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.

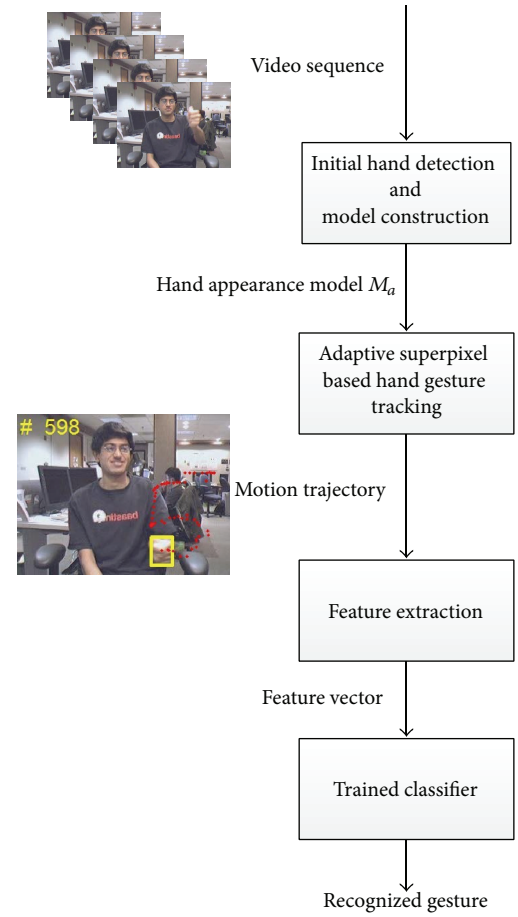

Figure 1: Overall framework of the proposed hand gesture recognition system.

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

$$D_p = \sum_{i=2}^{M} \left| I_i\left(S_p\right) - I_1\left(S_p\right) \right|, \quad p = 1, \ldots, P. \tag{1}$$

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

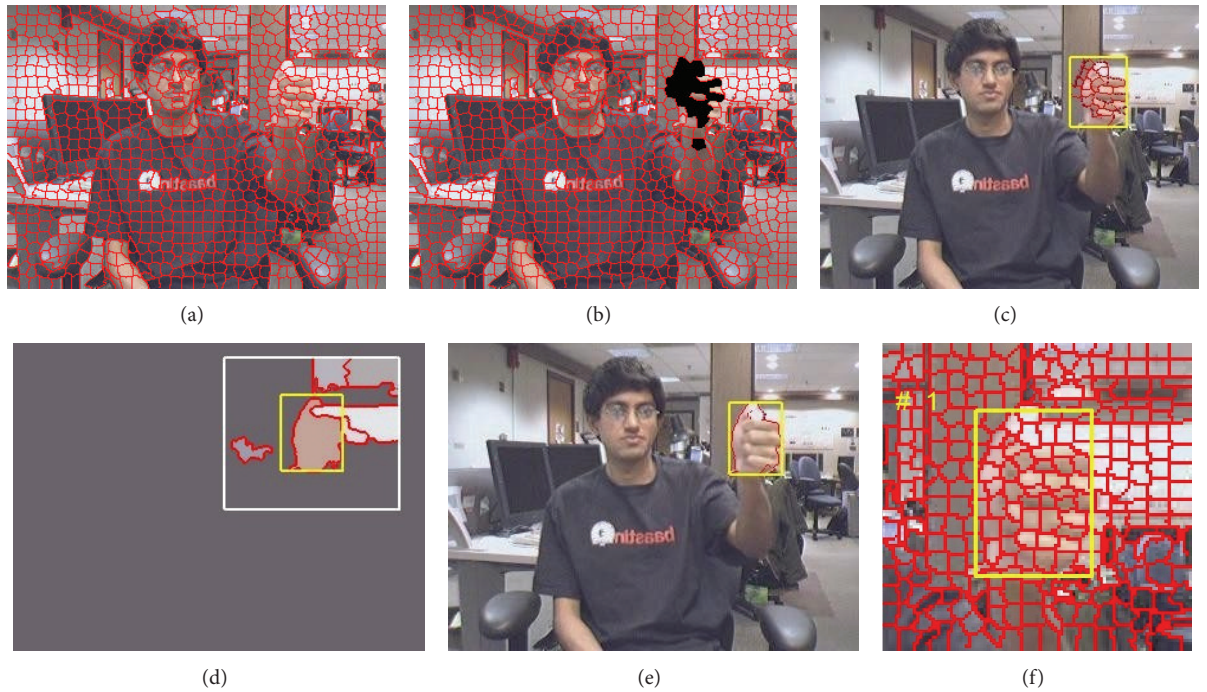

Figure 2: SLIC and CTM hand detection. (a) SLIC superpixels on the first frame. (b) Superpixels with slight motions. (c) Candidate hand region on the connected superpixels. (d) CTM objects on the surrounding of candidate hand region. (e) Refined hand region. (f) Superpixel on the surrounding of the hand 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)):

$$R_H = O_k \mid A_k = \max\left(\frac{\text{size}\,(R \cap O_i)}{\text{size}\,(R \cup O_i)}\right), \quad i = 1, \ldots, K. \tag{3}$$

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

$$S\left(X_1, X_t^n\right) = \frac{S\left(X_1, X_t^n\right)}{\sum_{i=1}^{N} S\left(X_1, X_t^i\right)},$$
$$\text{where} \quad S\left(X_1, X_t^n\right) = \exp\left(\frac{-\sum \left(I_1 - I_t^n\right)^2}{c}\right), \tag{4}$$

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:

$$C_i^c = \frac{\text{Size}^+(i) - \text{Size}^-(i)}{\text{Size}^+(i) + \text{Size}^-(i)}, \quad \forall i = 1, \ldots, n, \tag{5}$$

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

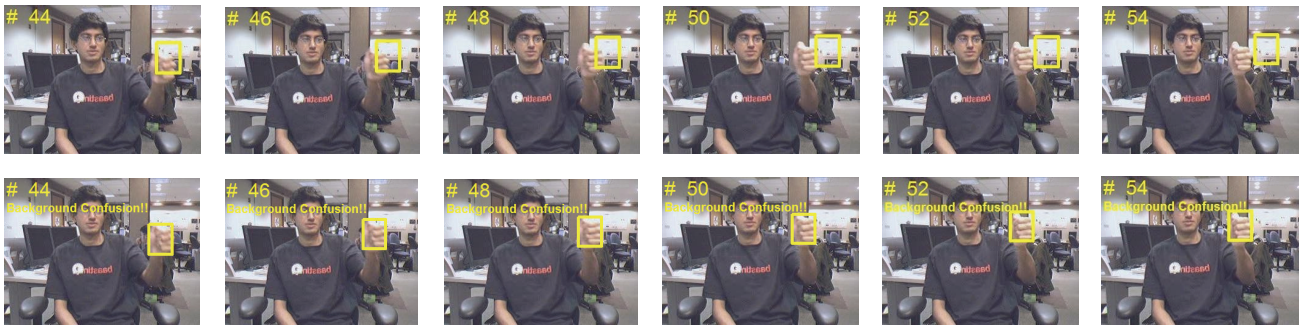

Figure 4: Typical example results. First row: background confusion occurred in SPT from frame 44 to frame 54. Second row: background confusion recovered in our hand tracking solution.

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.

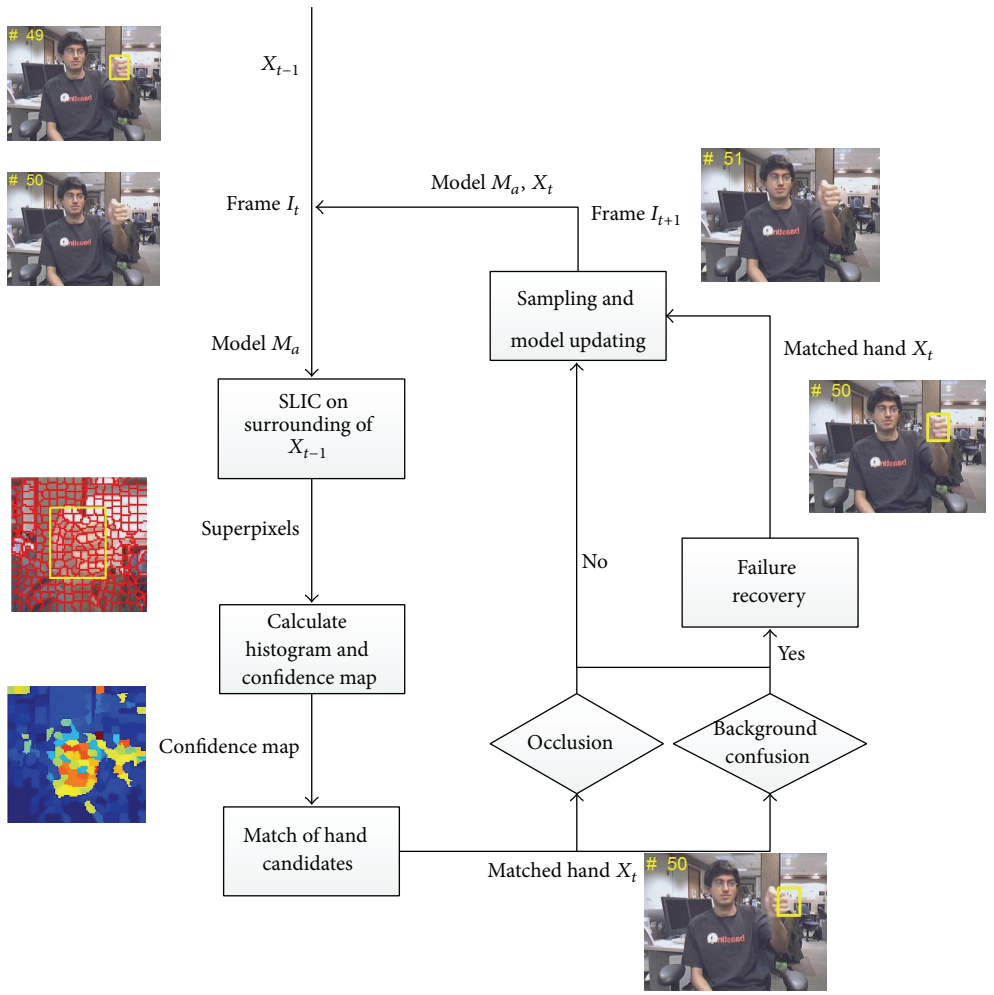

Figure 6: Workflow of the proposed adaptive superpixel based hand tracking.

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:

$$\omega(r,i) = \exp\left(-2 \times \frac{\left\| f_t^r - f_c(i) \right\|^2}{r_c(i)}\right), \quad \forall r = 1,\ldots,P_t; \quad i = 1,\ldots,n,$$
$$C_r^s = \omega(r,i) \times C_i^c, \quad \forall r = 1,\ldots,P_t, \tag{6}$$

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:

$$\frac{\sum S_t}{\text{size}(S_t)} < b, \quad S_t = x \text{ or } (S_t, S_{t-m}) \,\&\, (\sim S_{t-m}), \tag{8}$$

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:

$$X_t = \arg\max_{X_t^n} p\left(X_t^n \mid X_{t-1}\right) C_t^n. \tag{11}$$

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:

$$\operatorname{std}\left(\overset{L}{\underset{i=t-L+1}{X}}\right) < T. \tag{12}$$

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

$$\max\left(\frac{1}{2^h} \times \frac{H - h + 1}{H} \times \frac{1}{C_h}\right), \quad h = 1, \ldots, H \tag{13}$$

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.

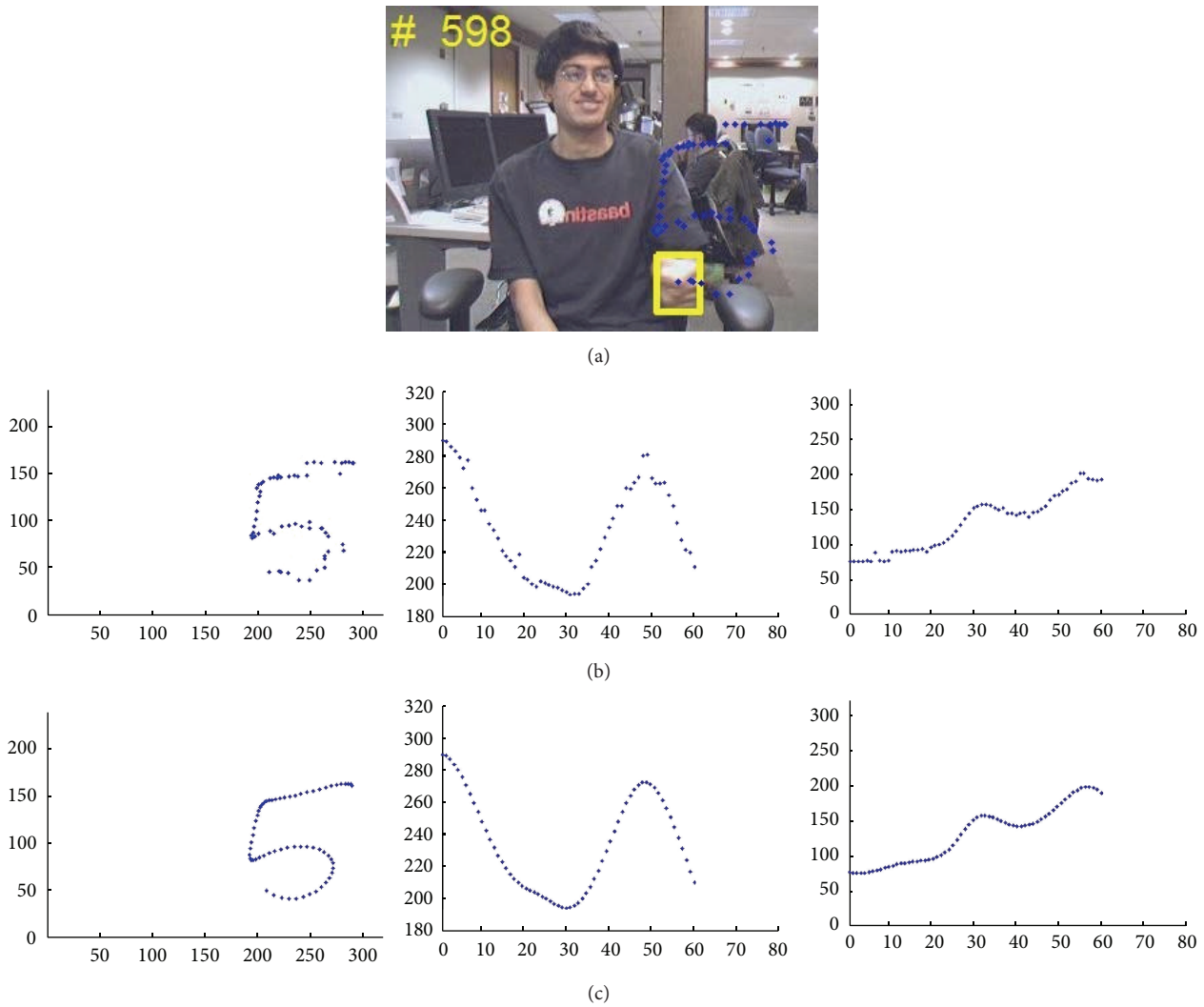

FIGURE 8: Trajectory interpolation. (a) Accumulated trajectory on the last frame of gesture "5"; (b) tracked trajectory; (c) interpolated and smoothed trajectory. Columns from left to right are trajectory plot with gesture's (X, Y); plot X of gesture trajectory; plot Y of gesture trajectory.

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.

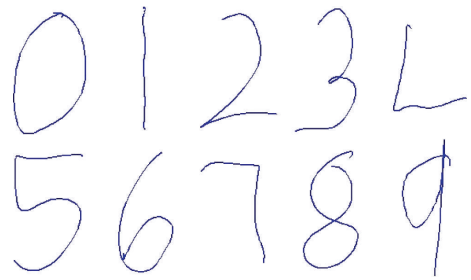

FIGURE 9: Ten classes of hand signed digit gestures.

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.