Research Article

Secure Access Control and Large Scale Robust Representation for Online Multimedia Event Detection

Changyu Liu,^{1,2} Bin Lu,³ and Huiling Li⁴

¹ School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China

² School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA

³ School of Computer Science, Wuyi University, Jiangmen 529020, China

⁴ State Key Laboratory of Pulp and Paper Engineering, South China University of Technology, Guangzhou 510640, China

Correspondence should be addressed to Bin Lu; lbscut@gmail.com

Received 2 April 2014; Accepted 30 June 2014; Published 22 July 2014

Academic Editor: Vincenzo Eramo

Copyright © 2014 Changyu Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We developed an online multimedia event detection (MED) system. However, there are a secure access control issue and a large scale robust representation issue when we want to integrate traditional event detection algorithms into the online environment. For the first issue, we proposed a tree proxy-based and service-oriented access control (TPSAC) model based on the traditional role based access control model. Verification experiments were conducted on the CloudSim simulation platform, and the results showed that the TPSAC model is suitable for the access control of dynamic online environments. For the second issue, inspired by the object-bank scene descriptor, we proposed a 1000-object-bank (1000OBK) event descriptor. Feature vectors of the 1000OBK were extracted from response pyramids of 1000 generic object detectors which were trained on standard annotated image datasets, such as the ImageNet dataset. A spatial bag of words tiling approach was then adopted to encode these feature vectors for bridging the gap between the objects and events. Furthermore, we performed experiments in the context of event classification on the challenging TRECVID MED 2012 dataset, and the results showed that the robust 1000OBK event descriptor outperforms the state-of-the-art approaches.

1. Introduction

As one of the most interesting aspects of multimedia content analysis, the multimedia event detection (MED) is becoming an important research area for computer vision in recent years. According to the definition by the National Institute of Standards and Technology (NIST) [1], an event (1) is a complex activity occurring at a specific place and time, (2) involves people interacting with other people and/or objects, (3) consists of a number of human actions, processes, and activities that are loosely or tightly organized and that have significant temporal and semantic relationships to the overarching activity, and (4) is directly observable. A MED task is to indicate whether an event is occurred in a specified test clip based on a standard event kit [1], which includes an event name, a textual definition, a textual explication with an attribute list, an evidential description, and a set of illustrative video examples. Although there are many other definitions

available, such as the MED definitions from the NIST, the research on the MED is still far from reaching its maturity. Most of the current researches are focused on specific areas, such as sports video [2], news video [3], and surveillance video [4]. These approaches do not perform well when used for the online or web based event detection due to two types of issues, which are the secure access control issue and the large scale robust representation issue. Thus, we developed an online multimedia event detection system, trying to provide general MED services.

The first issue is about how we can obtain a secure access control for the online multimedia event detection system. Compared with that of traditional distributed systems, it is a kind of service relationships between access control subjects and objects in the online multimedia event detection system. The service could establish, recombine, destruct, and even inherit efficiently to requested parameters which cannot be satisfied well by traditional access control models, such as

the role based access control model [5], where an authorization to the subject could not change parametrically and dynamically with its environment. In order to fuse dynamically by a recombination as well as a distribution of resources and services to enhance an access control over diversified applications of our online system, this paper proposed a tree proxy-based and service-oriented access control (TPSAC) models for the online multimedia event detection system and gave performance evaluations on the cloud simulation platform CloudSim [6].

The second issue is about how to obtain a large scale robust representation for the online multimedia event detection system. In order to perform a more generic and complicated online MED, a very large amount of labeled training videos are required for training event classifier. This would become a quite challenging task, especially when there are a large number of events involved. Because one event was usually composed of several objects that meet specific relationships, one solution to the challenge is to use event descriptor models that were learned from standard annotated image datasets, such as the ImageNet challenge dataset [7], to represent the events and to use the spatial bag of words tiling approach to represent their relationships which can bridge the gap between the objects and events. In this paper, we developed a 1000-object-bank (1000OBK) event descriptor for the online MED task. Inspired by the object bank scene descriptor of Li et al. [8, 9], the 1000OBK used 1000 pretrained object detector models to form a semantically-rich representation for MED. The individual object detectors in the 1000OBK are based on a mixture of two root components that were trained by the deformable part model of Felzenszwalb et al. [10, 11]. The outputs of detectors are transformed into a multiscale feature space by the max-pooling approach to generate 1000 response pyramids. While the original object bank [8, 9] offers a rich and high-level image representation, a severe issue lies in the curse of dimensionality when we want to make an extension to include a large number of detectors for the online MED. Note that, for each object, the original object bank [8, 9] uses a 252 dimension feature vector. Thus, the dimension of the feature vector is 252000 when there are 1000 object categories in the bank. The 1000OBK solves the problem by using a mean-pooling approach to sample interest patches on response pyramids of 1000 generic object detectors to generate a 252 dimension feature vector for each interest patch. The interest patches for each frame were then encoded by a spatial bag of words tiling approach to bridge the gap between the objects and the events. In the experiment and analysis section, we show that our 1000OBK is a robust event descriptor which achieves better performances than the state-of-the-art approaches on the TRECVID MED 2012 dataset.

There are three key components for our online multimedia event detection system, which are an access control component (component 1), an online event detection component (component 2), and an offline object detector training component (component 3), as shown in Figure 1. A user was firstly verified by the TPSAC model (see Section 3) in the component 1 to determine if he or she has the necessary permission to obtain an event detection service through

a dynamic permission tree. If the verification is successful, the user can proceed to use the pre-trained object detectors in Component 3 to extract and encode features (see Sections 4 and 5) of selected events for training and testing the event classifier (see Section 5), either by a manual mode where the user should choose positive videos as well as negative videos in the TRECVID MED 2012 dataset for a specific event through retrievals in our system, or by an automatic mode where the system would use the default classifier that be trained previously, as shown in Component 2 of Figure 1. These object detectors of 1000OBK event descriptor were trained offline on the ImageNet challenge 2012 dataset by a mixture model of the state-of-the-art deformable part model (DPMs) object detector (see Section 4) from [10, 11], as shown in Component 3 of Figure 1. Finally, the classification results for selected events and video clips were shown graphically in our web system.

2. Related Work

The study on access control has always been one of the hottest areas in the field of information security. One of its key goals is to prevent unauthorized users from modifying, deleting, and revealing illegally the protected data. The traditional access control models can be classified into a matrix model, a safety classification model, an authorization rule model, and an XML model and can be also divided into a discretionary access control model, a mandatory access control model, and a role based access control (RBAC) model. By introducing a role between the subject and the privilege, the RBAC model separates logically users from their privileges and achieves greater manageability and security. Many researchers made extensive studies on the RBAC since it was firstly proposed at 1992. With the development of the service oriented technology and the grid technology, the study on access control achieved a great progress in recent years. In the service oriented aspect, the paper [12] provided a service oriented access control model to solve the safety requirement issue that the authentication and authorization mechanism of middleware could not meet the resource sharing and the service requesting between many different nodes and systems. A workflow based and services oriented role based access control (WSRBAC) model was put forward in paper [13], by introducing a role actor to ensure that the access control strategy could change rapidly with the status of workflows and services between roles and users in service-oriented architecture (SOA). In the grid technology aspect, a community authorization service (CAS) was presented to settle basically the authorization management of global as well as local grids in paper [14]. But the CAS didn't take into account the dynamic grid environment due to its static authorization mechanism [15]. In this paper, we proposed a tree proxy-based and service-oriented access control (TPSAC) model for the online multimedia event detection system.

In computer vision, the MED is becoming a hot research area. There are several key components in one MED task, which are the training and testing datasets, the raw features,

FIGURE 1: Online multimedia event detection framework.

the feature encoding and pooling methods, and the classification methods. Firstly, the datasets play a fundamental role in MED tasks, which include the image datasets such as Caltech 101 [16], Caltech 256 [17], MIT dataset series [18], Label Me [19], ImageNet [7] and Pascal VOC [20], and the video datasets, such as HMDB 51 actions [21], KTH human action [22], UCF 50 [23], UCF 101 [24], UCF sports [25], hollywood human action [26], UIUC sports event [27], and TRECVID MED [1]. In this paper, we adopted the TRECVID MED 2012 dataset that is provided by the NIST. Secondly, there are four kinds of raw features, which are the visual feature, the audio feature, the text feature, and the concepts based feature. There are many popular visual features for representing multimedia contents, which include SIFT [28], color SIFT (CSIFT) [29], transformed color histogram (TCH) [30], motion SIFT (MoSIFT) [31], space-time interest point (STIP) [32], dense trajectory (DTF) [33], object bank [8, 9], and action bank [34]. Thirdly, the sparse coding [35], the fisher vector coding [36], and the spatial bag of visual words [37] are three main feature encoding and pooling approaches for representing multimedia contents. Last, there are many popular classifiers available for the MED task, such as support vector Machine (SVM) [38], Kernel ridge regression (KRR) [39], Kernel logistic regression (KLR) [40], adaptive boosting (AdaBoost) [41], and many modified versions of them. There are also many annual competitions for the object and event detection, such as the ImageNet challenge [7], the PASCAL VOC challenge [20], and the TRECVID workshop [1]. Supported by the NIST, the TRECVID MED task has not only attracted quickly many top research groups, such as the CMU E-Lamp research group which participated in the TRECVID MED task for the past six years and proposed many effective event detection approaches, but also became the popular testing bed for the MED.

Most of previous researches on MED have focused on detecting simple events for specific applications. Sadlier and O'Connor [42] proposed an audio-visual feature based framework for event detection in broadcast video of multiple different field sports. Zhang and Chang [43] developed a system for baseball video event detection and summarization using superimposed caption text detection and recognition. Xu et al. [44] proposed an approach for event detection from the live sports game using webcasting text and broadcast video. Li [45] proposed a probabilistic model to incorporate both content and time information in a unified framework to determine the approximate number of events from

the articles count-time distribution. These researches have contributed to MED in different ways. However, no work has been put forward on how to design a generic and robust online multimedia event detection system. In this paper, we give several solutions about these.

3. The TPSAC Model

3.1. Basic Concepts

Definition 1 (Permission Package Set, PPS). The permission package is a uniform storage format for meta permission sets of various roles and a dynamic permission attribute of the proxy-based tree *AccBTree*. The data structure of PPS package consists of three sequential parts, as shown in Figure 2. The first part acts as a type identifier. The second part denotes a permission bitmap. The third part is a value sequence, which contains a weight and a lifetime, of the meta permission set which is in the same order with that of the second part bitmap.

Definition 2 (Service, S). The *service* is an important concept in the SOA architecture. It is a kind of logic units, which is able to provide specific and reusable business functions to some other application systems. Loose coupling is one of the main service attributes.

Definition 3 (Proxy-based Tree Set, T). The proxy-based tree *AccBTree*, which encloses the role based meta permission set, is a dynamic object that was generated when a role was activated by a user. It can be described by a quadruple $\langle R, PPS, ST, Lifetime \rangle$, where R is the activated role set, ST is the tree status set, and $Lifetime$ is the tree lifetime.

Definition 4 (Meta Permission Sets, MPS). Because the system resource $OBS = FO \cup DO$, where FO is the function object and DO is the data object, the total meta permission set $MPS = \{(obj, op, i) \mid obj \in OBS, op \in OPS, i \in N\}$, where $MetaPerm = (obj, op, i)$ is a meta permission and an elementary unit during authorization, OPS is the operation set, and I is the lifetime or the weight that reflects the credibility and reliability of its role. Involved already the information of op and i in the PPS package, the meta permission set can be also expressed as $MPS = \{(obj, pps) \mid obj \in OBS, pps \in PPS\}$. After getting rid of the overlapping permission, the total valid permission set $P = \{p \mid \forall p_i \in MPS \wedge \forall p_j \in MPS \wedge i \neq j \rightarrow p_i \cap p_j = \emptyset\}$.

FIGURE 2: The data structure of PPS package.

Definition 5 (User, Permission, and Tree Assignment). Let $DU = \{u \mid u \in U\}$, $EU = \{r \mid r \in R\}$, $DP = \{p \mid p \in MPS\}$, $EP = \{r \mid r \in R\}$, $DT = \{t \mid t \in T\}$ and $ET = \{s \mid s \in S\}$. Then we define $UA = \{(u, r)\}$ as user assignment relationship on $DU \rightarrow EU$, $PA = \{(p, r)\}$ as a permission assignment relationship on $DP \rightarrow EP$, and $TA = \{(t, s)\}$ as a tree assignment relationship on $DT \rightarrow ET$.

3.2. Model Definitions. In order to satisfy fully the integrity constraints of task flows and enhance system security, the user permission is not only subject to the performed role but also subject to the actual task status. When a role is activated by one user, it activates accordingly a dynamic and temporary proxy-based tree *AccBTree*. As an atomic function in SOA, the *AccBTree* could be created parametrically in a task and be released after a task. Thus, the system could recycle the previous assigned user permission.

Definition 6 (TPSAC Model). $TPSAC = \langle U, R, MP, T, S \rangle$, where U is the user set, R is the role set, MP is the meta permission set, T is the Proxy-based tree set, S is the service set, UA is the user assignment, PA is the permission assignment, and TA is the tree assignment. Then RR is the compound and inherited role relation, and SR is the constrained service relation. The process when a role obtains a permission service is also the one when a proxy-based tree is spanned with its status changing. The status and the constraint condition of this tree determine whether a service is allowed. There are three phases in TPSAC, which are a global static permission assignment, a proxy-based tree spanning, and a dynamic permission adjustment.

3.3. Model Algorithms. The TPSAC based service tree is actually a resource tree with permission information. So the TPSAC tree is a kind of multiway tree. In order to locate and create conveniently a tree, this paper adopts a child-sibling list to transform the multiway tree to a binary permission tree. The basic classes we used for access control can be defined in Algorithm 2.

In order to generate a spanning tree, a depth-first traversal algorithm was adopted recursively in the function *TAccessBTree::CreateTree (TNode * root)*, where there were two steps in the spanning process of a binary permission tree (see Algorithm 1), which were a creation of a multiway permission tree and a transform from a multiway tree to a binary tree.

4. The 1000OBK Event Descriptor

4.1. The Feature Vector. Based on the original object bank [8, 9], we proposed a 1000-object-bank event descriptor, using the following steps.

(1) *Getting 3D Interest Points by the Harris3D Corner Detector.* We used the Harris3D corner detector [32] for extracting 3D interest points from key frames of given video clips. The task of the Harris3D corner detector is to find 3D interest points which have significant changes in spatial temporal directions.

(2) *Getting the HOG Feature Pyramid.* We resized firstly the video frames so that the width is 320 pixels and the height is determined by the aspect ratio and then used six scales, which are 1, 0.707, 0.5, 0.3535, 0.25, and 0.17677, to have a further resizing on those video frames to generate an image pyramid for each frame. For HOG feature, we set the bin size to be 9 in $0^\circ-360^\circ$, the cell type to be rectangular, the cell size to be 8×8 pixels, the block type to be R-HOG, the block size to be 2×2 cells, and the block spacing stride size to be 8 pixels.

(3) *Getting the Response Pyramid.* The state-of-the-art deformable part model (DPMs) object detector from [10, 11] was used. In our model, we used a mixture model of DPMs with two root components as our object detector. Each of these components works on 6 scales of a HOG pyramid. As a result, there are 12 response maps in the final response pyramid.

(4) *Generating the 1000OBK Feature Vector.* We constructed our 1000OBK feature vector from responses of many object detectors on the HOG feature pyramid we obtained before. For each of the interest points that obtained by the Harris3D corner detector on the key frames, we extracted 12 response patches with each patch from one 8×8 dimension subwindow that located on one of the 12 response maps. Then, a three-level spatial pyramid was applied to each of the 8×8 dimension response patches, with 1×1 grid on the first level, 2×2 grids on the second level, and 4×4 grids on the third level. Each grid was sampled by the max-pooling method from its corresponding areas of the response patch. So, we got a $21 (1 \times 1 + 2 \times 2 + 4 \times 4)$ dimension feature vector for every one of the 12 response patches, resulting in a total of $12 \times 21 = 252$ dimension vector for representing one interest patch from one object detector. A mean-pooling was then performed on each of the 252 dimension among the 1000 types of 252 dimension feature vector with each type for one object detector. Hence, the final dimension of the 1000OBK feature vector was 252, which was used to represent one patch of the interest point.

4.2. Object Detector Models. It can be observed from above steps that the object detector plays an important role for generating our 1000OBK event descriptor. In this paper, we used the deformable part model (DPMs) [10, 11] to train our 1000 objects detector models from the ImageNet challenge 2012 dataset, which provides the definition of 1000 objects along with 732-1300 training images, 50 validation images, and 100 test images for each of the 1000 object categories.


```

(1) CreateTree(TNode *root) <
(2) TNode *pNode, *tNode;
(3) m_pAdo: (r1, r2, ..., rn)
    = {ri | Parent_ID(ri) = ID(Root) };
(4) //Use m_pAdo to get the sub-tree of node root
(5) if (m_pAdo → RecordCount > 0){
(6)   m_pAdo → First();
(7)   for (int i = 0; i < m_pAdo → RecordCount;
        ++i){
(8)     tNode = new TNode();
(9)     tNode → id = RES_ID(m_pAdo);
(10)    tNode → pps = PPS(m_pAdo);
(11)    /****T_Rule: Tm → Tb****/
(12)    if (i == 0) root → firstchild = tNode;
(13)    else pNode → nextsibling = tNode;
(14)    pNode = tNode;
(15)    CreateTree(tNode); //Recursive Invocation
(16)    /****T_Rule: Tm → Tb****/
(17)    m_pAdo → Next();
(18)  }
(19) >

```

ALGORITHM 1: The spanning binary permission tree.

```

class TAccessBTree {
private: static TAccessBTree *m_pBTree;
        int m_rID; TNode *m_pRoot;
        bool CreateTree(TNode *root);
        bool SetState(int state_id); ...
public: static TAccessBTree*
        getInstance(int, char*);
        bool LocateRes(int, PPS&); ...
};
class TNode {public: int id; PPS* pps; TNode
            *firstchild; TNode *nextsibling; ...};

```

ALGORITHM 2: Basic classes for the TPSAC model.

Because there are no bounding boxes for the test images in the ImageNet challenge 2012 dataset, we split its training dataset into a training dataset where we chose 100 positive images and 400 negative images for each object category and a testing dataset where we chose randomly 1300 images for each object category. We used the trained 1000 object models as the root filters of our 1000OBK event descriptor.

5. Event Representation

5.1. Event Feature Extraction. Before feature extraction, we resized the video frames width to 320 pixels with their height to a size that determined by the aspect ratio and selected the key frames with a shot boundary detection algorithm. Specifically, the target video was firstly sampled on every 5 frames to get its color histogram and a histogram subtraction was conducted between every two consecutive sampled frames. Then, depending on whether the subtraction value was larger than a predefined threshold, a decision

of whether a frame was a shot boundary was made. Last, the frame in the middle of two shot boundary frames was taken as the key frame of this shot. After the key frames selection, the Harris3D corner detector [32] was used to detect interest points for each of the selected key frames. Then feature extraction was conducted on the frame patches that were near the interest points. In order to evaluate the event detection performance by our 1000OBK event descriptor, we also extracted object bank (OBK) feature [8, 9], MoSIFT feature [31], and Kanade-Lucas-Tomasi (KLT) trajectories feature [46, 47].

5.2. Encoding and Classification. In order to bridge the gap between the objects and events, a spatial bag of words tiling was adopted as a feature encoding method after the raw features extraction in our experiment. For each clip, we got four kinds of feature vectors, which are 1000OBK, OBK, MoSIFT, and KLT trajectories, for the detected interest points on those key frames. Each kind of features was clustered

by k -means algorithm. We used 50 clips for each event and 1000 clips for total 20 events, to generate four codebooks. Each of them had a size of 4096. After a generation of these codebooks, we can describe a key frame by such a 4096 dimensions feature vector. For each interest point of this key frame, we find its nearest cluster center and add one to that dimension. Normalization was performed then after the calculation for all the interest points in one key frame. We also performed a tilting approach with patterns of 1×1 , 2×2 and 1×3 , resulting in a $32768 = (1 * 1 + 2 * 2 + 1 * 3) \times 4096$ dimension feature vector for each key frame.

For events classification, we used the support vector machine (SVM) classifier, since it has been widely used by several research groups for TRECVID MED and has shown its robustness [48, 49]. Given a training set $\{(x_i, y_i)\}_{i=1}^n$ with input data $x_i \in R^n$ and class labels $y_i \in \{-1, 1\}$, the SVM tries to solve the following optimization problem:

$$\min_{w, b, \xi} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i \quad (1)$$

Subject to: $y_i (w^T \phi(x_i) + b) \geq 1 - \xi_i,$

$\xi_i \geq 0, \quad i = 1, \dots, n,$

where $\phi(x_i)$ is the function for mapping x_i into a higher dimensional space, $C > 0$ is the penalty parameter, and ξ_i is the nonnegative slack variables. Then, the decision function of SVM is as follows:

$$\text{sign} \left(\sum_{i=1}^n a_i y_i K(x, x_i) + b \right), \quad (2)$$

where $K(x, x_i) = \phi(x_i)^T \phi(x_j)$ is called the kernel function, a_i is the i th Lagrange multiplier which can be obtained by solving the dual form of the primal optimization problem, and N_{SV} is the number of support vectors which have $a_i \neq 0$. For SVM, we used the exponential Chi Square ($\exp\text{-}\chi^2$) Kernel [50, 51], which is also an additive kernel. It can be defined as follows, where γ is a kernel controlling parameter:

$$K(u, v) = \exp \left\{ -\gamma \sum_{i=1}^n \frac{(u_i - v_i)^2}{u_i + v_i} \right\}. \quad (3)$$

6. Experiment and Analysis

6.1. Dataset. We used the TRECVID MED 2012 development dataset that was provided by NIST for our experiments. The dataset had a total of 42466 clips and 20 prespecified events, which are E006: birthday party, E007: changing a vehicle tire, E008: flash mob gathering, E009: getting a vehicle unstuck, E010: grooming an animal, E011: making a sandwich, E012: parade, E013: parkour, E014: repairing an appliance, E015: working on a sewing project, E021: attempting a bike trick, E022: cleaning an appliance, E023: dog show, E024: giving directions to a location, E025: marriage proposal, E026: renovating a home, E027: rock climbing, E028: town hall meeting, E029: winning a race without a vehicle, and

E030: working on a metal crafts project. Figure 3 showed four sample frames for each of the 20 events in TRECVID MED 2012 dataset. Then, we split this dataset into a training set and a testing set. For the training set, we select randomly 20 positive clips and 100 negative clips for each of the 20 events. The other 40066 clips were used as the testing set. Then, we run our experiments three times with a different combination of positive clips as well as negative clips on the Blacklight Linux server at the Pittsburgh Supercomputing Center and reported the average results.

6.2. Evaluation Metrics. There are many evaluation metrics available for the MED. In our experiments three groups of them were used, which are the average precision (AP) and the mean average precision (MAP), the probability of missed detection (P_{Miss}) and the mean probability of missed detection (MP_{Miss}), and the minimum normalized detection cost (MinNDC) and the mean minimum normalized detection cost (MMinNDC). They are all the official evaluation metrics used by NIST for TRECVID MED evaluation [52, 53]. The metrics in the first group are defined in (4) and (5) separately, where N_p is the number of total positive clips for a specific event, N_{tp} is the number of true positive clips, N_{fp} is the number of false positive clips, i is for one event, n is the number of total events which is set to be 20 in this paper, and AP is the average precision and MAP is the mean average precision. A higher AP or MAP denotes a better performance. Consider

$$\text{AP} = \frac{1}{N_p} \sum_{N_{tp}=1}^{N_p} \frac{N_{tp}}{N_{tp} + N_{fp}} \quad (4)$$

$$\text{MAP} = \frac{1}{n} \sum_{i=1}^n \text{AP}_i. \quad (5)$$

The metrics in the second group are defined in (6) and (7) separately, where N_{Miss} is the number of positive clips for a specific event that is less than the detection threshold, N_p is the number of positive clips for a specific event, P_{Miss} is the probability of missed detection, and MP_{Miss} is the mean probability of missed detection. A lower P_{Miss} or MP_{Miss} denotes a better performance. Consider

$$P_{\text{Miss}} = \frac{N_{\text{Miss}}}{N_p} \quad (6)$$

$$\text{MP}_{\text{Miss}} = \frac{1}{n} \sum_{i=1}^n P_{\text{Miss}_i}. \quad (7)$$

The metrics in the third group are defined in (8) and (9) separately, where $C_{\text{Miss}} = 80$ is the cost for missed detection, P_{Miss} is the probability of missed detection, $C_{\text{FA}} = 1$ is the cost for false alarm, $P_T = 0.001$ is a constant defining the priori rate of event instances. P_{FA} is the probability of false alarm and can be calculated by $P_{\text{FA}} = N_{\text{FA}} / (N_T - N_p)$, where N_{FA} is the number of nonpositive clips for a specific event, N_T is the total number of clips, N_p is the number of positive clips for a specific event, MinNDC is the minimum

FIGURE 3: Sample frames of 20 events in TRECVID MED 2012 dataset.

normalized detection cost, and MMinNDC is the mean minimum normalized detection cost. A lower MinNDC or MMinNDC denotes a better performance. Consider

$$\text{MinNDC} = \frac{C_{\text{Miss}} \times P_{\text{Miss}} \times P_T + C_{\text{FA}} \times P_{\text{FA}} \times (1 - P_T)}{\min(C_{\text{Miss}} \times P_T, C_{\text{Miss}} \times (1 - P_T))} \quad (8)$$

$$\text{MMinNDC} = \frac{1}{n} \sum_{i=1}^n \text{MinNDC}_i. \quad (9)$$

6.3. Evaluating the TPSAC Model. For evaluating the performance of the TPSAC model, this paper adopted a widely used cloud simulation platform CloudSim [6], which was developed by the University of Melbourne. In order to make the experimental environment to be close to the real cloud computing scene, we firstly introduced 13 federate models from paper [54]. Then, we set up a service interactive scene TPSAC simulation in the CloudSim. After that, we configured the corresponding service provider (SP) and service consumer (SC) as $\text{SP} = \{\text{SCUT_CCNL_Srv1}, \text{CMU_CQ_Srv1}, \text{CMU_CQ_Srv2}\}$ and $\text{SC} = \{\text{USER1}, \text{USER2}, \text{USER3}, \text{USER4}, \text{USER5}\}$ and then chose the total SP entities of 1000 with 80%, 10%, and 10%, respectively, and the total SC entities of 2000 with 50%, 20%, 10%, 10%, and 10%, respectively. All these codes were executed on an Ubuntu 12.04 LTS desktop. The other experiment configurations were similar as [55, 56].

Two experiments were conducted to evaluate the TPSAC performance, which are the impact of the bitmap length and meta permission length on the TPSAC performance and the impact of the thread pool size on the TPSAC performance.

6.3.1. Varying the PPS's Bitmap Length and the Meta Permission Length. The goal of the first experiment was to test the relation among the TPSAC processing time T , the PPS bitmap length L_{Bitmap} , and the PPS meta permission length L_{MetaPerm} . Hence, we fixed the thread pool size at $L_{\text{Pool}} = 32$ and set $L_{\text{Bitmap}} = 8 * i$, $L_{\text{MetaPerm}} = 4 * j$, where $1 \leq i \leq 9$ and $1 \leq j \leq 4$. Then, we plotted the T value for each configuration, as shown in Figure 4.

It can be seen that (1) for the impact from the PPS bitmap length, the T value grew slowly when $L_{\text{Bitmap}} \leq 32$ and rapidly when $L_{\text{Bitmap}} > 32$ for all kinds of meta permission lengths and (2) for the impact from the meta permission length, the T value increased more sharply when $L_{\text{MetaPerm}} = 4$ or 8 than $L_{\text{MetaPerm}} = 12$ or 16. So we chose $L_{\text{Bitmap}} = 32$ and $L_{\text{MetaPerm}} = 8$.

6.3.2. Varying the Thread Pool Size. The goal of the second experiment was to test the relation between the T value and the thread pool size L_{Pool} . Hence, we fixed $L_{\text{Bitmap}} = 32$, $L_{\text{MetaPerm}} = 8$ and set $L_{\text{Pool}} = 2^k$, where $1 \leq k \leq 9$. Then, we also plotted the T value for each configuration. In order

FIGURE 4: Processing time with respect to PPS bitmap length.

FIGURE 5: Processing time with respect to thread pool size.

to analyze conveniently the influence from using a thread pool, a T value without a thread pool was also plotted. The experiment results were shown in Figure 5.

It can be concluded that (1) the usage of a thread pool could improve efficiently the performance of the TPSAC server, especially when there are a large number of concurrent requests from SP and SC and (2) the change of the thread pool size has a significant impact on the specific server. If the size is too small, an access control task shall be not treated timely. Otherwise, it will lead to a larger cost for the threads synchronization and switch; that is, the optimization problem of the thread pool size should be taken into account at the

TABLE 1: Comparison of different descriptors for average performance on TRECVID MED 2012 dataset.

Descriptors	MAP	MP _{Miss}	MMinNDC
1000OBK	0.2507	0.3695	0.6078
Object bank [8, 9]	0.1884	0.3963	0.6635
MoSIFT [31]	0.2264	0.3778	0.6221
KLT trajectories [46, 47]	0.1987	0.4270	0.6944

beginning. So, we preset the $L.Pool$ to be 32, which could then be changed with its dynamic environment.

6.4. Comparing 1000OBK with Baseline Approaches. We compared our 1000OBK with some of the best descriptors, which are object bank [8, 9], Kanade-Lucas-Tomasi (KLT) trajectories [46, 47], and MoSIFT [31]. For object bank [8, 9], we used the code at [57]. The object bank [8, 9] used two state-of-the-art detectors which are the latent SVM object detectors and a texture classifier by Hoiem [58] to generate a 44604 (177 (no. of objects) $\times 12$ (no. of scales) $\times (1^2 + 2^2 + 4^2)$ (no. of grids)) feature vector for each frame. For Kanade-Lucas-Tomasi (KLT) trajectories, we used the standard KLT tracker [46, 47]. Specifically, there are about 100 interest points are detected in each frame and these points are tracked through the video clip for 15 frames, which were described by HOG, HOF and MBH. For MoSIFT, we used the implementation of paper [31] with default parameter settings. The MoSIFT descriptor combines both local appearance which is same as the original SIFT and local motion which is computed through an optical flow pyramid to extend the SIFT descriptor to the third dimension. As shown in paper [31], the MoSIFT outperformed the-state-of-the-art approaches for human action recognition in a large and complex real world surveillance dataset.

Then, we used the LibSVM library [38] for SVM and tune the C from $2^{-5}, 2^{-3}, \dots, 2^{15}$ and the γ from $2^{-15}, 2^{-13}, \dots, 2^3$ under 20-fold cross validation with each fold for one of the 20 events. We ran our experiments three times with different combinations of positive clips as well as negative clips on the Blacklight Linux server at the Pittsburgh Supercomputing Center and reported the average results. Table 1 showed a comparison of these descriptors for the average MED performance. Note that a higher MAP indicates a better performance, while a lower MP_{Miss} or $MMinNDC$ indicates a better performance. Best results for each evaluation metric are highlighted. We can observe that our 1000OBK is the best descriptor, the MoSIFT is the second best descriptor, and the KLT trajectories or the object bank is the worst descriptor for all of the three evaluation metrics. Next, we made a MED performance comparison of these four descriptors on each of the 20 events, as shown in Figure 6. It can be seen that our 1000OBK event descriptor obtained the best performance on 10 events for average precision, on 10 events for P_{Miss} , and on 9 events for MinNDC. Note that there are total 20 events. The comparisons here demonstrated clearly that our 1000OBK event descriptor outperforms object bank, MoSIFT, and KLT trajectories for MED.

FIGURE 6: Comparison of MED performance among 1000OBK, object bank, MoSIFT, and KLT on the 20 events of TRECVID MED 2012 dataset, with (a–b) for AP, (c–d) for P_{Miss} , and (e–f) for MinNDC.

6.5. Evaluating the 1000OBK Event Descriptor. In this subsection, we evaluate the robustness and the event classification performance of our 1000OBK event descriptor.

6.5.1. Robustness with Respect to Training Size. We investigate the robustness of our 1000OBK event descriptor with respect to the training size of object detectors in this experiment. We train the 1000 object models by using multiple sizes of training examples, ranging from 25%, 50%, and 75% to 100% of the full ImageNet challenge 2012 training data, where we chose 100 positive images and 400 negative images as the full training data for each object category. It can be seen from the Figure 7 that there is a moderate drop for MMinNDC or MP_{Miss} and a slight increase for MAP, when the number of training size decreases from 100% to 25%. Note that a higher

MAP indicates a better performance and a lower MMinNDC or MP_{Miss} indicates a better performance. The results suggested that the event classification performance dropped slightly when the training size decreased tremendously. Thus, the 1000OBK is a robust event descriptor with respect to the training size.

6.5.2. Varying the Bank Size. One of the major contributions in this work is the extension of the object bank [8, 9] to adapt to different event classification tasks. However, it is not given that a larger bank size improves always the event classification performance. The curse of dimensionality may prevent this intuition. In order to evaluate the impact of bank size on MED results, we conducted an experiment that varied the size of the bank from 177 detectors which constituted the original

FIGURE 7: Event classification performance with respect to the training size.

FIGURE 8: Event classification performance with respect to the bank size.

object bank [8, 9] to the 1000 detectors which constituted our 1000OBK. For each different size k , we chose randomly k detectors from the 1000 trained models to form a new k -OBK, where $177 \leq k \leq 1000$. Then, we perform a full leave-one-out cross-validation on the TRECVID MED 2012 dataset. The results are shown in Figure 8. As we expected, the bigger bank size does indeed performed better. Specifically, we are able to achieve the MED results of MoSIFT [31] with a bank size of 500, the MED results of KLT trajectories [46, 47] with a bank size of 200, and the MED results of object bank [8, 9] with a bank size of 177, respectively.

Why could we improve the MED results a lot by varying the bank size from 177 to 1000? One of the reasons is that a

given event is described by a mixture of scores from different object detectors and a spatial relationship of these objects. These events involve usually many different kinds of object categories. Furthermore, as a nongroup-sparsity classifier, the χ^2 -SVM requires using more nonzero weights of objects for distinguishing different types of events. With this evidence, it is not surprising to notice the improvement of event classification performance with a larger bank size.

7. Conclusion

In this paper, we provided solutions to two key issues, which are a secure access control issue and a large scale robust representation issue, for a more generic and complicated online multimedia event detection. For the first issue, the paper proposed a tree proxy-based and service-oriented access control (TPSAC) model. For the second issue, the paper proposed a 1000-object-bank (1000OBK) event descriptor, a robust method for carrying out high-level event recognition on the TRECVID MED 2012 dataset. The 1000OBK leverages on the fact that a large number of object detectors, when pooled appropriately, can provide high-level semantically rich features that are superior to low-level features for MED. We also performed several experiments to demonstrate the effectiveness of our approaches for the online MED. In the future, we will use the 1000OBK representation in other useful vision applications, such as action recognition.

Conflict of Interests

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

Acknowledgments

This paper was supported by the Doctor Startup Foundation of Wuyi University under Grant no. 2014BS07 and the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center Contract no. D11PC20068. The model training and event testing work were performed on the Blacklight at the Pittsburgh Supercomputing Center (PSC). The authors would like to thank the PSC for providing them with a large scale parallel computing environment and thank anonymous reviewers for helpful comments.

References

- [1] P. Over, J. Fiscus, G. Sanders et al., "TRECVID 2012—an overview of the goals, tasks, data, evaluation mechanisms, and metrics," in *Proceedings of the TRECVID Workshop*, 2012.
- [2] C. Xu, Y. Zhang, G. Zhu, Y. Rui, H. Lu, and Q. Huang, "Using webcast text for semantic event detection in broadcast sports video," *IEEE Transactions on Multimedia*, vol. 10, no. 7, pp. 1342–1355, 2008.
- [3] F. Wang, Y. Jiang, and C. Ngo, "Video event detection using motion relativity and visual relatedness," in *Proceedings of the 16th ACM International Conference on Multimedia (MM '08)*, pp. 239–248, October 2008.

- [4] S. Oh, A. Hoogs, A. Perera et al., "A large-scale benchmark dataset for event recognition in surveillance video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 3153–3160, June 2011.
- [5] R. S. Sandhu, E. J. Coynek, H. L. Feinstein, and C. E. Youmank, "Role-based access control models," *IEEE Computer*, vol. 29, no. 2, pp. 38–47, 1996.
- [6] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. de Rose, and R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software—Practice and Experience*, vol. 41, no. 1, pp. 23–50, 2011.
- [7] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, "Imagenet: a large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '09)*, pp. 248–255, 2009.
- [8] L. J. Li, H. Su, E. P. Xing, and F. F. Li, "Object bank: a high-level image representation for scene classification semantic feature sparsification," *NIPS Proceedings*, vol. 2, no. 3, p. 5, 2010.
- [9] L. J. Li, H. Su, Y. Lim, and L. Fei-Fei, "Objects as attributes for scene classification," in *Trends and Topics in Computer Vision*, pp. 57–69, Springer, Berlin, Germany, 2012.
- [10] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.
- [11] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, June 2008.
- [12] L. Zhu, M. Zhou, T. Liu, and H. Mei, "Permission management model in service-oriented architecture," *Chinese Journal of Computers*, vol. 28, no. 4, pp. 678–685, 2005.
- [13] F. Xu, H. G. Lai, H. Huang, and L. Xie, "Service-oriented role-based access control," *Chinese Journal of Computers*, vol. 28, no. 4, pp. 686–693, 2005.
- [14] L. Pearlman, C. Kesselman, V. Welch et al., "The community authorization service: status and future," in *Proceedings of the Computing in High Energy Physics Conference (CHEP '03)*, pp. 1–9, 2003.
- [15] H. X. Cai, T. Yu, and M. L. Fang, "Access control of manufacturing grid," *Computer Integrated Manufacturing Systems*, vol. 13, no. 4, pp. 716–720, 2007.
- [16] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories," *Computer Vision and Image Understanding*, vol. 106, no. 1, pp. 59–70, 2007.
- [17] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," Tech. Rep. 7694, California Institute of Technology, 2007.
- [18] A. Quattoni and A. Torralba, "Recognizing indoor scenes," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops '09)*, pp. 413–420, June 2009.
- [19] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "LabelMe: a database and web-based tool for image annotation," *International Journal of Computer Vision*, vol. 77, no. 1–3, pp. 157–173, 2008.
- [20] M. Everingham, L. van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.
- [21] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: a large video database for human motion recognition," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 2556–2563, November 2011.
- [22] C. Schödl, I. Laptev, and B. Caputo, "Recognizing human actions: a local SVM approach," in *Proceedings of the 17th International Conference on Pattern Recognition (ICPR '04)*, pp. 32–36, August 2004.
- [23] K. K. Reddy and M. Shah, "Recognizing 50 human action categories of web videos," *Machine Vision and Applications*, vol. 24, no. 5, pp. 971–981, 2013.
- [24] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: a dataset of 101 human action classes from videos in the wild," *CRCV-TR-12-01*, 2012.
- [25] M. D. Rodriguez, J. Ahmed, and M. Shah, "Action MACH: a spatio-temporal maximum average correlation height filter for action recognition," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, Anchorage, Alaska, USA, June 2008.
- [26] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, June 2008.
- [27] L. J. Li and L. Fei-Fei, "What, where and who? Classifying events by scene and object recognition," in *Proceedings of the IEEE 11th International Conference on Computer Vision (ICCV '07)*, pp. 1–8, Rio de Janeiro, Brazil, October 2007.
- [28] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [29] K. Van De Sande, T. Gevers, and C. Snoek, "Evaluating color descriptors for object and scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1582–1596, 2010.
- [30] G. J. Burghouts and J. Geusebroek, "Performance evaluation of local colour invariants," *Computer Vision and Image Understanding*, vol. 113, no. 1, pp. 48–62, 2009.
- [31] M. Y. Chen and A. Hauptmann, "MoSIFT: recognizing human actions in surveillance videos," CMU CS-09-161, Carnegie Mellon University, 2009.
- [32] I. Laptev, "On space-time interest points," *International Journal of Computer Vision*, vol. 64, no. 2–3, pp. 107–123, 2005.
- [33] H. Wang, A. Kläser, C. Schmid, and C. L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *International Journal of Computer Vision*, vol. 103, no. 1, pp. 60–79, 2013.
- [34] S. Sadanand and J. J. Corso, "Action bank: a high-level representation of activity in video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 1234–1241, June 2012.
- [35] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '09)*, pp. 1794–1801, June 2009.
- [36] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *Proceedings of the 11th European Conference on Computer Vision (ECCV '10)*, vol. 6314 of *Lecture Notes in Computer Science*, pp. 143–156, Crete, Greece, 2010.

- [37] 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.
- [38] 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.
- [39] S. An, W. Liu, and S. Venkatesh, "Fast cross-validation algorithms for least squares support vector machine and kernel ridge regression," *Pattern Recognition*, vol. 40, no. 8, pp. 2154–2162, 2007.
- [40] B. Krishnapuram, L. Carin, M. A. T. Figueiredo, and A. J. Hartemink, "Sparse multinomial logistic regression: fast algorithms and generalization bounds," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 957–968, 2005.
- [41] G. Rätsch, T. Onoda, and K. R. Müller, "Soft margins for AdaBoost," *Machine Learning*, vol. 42, no. 3, pp. 287–320, 2001.
- [42] D. A. Sadlier and N. E. O'Connor, "Event detection in field sports video using audio-visual features and a support vector machine," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 10, pp. 1225–1233, 2005.
- [43] D. Zhang and S. Chang, "Event detection in baseball video using superimposed caption recognition," in *Proceedings of the 10th International Conference of Multimedia*, pp. 315–318, December 2002.
- [44] C. Xu, J. Wang, K. Wan, Y. Li, and L. Duan, "Live sports event detection based on broadcast video and web-casting text," in *Proceedings of the 14th Annual ACM International Conference on Multimedia (MM '06)*, pp. 221–230, October 2006.
- [45] Z. Li, "A probabilistic model for retrospective news event detection," in *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '05)*, 2005.
- [46] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the International Joint Conferences on Artificial Intelligence*, 1981.
- [47] R. Messing, C. Pal, and H. Kautz, "Activity recognition using the velocity histories of tracked keypoints," in *Proceedings of the 12th International Conference on Computer Vision (ICCV '09)*, pp. 104–111, October 2009.
- [48] L. Bao, L. Zhang, S. I. Yu et al., "Informedia@TRECVID2011," in *Proceedings of the NIST TRECVID Workshop*, 2011.
- [49] L. Cao, S. F. Chang, N. Codella et al., "IBM Research and Columbia University TRECVID—2011 multimedia event detection (MED) system," in *Proceedings of the NIST TRECVID Workshop*, 2011.
- [50] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: a comprehensive study," *International Journal of Computer Vision*, vol. 73, no. 2, pp. 213–238, 2007.
- [51] F. Li, G. Lebanon, and C. Sminchisescu, "Chebyshev approximations to the histogram χ^2 kernel," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 2424–2431, June 2012.
- [52] "Trecvid Multimedia Event Detection Evaluation Plan," <http://www.nist.gov/itl/iad/mig/upload/MED11-EvalPlan-V03-20110801a.pdf>.
- [53] "Trecvid Multimedia Event Detection Evaluation Plan," http://www.nist.gov/itl/iad/mig/upload/MED13.Evaluation_Plan_v2.pdf.
- [54] C. Y. Liu, H. J. Wang, J. X. Chen, and X. J. Zou, "Research on service-oriented and HLA-based simulation model of juice production line," in *Proceedings of the International Conference on Measuring Technology and Mechatronics Automation (ICMTMA '10)*, vol. 3, pp. 167–170, Changsha, China, March 2010.
- [55] C. H. Hu, X. H. Chen, M. Wu et al., "A service trust negotiation and access control strategy based on SLA in cloud computing," *Science China Information Sciences*, vol. 42, no. 3, pp. 314–332, 2012.
- [56] C. Y. Liu, S. B. Dong, H. L. Li, B. Lu, and A. Hauptmann, "A TPSAC model and its application to mechanical cloud simulation," *International Journal of Security and Its Applications*, vol. 8, no. 1, pp. 45–56, 2014.
- [57] <http://vision.stanford.edu/projects/objectbank/>.
- [58] D. Hoiem, A. A. Efros, and M. Hebert, "Automatic photo pop-up," *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 577–584, 2005.