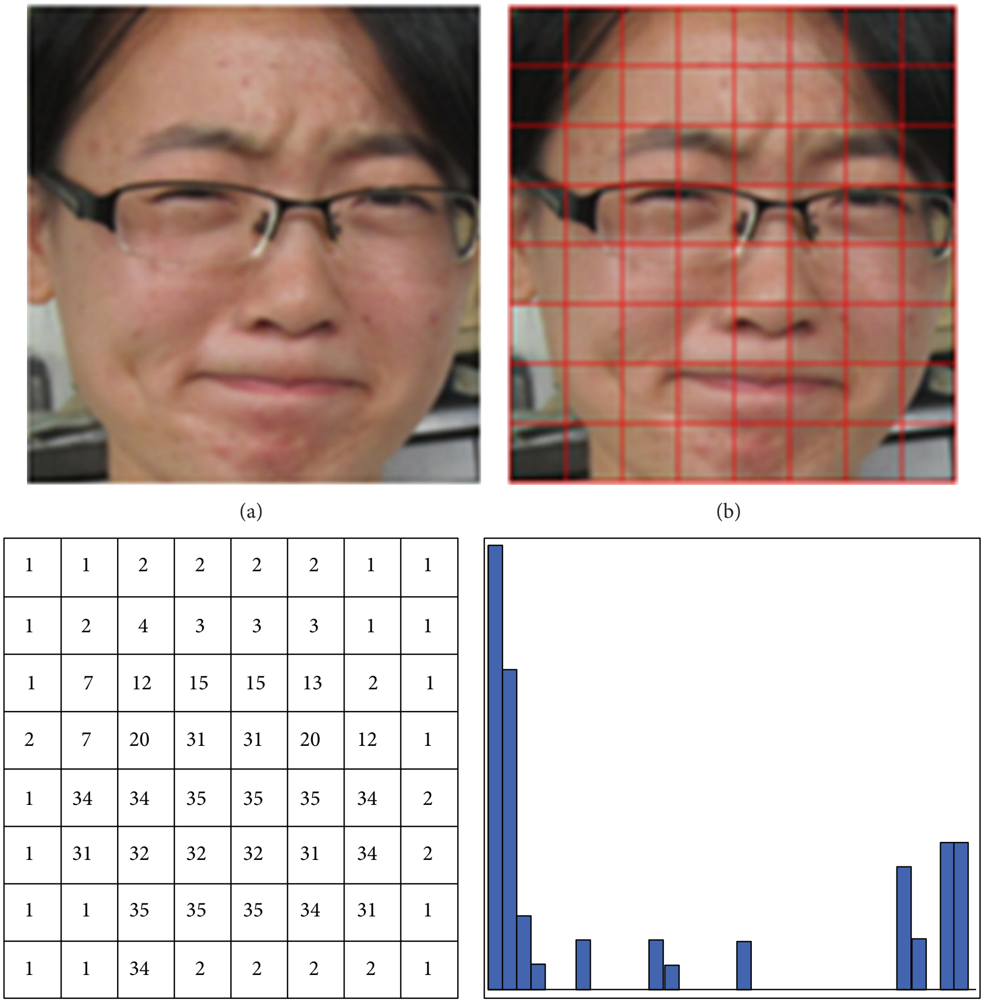
where $n(d_i) = \sum_j n(d_i, w_j)$ is the length of document d_i.

Given a new document, the conditional probability distribution over aspect $p(z \mid d_{\text{new}})$ can be inferred by maximizing the likelihood of d_{new} using a fixed word-aspect distribution $p(\omega_j \mid z_k)$ learned from the observed data [21]. The iteration of inferring $p(z \mid d_{\text{new}})$ is the same as the learning process except that the word-topic distribution $p(\omega_j \mid z_k)$ in (12) is a fixed value, that is, learned from training data.

3.2. pLSA-Based Pain Expression Recognition. In this paper, we treat each block in an image as a single word w_j, an image as a document d_i, and a pain category as a topic variable z_k. For the task of pain classification, our goal is to classify a new face image to a specific pain class. During the inference stage, given a testing face image and the document specific coefficients $p(z_k \mid d_{\text{test}})$, we can treat each aspect in the pLSA model as one class of pains. So, the pain categorization is determined by the aspect corresponding to the highest $p(z_k \mid d_{\text{test}})$. The pain category k of d_{test} is determined as

$$k = \arg\max_k p\left(z_k \mid d_{\text{test}}\right). \quad (14)$$

For pain recognition with large amount of training data, this would result in long training time. In this paper, we adopt a supervised algorithm to train pLSA, which is similar to [27].



(a)

(b)

1	1	2	2	2	2	1	1
1	2	4	3	3	3	1	1
1	7	12	15	15	13	2	1
2	7	20	31	31	20	12	1
1	34	34	35	35	35	34	2
1	31	32	32	32	31	34	2
1	1	35	35	35	34	31	1
1	1	34	2	2	2	2	1

(c)

(d)

Figure 1: The processing pipeline of the "bag-of-words" representation: (a) give an image, (b) divide into $L \times L$ blocks, (c) represent each block by a "visual word," and (d) ignore the ordering of words and represent the facial image as a histogram over "visual words."

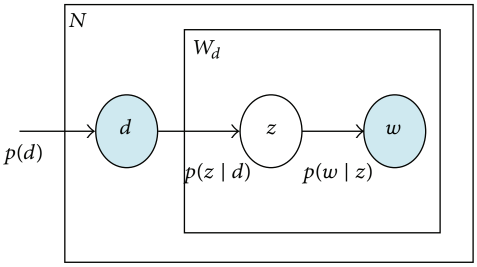


Figure 2: Graph model of pLSA. Nodes represent random variables. Shaded nodes are observed variables and unshaded ones are unseen variables. The plates stand for repetitions.

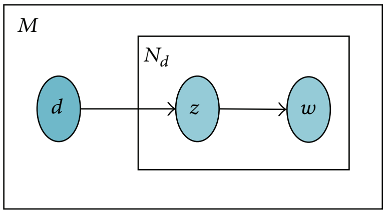


Figure 3: Graph model of SpLSA. Nodes represent random variables. Shaded nodes are observed variables and unshaded ones are unseen variables. The plates stand for repetitions.

Each image has class label information in the training images, which is important for the classification task. Here, we make use of this class label information in the training images for the learning of the pLSA model, since each image directly corresponds to a certain pain class on train sets; the image for training data becomes observable. This model is called supervised pLSA (SpLSA). The graphical model of SpLSA is shown in Figure 3.

The parameter $p(w_j|z_k)$ in the training step defines the probability of a word w_j drawing from a topic z_k. Letting each topic in pLSA correspond to a pain category, the distribution $p(w_j|z_k)$ in the training can be simply estimated as

$$p\left(\omega_j \mid z_k\right) = \frac{n_{j,k}}{n_k}, \tag{15}$$

where n_k is the number of the images corresponding to the kth pain class and $n_{j,k}$ is the number of the jth word (block) in the images corresponding to the kth pain class. This means that the $p(\omega_j \mid z_k)$ calculated by this way can be used to initialize the $p(w \mid z)$ in the EM algorithm for model learning, which makes the EM algorithm converge more quickly. The supervised training of pLSA is summarized in Algorithm 1. Once the distribution $p(w \mid z)$ is computed by the EM algorithm, for a testing face image d_{test}, the posterior distribution $p(z_k \mid d_{\text{test}})$ can be calculated the same as in original pLSA. The training of pLSA for classification is summarized in Algorithm 2.

Algorithm 1. Supervised training of the pLSA.

Step 1. For all k and j, calculate

$$p\left(\omega_j \mid z_k\right) = \frac{n_{j,k}}{n_k}, \tag{16}$$

as the initialization of the $p(w \mid z)$ and random initialization of the $p(z \mid d)$.

Step 2. E-Step: for all (d_i, w_j) pairs, calculate

$$p\left(z_k \mid d_i, w_j\right) = \frac{p\left(\omega_j \mid z_k\right) p\left(z_k \mid d_i\right)}{\sum_l p\left(w_j \mid z_l\right) p\left(z_l \mid d_i\right)}. \tag{17}$$

Step 3. M-Step: substitute $p(z_k \mid d_i, w_j)$ as calculated in Step 2; for all k and j, calculate

$$p\left(\omega_j \mid z_k\right) = \frac{\sum_i n\left(d_i, w_j\right) p\left(z_k \mid d_i, w_j\right)}{\sum_m \sum_i n\left(d_i, w_m\right) p\left(z_k \mid d_i, w_m\right)}. \tag{18}$$

Step 4. M-Step: substitute $p(z_k \mid d_i, w_j)$ as calculated in Step 2; for all i and k, calculate

$$p\left(z_k \mid d_i\right) = \frac{\sum_j n\left(d_i, w_j\right) p\left(z_k \mid d_i, w_j\right)}{n\left(d_i\right)}. \tag{19}$$

Step 5. Repeat Steps 2–4 until the convergence condition is met.

The supervised training algorithm not only makes the training more efficient, but also improves the overall recognition accuracy significantly.

Algorithm 2. Training of the pLSA for classification

Step 1. For all k and j, calculate

$$p\left(\omega_j \mid z_k\right) = \frac{n_{j,k}}{n_k}. \tag{20}$$

Step 2. E-Step: for all (d_{test}, w_j) pairs, calculate

$$p\left(z_k \mid d_{\text{test}}, w_j\right) = \frac{p\left(\omega_j \mid z_k\right) p\left(z_k \mid d_{\text{test}}\right)}{\sum_l p\left(w_j \mid z_l\right) p\left(z_l \mid d_{\text{test}}\right)}. \tag{21}$$

Step 3. Partial M-Step: fix $p(w_j \mid z_k)$ as calculated in Step 1; for all k, calculate

$$p\left(z_k \mid d_{\text{test}}\right) = \frac{\sum_j n\left(d_{\text{test}}, w_j\right) p\left(z_k \mid d_{\text{test}}, w_j\right)}{n\left(d_{\text{test}}\right)} \tag{22}$$

Step 4. Repeat Steps 2 and 3 until the convergence condition is met.

Step 5. Calculate pain class

$$k = \arg\max_k p\left(z_k \mid d_{\text{test}}\right). \tag{23}$$

4. Experimental Results and Analysis

The effectiveness of the proposed algorithm was verified by using C++ and MATLAB hybrid implementation on a PC with Pentium 3.2 GHz processor and 4 G RAM.

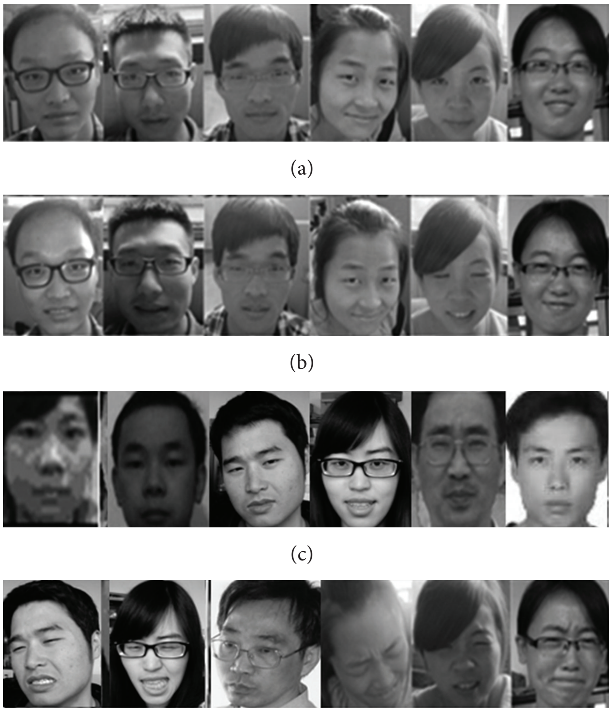
We have built a database of painful and normal face images. In this database, there are four groups of images ("no pain," "slight pain," "moderate pain," and "severe pain"), and each group includes 20 males and 20 females. The face images were taken under various laboratory-controlled lighting conditions, and each face image was normalized to a size of 64 × 64. Some sample images are shown in Figure 4.

In experiments, 30 face images per class are randomly chosen for training, while the remaining images are used for testing. We preprocessed these images by aligning and scaling them so that the distances between the eyes were the same for all images and also ensuring that the eyes occurred in the same coordinates of the image. We run the system 5 times and obtain 5 different training and testing sample sets. The recognition rates were found by averaging the recognition rate of each run.

Each facial image is divided into blocks whose size is $L \times L$. First, we studied the effect of the size of image block on the recognition accuracy. Figure 5 represents the recognition accuracy curve with different block sizes L. It can be concluded that the accuracy peaks when the block sizes L is 8. Therefore L is set as 8.

In order to determine the value of M, that is, the number of the visual word set, the relation between M and recognition accuracy was observed, which is displayed in Figure 6. It is revealed in Figure 4 that the recognition accuracy is risen up at the beginning with the increasing of M recognition and if M is larger than or equal to 60, the recognition accuracy is stabled to 0.922. As a result, M is set as 60.

To examine the accuracy of our proposed pain recognition approach, we compare our method to three state-of-the-art approaches for pain recognition using the same data. The first method is "AAM + SVM" [4], which used active appearance models (AAM) to extract face features and SVM to classify pain. The second method is "Eigenimage"


(a)

(b)

(c)

(d)

Figure 4: Examples of recognizing pain expression from facial videos. (a) No pain, (b) slight pain, (c) moderate pain, and (d) severe pain.

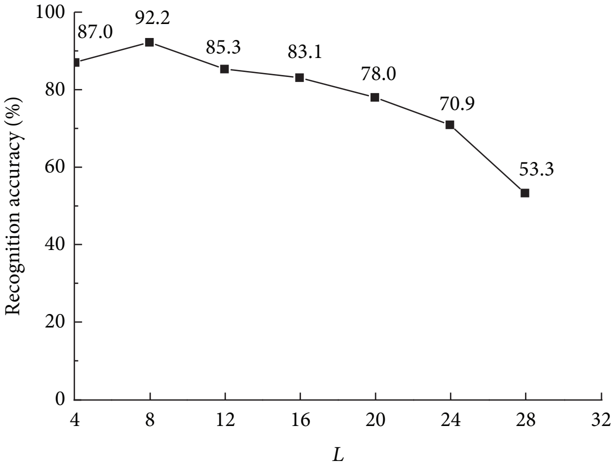


Figure 5: Recognition accuracy curve with different block sizes.

Table 1: Confusion matrix for pain recognition.

(a)

	A	B	C	D
A	**0.95**	0.04	0.01	0.00
B	0.04	**0.91**	0.05	0.00
C	0.01	0.03	**0.91**	0.05
D	0.00	0.02	0.05	**0.93**

(b)

	A	B	C	D
A	**0.84**	0.10	0.03	0.02
B	0.11	**0.78**	0.08	0.03
C	0.03	0.09	**0.79**	0.08
D	0.01	0.05	0.12	**0.83**

(c)

	A	B	C	D
A	**0.85**	0.11	0.03	0.01
B	0.10	**0.80**	0.08	0.02
C	0.02	0.08	**0.79**	0.08
D	0.01	0.04	0.12	**0.84**

(d)

	A	B	C	D
A	**0.88**	0.09	0.02	0.01
B	0.09	**0.82**	0.07	0.02
C	0.02	0.08	**0.83**	0.07
D	0.00	0.02	0.08	**0.90**

Table 2: Comparison of different reported results on pain dataset.

Method	Accuracy (%)
Our method	92.20
“AAM + SVM” [4]	81.20
“Eigenimage” [11]	82.50
“SLPP + MKSVM” [13]	86.50

[11], which used Eigenface for pain recognition. The third method is “SLPP + MKSVM” [12], which used SLPP to extract feature of pain expression and multiple kernels support vector machines (MKSVM) for recognizing. 200 different expression images are used for this experiment. Some images contain the same person but in different moods. The recognition results are presented in the confusion matrices shown in Table 1. Each cell in the confusion matrix is the average results; our results are at the upper left; the results of “AAM + SVM,” “Eigenimage,” and “SLPP + MKSVM” are presented in the upper right, the lower left, and the lower right, respectively. where A, B, C, and D indicate no pain, slight pain, moderate pain, and severe pain, respectively. As Table 1 shows, our method improves the recognition accuracies in all categories. It achieves 92.2% average recognition rate, whereas “AAM + SVM” obtain 81.2%, “Eigenimage” gets 82.5%, and “SLPP + MKSVM” attains 86.5%, as shown in Table 2. The reason is that we improve the recognition accuracy in the two stages of pain feature extraction and pain expression recognition. In the stage of pain feature extraction, we use motion features that are reliable with noisy image sequences and describe pain effectively, while other methods used static features, which cannot effectively describe the expression of pain. In the stage of expression recognition, we use bag-of-words framework and pLAS model to classify expression images. In addition, we make use of this class label information in

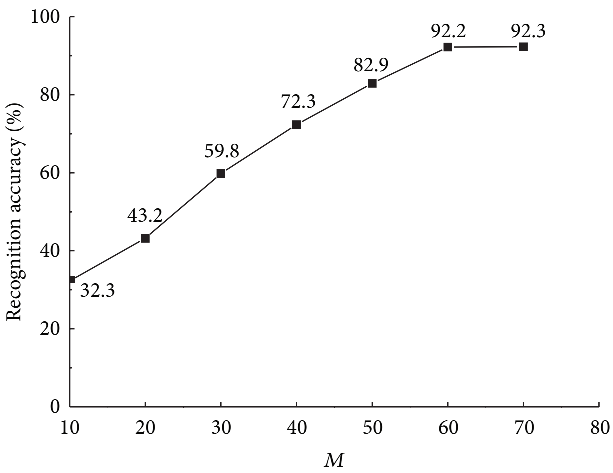


Figure 6: Relation curve between M and accuracy.

the training images for the learning of the pLSA model, which can improve the overall recognition accuracy significantly.

5. Conclusion

Pain recognition can provide significant advantage in patient care and cost reduction. In this paper, we present a novel method to recognize the pain expression and give the pain level at the same time. The main contribution can be concluded as follows.

(1) Visual words are used for pain expression. Optical flow model is used for extracting facial velocity features; then we convert facial velocity features into visual words using “bag-of-words” models.

(2) We use pLSA topic models for pain expression recognition. In our models the “latent topics” directly correspond to different pain expression categories. In addition, in order to improve the recognition accuracy, the class label information was used for the learning of the pLSA model.

(3) Experiments were performed on a pain expression dataset built by ourselves and evaluate the proposed method. Experimental results reveal that the proposed method performs better than previous ones.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by Research Foundation for Science & Technology Office of Hunan Province under Grant no. 2014FJ3057, by Hunan Provincial Education Science and “Twelve Five” planning issues (no. XJK012CGD022), by the Teaching Reform Foundation of Hunan Province Ordinary College under Grant no. 2012401544, and by the Foundation for Key Constructive Discipline of Hunan Province.

References

[1] D. L. Wong and C. M. Baker, “Pain in children: comparison of assessment scales,” *Pediatric Nursing*, vol. 14, no. 1, pp. 9–17, 1988.

[2] K. D. Craig, K. M. Prkachin, and R. V. E. Grunau, “The facial expression of pain,” in *Handbook of Pain Assessment*, D. C. Turk and R. Melzack, Eds., pp. 153–169, Guilford Press, New York, NY, USA, 2nd edition, 2001.

[3] D. Turk and R. Melzack, “The measurement of pain and the assessment of people experiencing pain,” in *Handbook of Pain Assessment*, D. C. Turk and R. Melzack, Eds., pp. 1–11, Guilford Press, New York, NY, USA, 2nd edition, 2001.

[4] A. B. Ashraf, S. Lucey, T. Chen et al., “The painful face—pain expression recognition using active appearance models,” in *Proceedings of the 9th International Conference on Multimodal Interfaces (ICMI ’07)*, pp. 9–14, ACM, Nagoya, Japan, November 2007.

[5] A. B. Ashraf, S. Lucey, J. F. Cohn et al., “The painful face—pain expression recognition using active appearance models,” *Image and Vision Computing*, vol. 27, no. 12, pp. 1788–1796, 2009.

[6] P. Lucey, J. F. Cohn, K. M. Prkachin, P. E. Solomon, S. Chew, and I. Matthews, “Painful monitoring: automatic pain monitoring using the UNBC-McMaster shoulder pain expression archive database,” *Image and Vision Computing*, vol. 30, no. 3, pp. 197–205, 2012.

[7] P. Lucey, J. Cohn, S. Lucey, I. Matthews, S. Sridharan, and K. M. Prkachin, “Automatically detecting pain using facial actions,” in *Proceedings of the 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops (ACII ’09)*, pp. 1–8, Amsterdam, The Netherlands, September 2009.

[8] P. Lucey, J. F. Cohn, I. Matthews et al., “Automatically detecting pain in video through facial action units,” *IEEE Transactions on Systems, Man, and Cybernetics B: Cybernetics*, vol. 41, no. 3, pp. 664–674, 2011.

[9] K. M. Prkachin, S. Berzinzs, and R. S. Mercer, “Encoding and decoding of pain expressions: a judgment study,” *Pain*, vol. 58, no. 2, pp. 253–259, 1994.

[10] K. M. Prkachin and P. E. Solomon, “The structure, reliability and validity of pain expression: evidence from patients with shoulder pain,” *Pain*, vol. 139, no. 2, pp. 267–274, 2008.

[11] M. Monwar, S. Rezaei, and K. Prkachin, “Eigenimage based pain expression recognition,” *IAENG International Journal of Applied Mathematics*, vol. 36, no. 2, pp. 1–6, 2007.

[12] W. Zhang and L. M. Xia, “Pain expression recognition based on SLPP and MKSVM,” *International Journal of Engineering and Manufacturing*, vol. 3, pp. 69–74, 2011.

[13] K. M. Prkachin, “The consistency of facial expressions of pain: a comparison across modalities,” *Pain*, vol. 51, no. 3, pp. 297–306, 1992.

[14] K. M. Prkachin and S. R. Mercer, “Pain expression in patients with shoulder pathology: validity, properties and relationship to sickness impact,” *Pain*, vol. 39, no. 3, pp. 257–265, 1989.

[15] J. F. Cohn, Z. Ambadar, and P. Ekman, “Observer-based measurement of facial expression with the facial action coding system,” in *The Handbook of Emotion Elicitation and Assessment*, pp. 203–221, Oxford University Press, New York, NY, USA, 2007.

[16] P. Ekman, W. V. Friesen, and J. C. Hager, *Facial Action Coding System: Research Nexus*, Network Research Information, Salt Lake City, Utah, USA, 2002.

[17] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1–3, pp. 185–204, 1981.

[18] A. A. Efros, A. C. Berg, G. Mori, and J. Malik, "Recognizing action at a distance," in *Proceedings of the 9th IEEE International Conference on Computer Vision*, vol. 2, pp. 726–733, IEEE, Nice, France, October 2003.

[19] Y. Zhang, R. Jin, and Z. H. Zhou, "Understanding bag-of-words model: a statistical framework," *International Journal of Machine Learning and Cybernetics*, vol. 1, no. 1–4, pp. 43–52, 2010.

[20] T. Li, T. Mei, I. S. Kweon, and X. Hua, "Contextual bag-of-words for visual categorization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 4, pp. 381–392, 2011.

[21] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1–3, pp. 185–204, 1981.

[22] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 50–57, ACM, New York, NY, USA, 1999.

[23] A. Bosch, A. Zisserman, and X. Munoz, "Scene classification via pLSA," in *Computer Vision—ECCV: Proceedings of the 9th European Conference on Computer Vision, Graz, Austria, May 7–13, 2006, Part IV*, vol. 4, pp. 517–530, Springer, Berlin, Germany, 2006.

[24] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman, "Learning object categories from Google's image search," in *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV '05)*, vol. 2, pp. 1816–1823, Beijing, China, October 2005.

[25] B. C. Russell, W. T. Freeman, A. A. Efros, J. Sivic, and A. Zisserman, "Using multiple segmentations to discover objects and their extent in image collections," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, pp. 1605–1614, June 2006.

[26] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman, "Discovering objects and their location in images," in *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV '05)*, vol. 1, pp. 370–377, Beijing, China, October 2005.

[27] J. Wang, P. Liu, M. F. H. She, A. Kouzani, and S. Nahavandi, "Supervised learning probabilistic latent semantic analysis for human motion analysis," *Neurocomputing*, vol. 100, pp. 134–143, 2013.