Journal of Information Hiding and Multimedia Signal Processing ©2015 ISSN 2073-4212
Ubiquitous International Volume **6**, Number **6**, November **2015**

A Real-time Action Detection System for Surveillance Videos Using Template Matching

Mao-Hsiung Hung*

College of Information Science and Engineering
Fujian University of Technology
No.3, Xueyuan Road, University Town, Minhou, Fuzhou, 350118, China
*:Corresponding author, mhhung@fjut.edu.cn

Jeng-Shyang Pan

College of Information Science and Engineering
Fujian University of Technology
No.3, Xueyuan Road, University Town, Minhou, Fuzhou, 350118, China
jengshyangpan@fjut.edu.cn

Received May, 2015; revised July, 2015

Abstract. *In past two decades, computer vision techniques applied to prevent criminals for surveillance videos receive much attention. Because hands-up actions in surveillance videos likely implicate that someone commits robbery, hands-up becomes a good clue to notify robbery happening. This paper presents a real-time action detection system to recognize hands-up actions in surveillance videos. We develop a template matching method to apply to the foreground extracted by background subtraction. According to the matching results, a multi-frame decision delivers detection results of hands-up actions in a video. The experimental results indicate that our proposed system is effective and efficient to detect hands-up actions, and achieves a good performance of 90% recall and 89% precision rates.*
Keywords: Action detection, Hands-up action, Template matching.

1. **Introduction.** In the past two decades, surveillance systems are widespread in many streets and stores due to security problems. However, because surveillance videos are very tedious, monitoring videos in manual ways to promptly prevent criminals becomes infeasible. Therefore, computer vision techniques applied to prevent criminals for surveillance videos receive much attention.

"Hands-up" is often used by somebody who is threatening people with a gun or other weapons to tell them to raise both hands in the air. Because hands-up are often commanded by robbers during store robbery, hands-ups in surveillance videos likely implicate that someone commits robbery. Thus, hands-up becomes a good clue to notify store robbery happening. It will be helpful to notify store robbery to security departments through silent alarms by integrating vision-based hands-up detection with surveillance systems [1].

In past years, many researchers have contributed various video-based action detection methods and we will review some works related to hands-up detection in the following. Shechtman and Irani [2] presented local self-similarity descriptor, as shown in Fig.1. Fig.1(a) shows a hands-up image by hand drawing. First, the center block in the hands-up image and its neighbor area are given, as shown in Fig.1(b). Then, the correlation image between the center block and its neighbor area is calculated. A log-polar map of 20

angle intervals and 4 radius intervals is used to partition the correlation image into several fan regions, as shown in Fig.1(c). Finally, a feature descriptor is extracted according the log-polar map, as shown in Fig.1(d). Based on the descriptor, the action detection is performed by searching blocks to match pre-stored descriptors of action templates. Working without segmentation is the method's advantages. However, the effectivity of the descriptor mostly depends on the homogeneous content of target objects such as performers in action. Many target objects are complicated contents in practical, so that the descriptor extraction likely fails due to poor homogeneity of object contents.

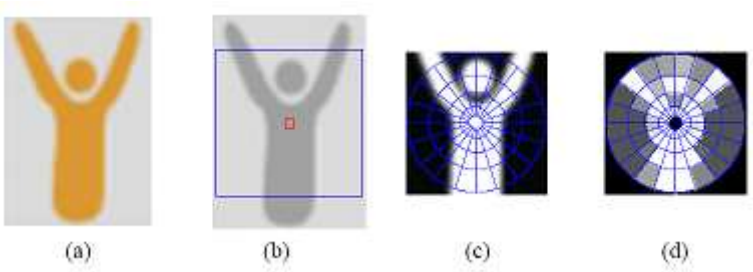

FIGURE 1. Extraction of local self-similarity descriptor: (a) hands-up image by hand drawing, (b) center block and neighbor area, (c) correlation image and log-polar map, (d) descriptor extraction

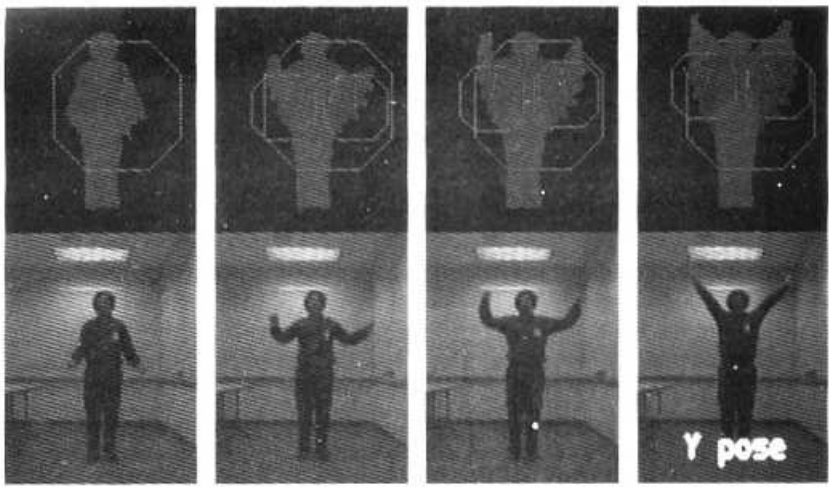

FIGURE 2. Y pose detection of tHMI [5]

G. Bradski and J. Davis [3]-[5] proposed a motion representation named by tHMI (timed Motion History Image). The tHMI representation consists of motion timestamps of a silhouette, that updates along time. The authors set some ROIs (region of interest) in tHMI and used to represent the ROI's motion. As shown in Fig.2, three ROIs of octagon are assigned in a body and two arms. When the performer raise his arms, the body's ROI does not move but the two arm's ROIs move upward. The three motions can infer a Y pose performing. The motion of the three ROIs also infer other actions such as T pose and |- pose. However, identifying and tracking the silhouette are still difficult to achieve in the complicated backgrounds.

D. H. Jang et al. [6] applied a neural network to classify four poses including hands-up, hands-down, left arm raising and right arm raising. The method uses multi-scale

background subtraction to obtain foreground and quantizes it into a 16x16 image in the spatial space. The quantized image is used for training by the neural network. However, in this method, the silhouette of the body and arms is required to robustly segment under various background changes.

Recently, a local feature of Histogram of Oriented Gradients (HOG) received much attentions in many researches related to action recognition [18]-[21]. The spatial or spatial-temporal gradients in the consecutive frames are computed, and then the histogram of gradients in various orientations is used to describe motion characteristic of actions. Although, HOG methods do not need silhouette segmentation and have good generalization for different actions, the parameter adjustment is still complicated in the training procedure. Therefore, HOG methods cannot be easy to used by general users without technical knowledge.

Most of other paper [7]-[13] related to action recognition are of multi-class recognition algorithms. These algorithms are very complicated and not suitable for single-class recognition such as hands-up detection. A recognition system for single class should be more simple and more practical. According to the principles, our proposed method has the following characteristics: 1) applying a simple algorithm of background subtraction to extract foregrounds, 2) recognizing actions under incomplete foreground segmentation results, 3) using templates as prior knowledge for recognition, and 4) easy to build templates for general users.

In this paper, we propose a hands-up detection system for surveillance videos. First, a background subtraction is applied to obtain foreground objects. The hands-up templates of system user can be captured and stored in prior. Next, a new matching scheme is proposed to perform template matching in a video frame. Finally, the processing verifies the matching results of multiple frames to check whether hands-up actions happens. The remainder of this paper is organized as follows. Section 2 describes the proposed system. Section 3 demonstrates the performance evaluation of the proposed methods and discusses the experimental results. The conclusions are drawn in Section 4.

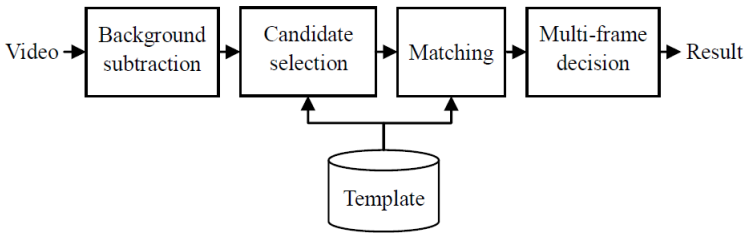

FIGURE 3. Flowchart of proposed system

2. **Proposed System.** The flowchart of our proposed system is shown in Fig.3. First, a background subtraction is used to obtain a foreground. Secondly, several candidate blocks are selected to match template. Thirdly, template matching is applied on each candidate block. Finally, a decision of whether hands-up action happens, is made from the matching results of the multiple frames. The remaining sections will introduce these processing of our proposed system.

2.1. **Background subtraction.** Background subtraction is one of the most popular method to segment foreground for moving objects in surveillance videos captured by fixed cameras. We observe that a hands-up action spends about 0.5-1 second. Thus, a one-second long video before a hands-up action can be used to build a good reference for the background model. To build the background model, the pixel-wise median operation through previous frames is implemented as

$$I_{Med}(x, y, k) = \text{Median}(I(x, y, k-1), I(x, y, k-2), ..., I(x, y, k-\Delta T)) \quad (1)$$

where $I(x, y, k)$ means a pixel in (x, y) at k-th frame and ΔT means a time interval, e.g. $\Delta T = 31$ frames (about one second at fps=30Hz). Furthermore, to speed up the background subtraction, we developed a fast algorithm to reduce the computation time of the temporal median operation [14]

. After background modelling, the binarization is done by Eq.(2). If the absolute difference between the current frame of $I(x, y, k)$ and the background model of $I_{Med}(x, y, k)$ is greater than a preset threshold, the binarized image of $B(x, y, k)$ will be assigned to bit-1, otherwise to bit-0.

$$B(x, y, k) = \begin{cases} 1, \text{ if } |(I(x, y, k) - I_{Med}(x, y, k)| > Th \\ 0, \text{ otherwise} \end{cases} \quad (2)$$

After the binarization, we use majority operation to combine the results of the RGB components as

$$B'(x, y, k) = \text{Majority}(B_R(x, y, k), B_G(x, y, k), B_B(x, y, k)) \quad (3)$$

where $B_R(.)$,$B_G(.)$ and $B_B(.)$ means the three binary images of RGB components resulted by Eq.(2) After background subtraction, there are some noises in the foreground object. To eliminate these noises, we apply a morphological erosion operation on the binary image obtained by the background subtraction. The illustration of foreground segmentation is shown in Fig.4.

FIGURE 4. Illustration of foreground segmentation: (a) background model before hands up, (b) current frame while hands up, (c) foreground of current frame

In several researches of shape recognition, the complete contour (or shuttle) of the body is often expected to obtain after background subtractions. Although there are existing complicated techniques which can group multiple regions into a single object to build a body part [15], complete shuttles are very difficult to be obtained in practical. In our work, the segmentation results of foreground object are quiet incomplete and fractionized in a body part, as shown in Fig.4(c). The incomplete segmentation of the body part is very hard to be applied on feature descriptor approaches such as CSS and ART proposed by MPEG-7, because these approaches require complete shapes to extract feature effectively.

Therefore, we consider template matching approaches to deal with incomplete foreground segmentation for our system.

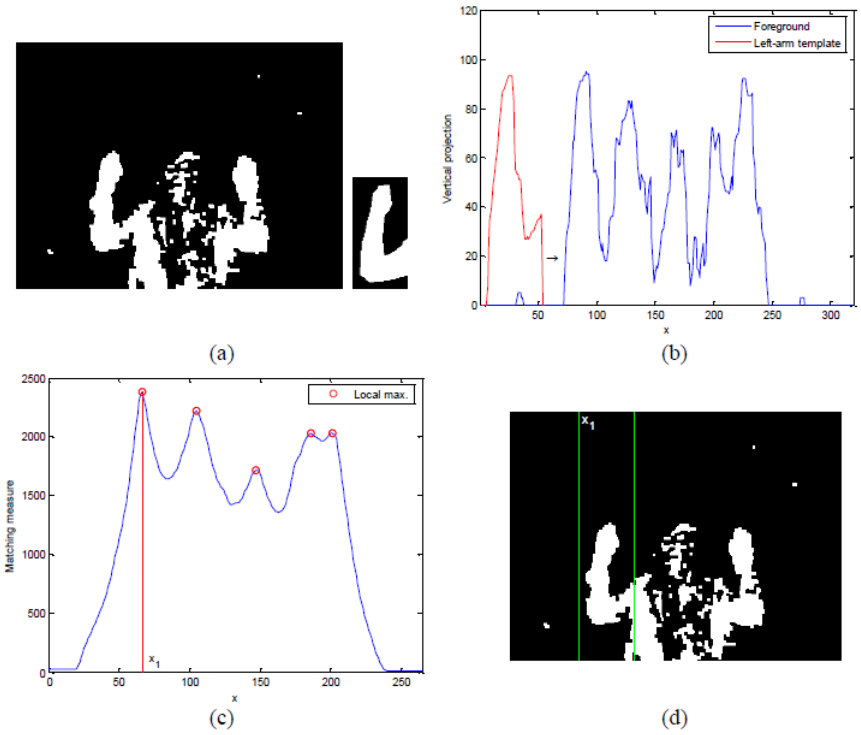

FIGURE 5. An example of candidate stripe in x-direction: (a) foreground and left-arm template, (b) vertical projection of (a), (c) matching result of (b), (d) the stripe at the first local maximum of (c)Four examples of foreground segmentation

2.2. **Candidate selection for template matching.** In common, the template matching consumes a huge amount of computation, if the matching runs through the full image of each frame. To reduce the computational cost, we develop a scheme of candidate selection to search blocks which are potentially matched by templates. Moreover, the candidate selection also makes template matching more effective.

A projection of a binary image is defined by the summation of bit-1 pixels along x-axis or y-axis. The vertical and horizontal projections are computed respectively along y-axis and x-axis. We consider the L-shape arm template used in our proposed system, and learn that the projection feature of the template is very significant. Thus, we choose the projection as a measure to select candidates. Eq.(4) is used to compute the matching measure by the vertical projection.

$$g(x) = \sum_{s=0}^{L_w-1} min(f(x+s), w(s)), \text{ for } x = 0, 1, ..., L_f - L_w \tag{4}$$

where $f(.)$ is the vertical projection of the foreground and its length is L_f, and $w(s)$ is the vertical projection of a template and its length is L_w. $f(s)$ for $s = 0, 1, ..., L_w - 1$ represents a L_w-long part of the foreground projection, so $f(x+s)$ for $x = 0, 1, ..., L_f - L_w$, makes $f(s)$ walking through the foreground projection along x-axis. The similarity measure of two projections of $f(x+s)$ and $w(s)$ is computed by the summation of a minimal

operation. Fig.5 illustrates an example of the candidate selection of the vertical projection along x-axis. Fig.5(a) shows a foreground and a left arm template, and Fig.5(b) shows the two vertical projections of Fig.5(a). Fig.5(c) shows the measure curve of matching the two projections along x-axis. The location of local maximums on the curve means where the foreground and the template match well by the project measure. Based on the location of local maximums, we can determine the candidate stripes for the matching template. Fig.5(d) shows the stripe at the first local maximum of Fig.5(c).

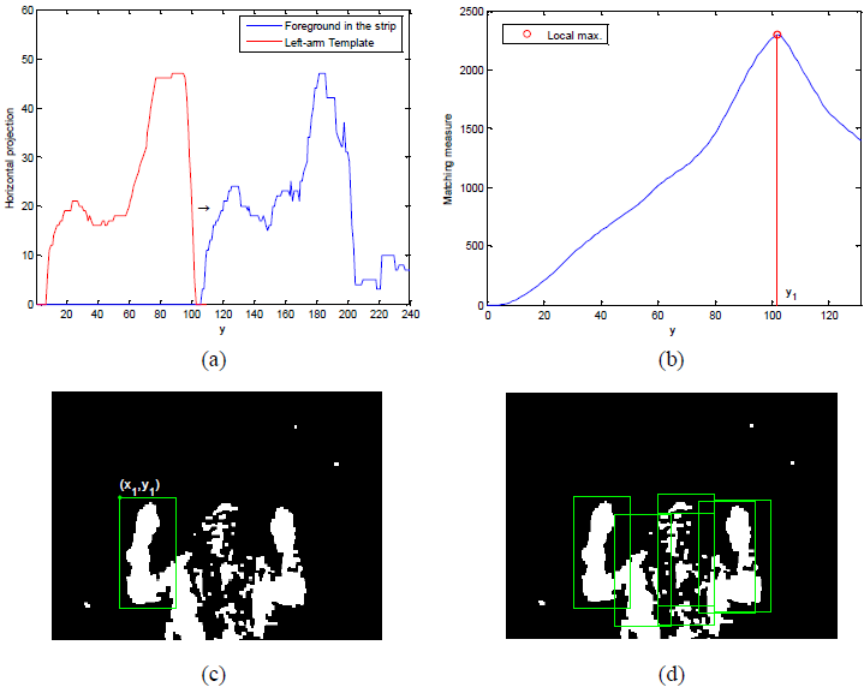

FIGURE 6. An example of candidate block: (a) horizontal projection of foreground and left-arm template, (b) matching result of (a), (c) the block at local maximum of (b), (d) all of candidate blocks between foreground and left-arm template

After then, considering the horizontal projection of along y-axis, we find candidate blocks within a single candidate stripe. The similar computation of Eq.(4) obtains the measure of matching two horizontal projections of the candidate stripe and the arm template. Then, the blocks locating the local maximum are considered as the candidate. Fig.6 demonstrates a example of the candidate block selection. Fig.6(a) shows the horizontal projections of the candidate stripe and the template. Fig.6(b) shows the matching measure of the two projection and its local maxima. Fig.6(c) shows the candidate block corresponding to the maxima of Fig.6(b). Fig.6(d) shows all candidate blocks from all candidate stripes. Using the candidate selection, the matching between the template and the foreground becomes very few in the next processing. As a result, the computational cost of template matching can be greatly decreased.

2.3. **Template generation and matching.** Generally, the number of system users such as staff in a store is limited. We assume that users' arm shapes of hands-up actions have small changes in surveillance videos. Thus, user's arm shapes can be used to build templates and then stored them in the database in advance. During the testing, these

templates are retrieved to match the input foreground.

In template generation, a semi-automatic or manual method is used to track arm contours. The arm template includes upper arm, lower arm and hand. When a person puts his hands-up, the left arm forms a L shape and the right arm forms a mirror-L shape, as shown in Fig.7(a). The techniques of the contour-based segmentation such as active contour models [16] can obtain the arm contour automatically, so that users just need very few efforts to build the template.

The template building is shown as Fig.7(b). A shape often composes foreground pixels and background pixels. In our work, we define the template shape as foreground, background and transparent areas. Pixels inside the arm contour are foreground, the dilation pixels outside the arm contour are background and other pixels are transparent. The transparent pixels do not get involved with computation of similarity measuring of template matching. This way can avoid that the pixels far outside the arm affect the similarity measuring. Fig.8 exhibits a template database of nine persons, which contains nine pairs of arm templates segmented out of these persons.

FIGURE 7. Template building: (a) manual segmentation of two arms when hands up, (b) left- and right-arm templates from the segmentation

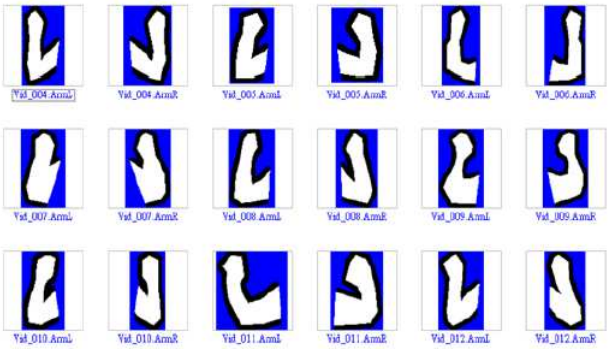

FIGURE 8. Template database of nine persons

Based on the template building, we propose a new shape similarity measure for template matching. The foreground and background of a testing shape (X) are respectively denoted by X_F and X_B. The foreground and background of a template shape (T) are respectively denoted by T_F and T_B. All the four images of X_F, X_B, T_F, T_B are binary.

The similarity measure of X and T is computed by Eq.(5).

$$\begin{cases} s = w_F \cdot area(X_F \cap T_F) + w_B \cdot area(X_B \cap T_B) \\ p = w_F \cdot area(X_F \cap T_B) + w_B \cdot area(X_B \cap T_F) \\ SM(X,T) = s - p \\ NSM(X,T) = SM(X,T)/SM(T,T) \end{cases} \tag{5}$$

We first calculate the two terms of s and p. s means a score between X and T. p means a penalty between X and T. The score (s) is defined by the weighting sum of two intersection areas. The two intersections of foreground and background are $X_F \cap T_F$ and $X_B \cap T_B$. $Area(.)$ returns the sum of bit-1 pixels in a binary image. The penalty (p) is also defined by the weighting sum of two intersection areas, but the two intersection of foreground and background formed in the opposite way, are $X_F \cap T_B$ and $X_B \cap T_F$. Then, the similarity measure of X and T is defined by $s-p$, which denoted by $SM(X,T)$. Finally, we normalize the $SM(X,T)$ into [0,1] by dividing $SM(T,T)$. In Eq.(6), the two weightings of w_F and w_B for foreground and background are defined by the two area ratios for a template shape.

$$\begin{cases} w_F = area(T_F)/(area(T_F) + area(T_B)) \\ w_B = area(T_B)/(area(T_F) + area(T_B)) \end{cases} \tag{6}$$

We match the pre-stored templates within the selected candidate blocks. Finally, we obtain the best matching of a pair arms from the template database. Fig.9 shows the matching results for different five persons. The results indicate that even if the extracted foregrounds are incomplete and fractured, our proposed template matching is still successful.

FIGURE 9. Matching results for different five persons

2.4. **Multi-frame decision.** After the template matching, we obtain a measure of the best matching from the template database for every frame. Thus, a signal of the matching measure in every frame can be extracted from a video sequence, as shown in Fig.10. When the signal increases to a high value at a frame, it supposes that a person in the video behaves a hands-up action. Thus, we use the multi-frame signal to decide whether hands-up action happens or not. The original signal is too noisy to be applied, so we smooth it by Gaussian filtering to obtain a smoothed version. Finally, we detect out hands-up actions in high local maximums of the smoothed signal of the matching measure. A threshold (Th) can be a preset parameter and a hands-up is detected out once a local maximum is more than Th.

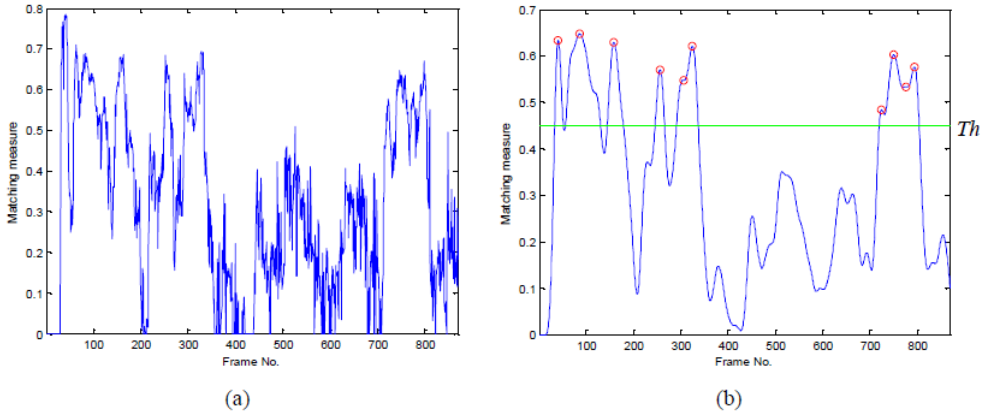

FIGURE 10. Decision result in multi-frames: (a) original, (b) smooth version

3. **Experimental Results.** We capture nine testing videos which contains several hands-up actions performed by nine persons. The frame number and length of the nine videos are listed in Table 1. The video size of the videos is 160x120 and the frame rate is 30 fps. The nine persons include four males and five females with different body figures and clothes. The gestures of hands-up are not fixed, and the position of the two hands can be in high or low.

TABLE 1. Experimental data of testing videos

Video clip	No.1	No.2	No.3	No.4	No.5	No.6	No.7	No.8	No.9	Total
Frame #	1590	810	600	450	870	1470	2100	720	1350	9960
Length (sec)	53	27	20	15	29	49	70	24	45	332
Processing time (sec)	44	22	9	8	14	40	58	19	37	251
# of hands-up	9	7	6	9	5	22	17	7	6	88

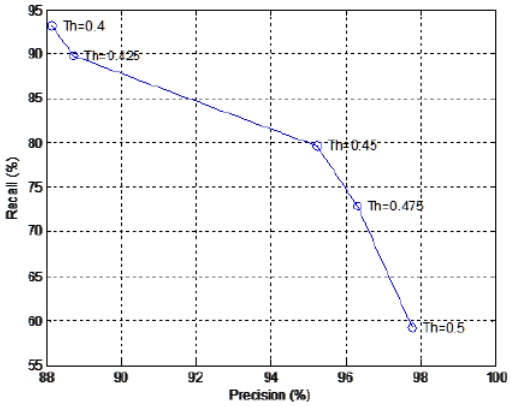

FIGURE 11. Recall vs. precision curve with varied Th

We use C++ and OpenCV library to develop the system. Our testing bench uses Visual C++ with Intel Core i-5 CPU at 2.5GHz. The processing time of our proposed method for the testing videos is listed in Table 1. The total processing time is 251 sec, so that

the processing capacity of our propose method is 39.68 frame/sec and meets real-time requirement, i.e. more than 30 frame per second.

To evaluate the performance of our proposed method, we manually label the time duration (consecutive frames) whether hands-up actions happen or not, as ground truth in the testing videos. Thus every duration of the testing videos can be labeled as the Positive or Negative hands-ups. In our experiments, hands-up actions happen 88 times in total in the testing videos, as listed in Table 1. Thus, the three of permutations of the ground truth vs. detection result are: labeled Positive and detected Positive (true positive, TP), labeled Positive but detected Negative (false negative, FN), labeled Negative but detected Positive (false positive, FP). Finally, we compute the recall rate and the precision rate by Eq.(7).

$$Recall = \frac{TP}{TP + FN} \times 100\%, Precision = \frac{TP}{TP + FP} \times 100\% \tag{7}$$

We plot a recall vs. precision curve with a varied threshold (Th) in multi-frame decision, as shown in Fig.11. Th varies from 0.4 to 0.5 by 0.025 increment. According to the curve, we found that the precision rate increases from 88% to 98%, but the recall rate drops sharply from 93% to 60%. The operation point of $Th = 0.425$ achieves a good performance of Recall=90% and Precision=89%. However, when $Th \geq 0.425$, the recall rate decreases below 80%. It means that the Th parameter is quite sensitive.

Fig.12 exhibits hands-up detection results for testing videos. The results indicate that our proposed method successfully detects hands-up with various gestures including different hands' heights and body's directions. Furthermore, there are several correct detection under different wearing. It indicates that our proposed method is robust under various conditions.

4. **Conclusions.** This paper have presented a system to detect hands-up action for surveillance videos using template matching. The arm templates of users are built and stored in advance. After background subtraction, our proposed candidate selection and template matching are very effective and efficient to search arm templates in foreground. Finally, multi-frame decision is used to point out the action duration in videos. The experimental results indicate a good performance of 90% recall and 89% precision rates. More importantly, our proposed system can meet real-time requirement.

In surveillance videos, there are other actions to represent some abnormal events, such as waving for a help and punching for a fight. The detection of these actions also benefit for criminal preventing and security. Our proposed system can be extended for other action detection in the further works.

Acknowledgment. This work was supported in part by National Science Counsel Granted NSC 99-2221-E-151-052.

REFERENCES

[1] J. S. Pan, Z. Meng, K.-K. Tseng and C. L. Lu, A robbery monitoring and alarm method and system, Feb 11-2015, *CN Patent* 102,867,383.

[2] E. Shechtman and M. Irani, Matching Local Self-Similarities across Images and Videos, I*EEE Conference on Computer Vision and Pattern Recognition 2007* (CVPR'07), June 2007.

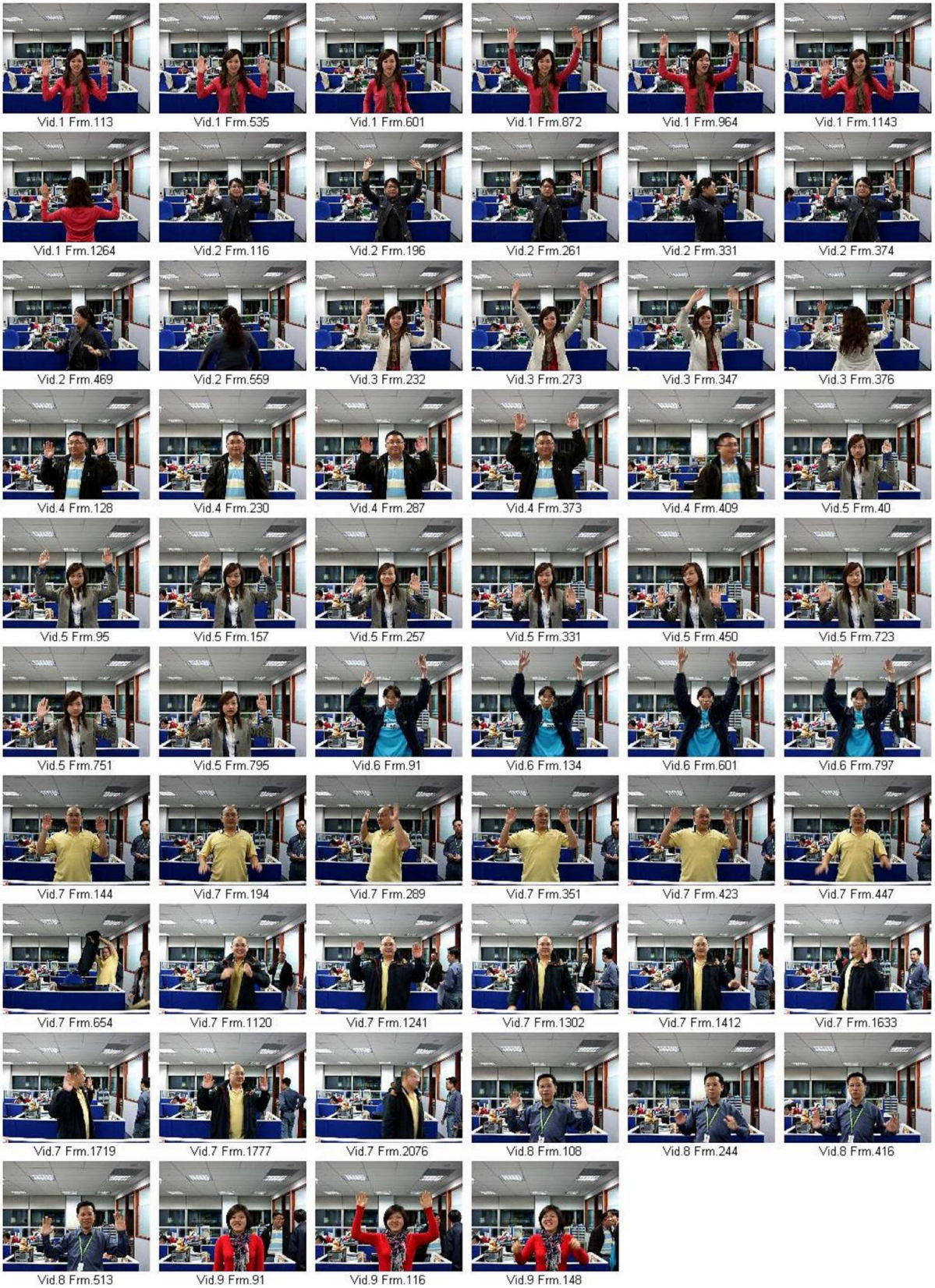

FIGURE 12. Detection results

[3] G. Bradski and J. Davis, Motion segmentation and pose recognition with motion history gradients, *IEEE Workshop on Applications of Computer Vision*, 2000.
[4] J. Davis and G. Bradski, Real-time motion template gradients using Intel CVLib, *ICCV Workshop on Framerate Vision*, 1999.
[5] G. Bradski and A. Kaebler, Learning OpenCV, *OReilly Media Inc*, 2008
[6] D. H. Jang, Y. J. Chai, X. H. Jin and T. Y. Kim, Realtime Coarse Pose Recognition using a Multi-Scaled Local Integral Histograms, *Proc. of International Conference on Convergence Information Technology*, pp. 1982-1987, 2007.
[7] T. Urano, T. Matsui, T. Nakata and H. Mizoguchi, Human Pose Recognition by Memory-based Hierarchical Feature Matching, *Proc. of IEEE International Conference on Systems, Man and Cybemetics*, pp.6412-6416, 2004.
[8] N. Ikizler and P. Duygulu Histogram of oriented rectangles: A new pose descriptor for human action recognition, *Image and Vision Computing*, vol.27, no.10, pp.1515-1526, Sep. 2009.
[9] X. Wu, W. Liang and Y. Jia, Action recognition feedback-based framework for human pose reconstruction from monocular images, *Pattern Recognition Letters*, vol.30, no.12, pp.1077-1085, Sept. 2009.
[10] F. Huo, E. Hendriks, P. Paclik and A.H.J Oomes, Markerless human motion capture and pose recognition, *Proc. of 10th Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS09)*, pp.13-16, May 2009.
[11] T. Kawano, K. Yamamoto, K. Kato and H. Hongo, Integration of pose recognition for a person wearing short or long sleeves, *Proc. of 16th International Conference on Pattern Recognition (ICPR02)*, vol.3, pp.1023-1026, Aug. 2003.
[12] C. Chen and G. Fan, Hybrid Body Representation for Integrated Pose Recognition, Localization and Segmentation,*Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp.1-8, June 2008.
[13] H. Yuan, A Semi-supervised Human Action Recognition Algorithm Based on Skeleton Feature, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 6, no. 1, pp. 175-182, January 2015.
[14] M.-H. Hung, J.-S. Pan and C.-H. Hsieh, A Fast Algorithm of Temporal Median Filter for Background Subtraction, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 5, no. 1, pp. 33-40, Jan. 2014.
[15] C. M. Kuo, C. H. Hsieh, Y. R. Huang, Automatic extraction of moving objects for head-shoulder video sequence, *J. Visual Communication and Image Representation,* vol. 16, no. 1, pp.68-92 (2005).
[16] M. Kass, A. Witkin, D. Terzopoulos, Snakes: Active contour models, *International Journal of Computer Vision*, vol.1, no.4, pp.321-331, 1988.
[17] L. Shao, L. Ji, Y. Liu, J. Zhang, Human action segmentation and recognition via motion and shape analysis, Pattern Recognition Letters, vol. 33, no. 4, pp. 438-445, 2012.
[18] P. Dollar, V. Rabaud, G. Cottrell, S. Belongie, Behavior Recognition Via Sparse Spatio-Temporal Features, Proc. of 2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, pp.65-72, Oct. 2005.
[19] N. Dalal and B. Triggs, Histograms of Oriented Gradients for Human Detection, *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp.886-893,June 2005.
[20] I. Laptev, On Space-Time Interest Points, *International Journal of Computer Vision,* vol. 64, no. 2-3, pp. 432-439, 2005.
[21] I. Laptev, M. Marszalek, C. Schmid, B. Rozenfeld, Learning realistic human actions from movies, *Proc. of IEEE Conference on on Computer Vision and Pattern Recognition*, pp.1-8, 2008.