Improvements of RGB-D Hand Posture Recognition Using an User-Guide Scheme

Huong-Giang Doan*†, Hai Vu*, Thanh-Hai Tran*, Eric Castelli*
*International Research Institute MICA HUST - CNRS/UMI - 2954 - INP Grenoble
†Industrial Vocational College Hanoi
Email: {huong-giang.doan,hai.vu,thanh-hai.tran,eric.castelli}@mica.edu.vn

***Abstract*—This paper argues that an user-guide plays an important role to make a robust and real-time hand posture recognition system. Instead of designing a new hand posture recognition algorithm, we propose an user-guide scheme which handles issues of environmental conditions as well as appearance-based features for hand detections. This guide estimates heuristic parameters whose values strongly affect the recognition results. The experimental results confirm that even by utilizing a simple hand posture recognition algorithm, the proposed method significantly improves the recognition rate. Without training end-user, the recognition rate achieves only 63%, whereas it obtains 87% with the proposed guide scheme. To guide an end-user, the proposed scheme requires averagely 15 seconds in advance. Therefore, the proposed method is feasible to deploy practical applications, particularly, to control devices in a smart-home such as televisions, game consoles, or lighting systems.**

Keywords*—Human Computer Interaction; User-guide scheme; Hand posture recognition.

I. Introduction

Communications between human and computer (or recently, between human and smart device) become more and more natural and intuitive. Human-Computer Interaction (HCI) therefore has been a wide field of research in the last 30 years, which has led to uncountable algorithms and methods [1],[2]. However, utilizing hand posture as a natural interaction is still very challenging problem due to the complexity of hand shapes, and high computational costs of the vision algorithms. On one hand, robustness plays an important role because the recognition algorithms suffer from various hand postures under different lighting conditions and cluttered backgrounds. On the other hand, in order to ensure the natural and intuitive feedbacks, the vision-based hand posture recognition algorithms should process video stream in real time. This paper tackles a trade-off (or a balance) between accuracy-real time performance and a cost of the user-dependence. This cost pays for guiding end-users to improve the robustness and real-time performances. For this reason, in this work, we propose an efficient user-guide scheme to require minimal learning time and user intentions.

While most of the hand gesture techniques toward an user-independence system which can adapt different users rather than a specific user. In this paper, we argue that a hand posture recognition system is able to be an user-dependence with a reasonable user-guide scheme. This scheme plays an important rule in order to achieve the robust and real time criteria. We do not pay special attentions for designing hand detection or recognition algorithms. Instead of that, we take into account designing an efficient user-guide scheme so that the heuristic parameters in the hand detection and segmentation procedures are adaptively selected. The proposed system will learn contexts of the current environments through the user-guide procedure. They are to answer questions: (1) how is current background scene; (2) How are appearances (e.g., color) of the hand's skin; (3) How far from user's hand to the sensor. For instance, a depth image from a Kinect sensor gives not only distance from hand to the Kinect sensor, but the distance to other objects in a certain scene. By using the result of (3), the hand-distance parameters can be learnt quickly. The candidates of the hand region therefore can be separated in the depth image using such parameters. Similarly, we prune the hand candidates through skin colors learnt in (2). Consequently, we obtain a high accuracy of the segmented hand. We then simply deploy a contour-based hand posture recognition algorithm. Obviously, the proposed system is simply and easily deploying thus archiving a real time performance. The proposed user-guide scheme is activated whenever an end-user starts to use the system. It consumes averagely 15 seconds in advanced for each user. This is an acceptable learning cost for end-user. Consequently, the proposed system is feasible to deploy practical applications. The rest of paper is organized as follows: Section II briefly survey related works. Section III describes the proposed user-guide scheme. The experimental results are shown in Section IV. Finally, Section V concludes and suggests further research directions.

II. Related works

In this study, we pursue a hand posture recognition system for controlling devices (e.g., televisions, lighting systems) in a smart-room. Therefore, we briefly survey recent trends that feasibly deploy to home appliances. Microsoft Xbox-Kinect is a success commercial product recognizing hand/body gestures to control game consoles [3]. Many technology companies launch smart-devices using like-Kinect sensors (e.g., Asus Kinect, softKinect). For instance, Samsung smart-TV can manipulate TV-functions using dynamic hand gestures. Omron introduces the smart-TV integrated facial and hand recognition. PointGrab [4] proposed an unique solution based on shape and motion recognition including gesture, face, and user behavior analytic. Increasingly, in-air gesture recognition is being incorporated into consumer electronics and mobile devices like WiSee system [5]. WiSee focuses on gesture recognition and shows how to extract gesture information from wireless transmissions. It promises new trend for home appliances because this technique can operate in non-line-of-sight scenario, that is a limitation of the vision-based system.

In term of the deploying applications using hand gestures,

978-1-4673-7338-8/15/$31.00

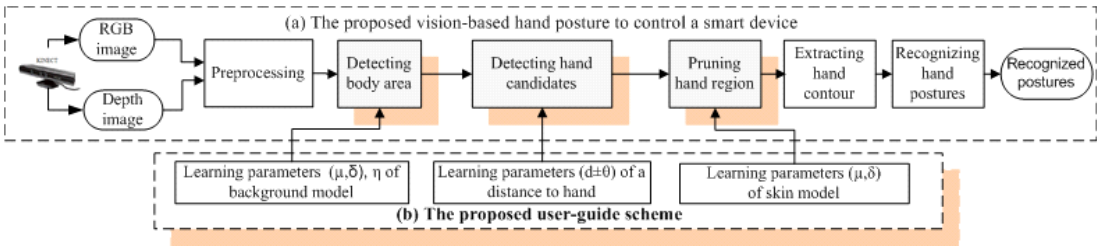


Fig. 1. The proposed framework for hand postures recognition using depth and color (RGB-D) data.

[6] proposes a static hand language recognition system to support the hearing impaired people; [7] uses hand postures to control a remote robot in mechanical systems; Similar systems have been deployed for game simulations such as [8],[9]. The fact that there are uncountable solutions for a vision-based hand posture recognition system. Readers can refer good surveys such as [1],[2] for technical details. Roughly speaking, the vision-based hand posture recognition techniques consist of two phases: hand detection and recognition [1],[2]. In order to achieve high performance of the hand detection, the relevant works often customize or combine multiple features such as color, edge, shape, motion of hand (e.g., [10],[11]). However, such combination schemes consume a high computational time, e.g., [10] requires 2 minutes to detect hands in a photo. To achieve an acceptable recognition rate, a posture classifier or a statistical model always requires sophisticated algorithms. For instance, to build Bayesian network model, [12] constructs multiple layers of the texture, color, shape of the hand; or in order to extract precisely hand contours, [1] needs four layers processing. Obviously, addressing such issues always needs to compute intensively and thus makes current hand posture recognition systems fragile and unstable. Recent works such as [13],[14] utilize depth information to extract correctly hand regions. [15] extracts skeleton of hands and applies Hidden Markov Models for classifying various hand gestures. [16] extracts blob of hand for tracking using both Time-of-Flight and RGB camera. [17] introduces a method for the 3D hand pose and hand configuration. In such works, a large database of hand is prepared in advanced. Briefly, these works try to efficiently utilize depth and color features according to a complicated combination scheme. A common way is that it detects hand using depth feature, and shortcoming of depth results is compensated by RGB feature.

Different from above works, in this study, we focus on an user-guide scheme so that the proposed system can learn current contexts or environmental conditions such as how far from an end-user to device; or how is the current background. Regarding to such issues, some researchers have deployed learning methods such as Boosting method [18], or active boosting in [19], on-line learning [20], online bagging and boosting [21], so on. However, such learning approaches are to design intelligent machines so that the target systems can adapt various goals such as detecting car ([19]), recognizing human, or various type of objects ([18],[20]). In contrast, our proposed user-guide scheme aims to guide an end-user in a passive way. Such learning scheme needs to be simple enough for end-user, low costs, and minimal user's intensions. However, it must be an adaptive system for different end-users as well as various contexts of the environment.

III. PROPOSED APPROACH

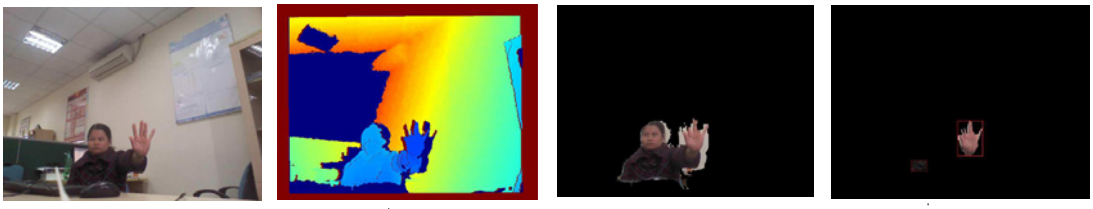
A. The proposed framework

An overview of the proposed framework is shown in Fig. 1 By using a fixed the Kinect sensor [3], a RGB image I and a depth data D are concurrently collected. The main flow-work for detecting hand from (I, D) images consists of a series of the cascaded steps, as shown in Fig. 1(a). Main steps are briefly explained below:

- Preprocessing step: Because I and D images are taken by the sensors that are not measured by the same coordinates. A pre-processing is required in order to calibrate them. In this work, we utilize calibration method proposed by [22].
- Body detection step: Given a background model $BG(\mu, \sigma, \eta)$ (e.g., BG is a Gaussian model with a pair of the parameters $BG(\mu, \sigma)$ and a noise model $BG(\eta)$, body regions are extracted from images I, D by evaluating a probability P:
$$B = P(D|BG(\sigma, \mu, \eta)) \quad (1)$$
- Hand detection step: Given body regions B and a distribution of the depth data D (e.g., $hist(D)$), the candidates of hand H are extracted by:
$$H = hist(B \cap D) < thresh_{hand} \quad (2)$$
- Pruning hand region: Given hand regions H and a statistical hand model such as appearance-based features of hand (e.g., a model Ω of color distribution of hand skin, named Ω_c), hand regions H^* are precisely determined by evaluating a probability P_c:
$$H^* = P_c(H|\Omega_c) \quad (3)$$

Intuitively, the flow-work is a common solution (e.g., similar to a previous work [16]). The major advantages are that it requires lowest computational time due to utilizing simple algorithms. However, this flow-work also requires many heuristic parameters such as a pair of the parameters (σ, μ) in (1), $thresh_{hand}$ in (2), Ω_c in (3)). These parameters are usually pre-determined in the common approaches [16]. Contrary that work, in this paper, we focus on a learning scheme to adaptively select these heuristic parameters to each user. Fig. 1(b) shows associated procedures to select corresponding parameters in equations (1), (2), (3). Moreover, the proposed method requires assumptions:

- The Kinect sensor is mounted on a fixed rack or a tripod. The end-user stands in a valid range of depth feature (e.g., 0.5÷4m for the Kinect sensor [3]).



(a) RGB image (b) Depth image (c) Body extraction (d) Candidates of hand

Fig. 2. Results of hand region detection

- The end-user controls devices by raising his/her hand in front of the body and he/she stands at a fixed position during controlling device.

The first assumption is reasonable in real practical application because position of a device such as a television or a game console usually is fixed. The second assumption is acceptable because of habits of end-users when they control a device. Details of the proposed framework are described in subsections below.

B. Estimating parameters of background model for body detections

Because the sensor and the environment are fixed, we firstly detect human regions using background subtraction techniques. Both depth and RGB data can be used for the background subtraction. However, the depth data D is less sensitive to illumination changes, therefore, we use depth images for background subtraction. Among numerous techniques of the background subtractions, we adopt Gaussian Mixture Model (GMM) [23] because this technique have been shown to be the best suitable for our system.

Let us denote depth frame D_t at time t: The background model at a pixel $p(i,j)$ has two parameters $\lambda_{pBG} = (\mu_p, \eta_p)$ which are mean and noise, respectively. Because of stability of the depth feature, our background model contains only a Gaussian. Given a depth sequence including n frames, an observation of a pixel p along temporal dimension is denoted by: $s_p = [p_0(i,j), p_1(i,j), ..., p_n(i,j)]$. Each signal s_p contributes to background model by conditions below:

$$\begin{cases} \mu_{BG}(i,j) = & \frac{\sum_{i=0}^{n} D_t(i,j)}{n} & \text{if } std(s_p) < \tau \\ \mu_{BG}(i,j) = & mean(kmeans(s_p, 2)) & \text{otherwise} \end{cases} \tag{4}$$

and

$$\begin{cases} \eta_{BG}(i,j) = & 0 & \text{if } std(s_p) < \tau \\ \eta_{BG}(i,j) = & 255 & \text{otherwise} \end{cases} \tag{5}$$

According to (4) and (5), the parameter τ aims to evaluate how a signal s_p is stable. The parameters $(\mu, \eta)_{BG}$ are learnt with every pixel of a depth frame D_t. Utilizing results of the learnt background mode μ_{BG} and η_{BG}, a pixel p belongs to a body region through evaluating a probability P as given in (1) or more simply when the condition below is ensured:

$$B_{p,t} = |D_{p,t} - \eta_{BG}(p) - \mu_{BG}(p)| > 0 \tag{6}$$

Figure 2(a-c) shows results of the background subtraction using (6). Given a region of human body (as shown in Fig. 2(c)), we continuously extract candidates of the hand.

C. Estimating the distance from hand to the Kinect sensor for extracting hand candidates

As denoted in the second assumption, human's hand is raised in front of the body during times of the controlling. Candidates of the hand can be extracted from the body regions B by evaluating the distances following (2). In (2), an important parameter is $thresh_{hand}$ which can be learnt by following procedure.

We firstly built a histogram of depth data of the only body regions (B regions). Intuitively, there are two local peaks in this histogram. One peak covers a range from the Kinect sensor to hand, and another peak represents depth data from the Kinect sensor to other body parts. Two peaks are separated by $thresh_{hand}$ which an end-user is asked to move his/her hand in few times. This trick elicits hand regions in consecutive frames because area of the hand would have to be strongly fluctuated. The moving parts are calculated using differences between consecutive depth frames D_{t-2}, D_{t-1}, D_t by:

$$\begin{cases} D_{t-2,t-1}(i,j) = D_{t-1}(i,j) - D_{t-2}(i,j) \\ D_{t-1,t}(i,j) = D_t(i,j) - D_{t-1}(i,j) \\ D_{hand}(i,j) = (D_{t,t-1}(i,j)) \cap (D_{t-2,t-1}(i,j)) \end{cases} \tag{7}$$

Figure. 3 illustrates how to detect the moving parts from three consecutive frames with fixed position of an end-user. Fig. 3(d) shows differences of depth between frames Fig. 3(a) and Fig. 3(b), whereas moving parts between Fig. 3(b) and Fig. 3(c) are shown in Fig. 3(e). An union operator and following by a binary threshold one give hand regions, as shown in Fig. 3(f). Histogram of the hand regions then is calculated by:

$$H_{hand} = \frac{\sum_{(i,j)\in(w,h)}^{w\times h} \delta\{p_{hand}(i,j)\}}{n} \tag{8}$$

Figure. 3(g) shows a depth histogram of the hand regions in left panel. It is built from results of the detecting moving parts, whereas the histogram of other body parts is shown in right panel. Obviously, the parameter $thresh_{hand}$ is identified. By using $thresh_{hand}$, an intersection histogram operator is applied to B, D to eliminate body parts. However, few candidates of the hand H are existing, as shown in Fig. 2(d). We then select and precise hand regions using appearance-based features such as skin color of the hand.

D. Estimating parameters for pruning hand regions

The hand detection results as shown in Fig. 2(d) often consist of contaminated regions. They are the same distance to a true hand region. Moreover, because low resolution of

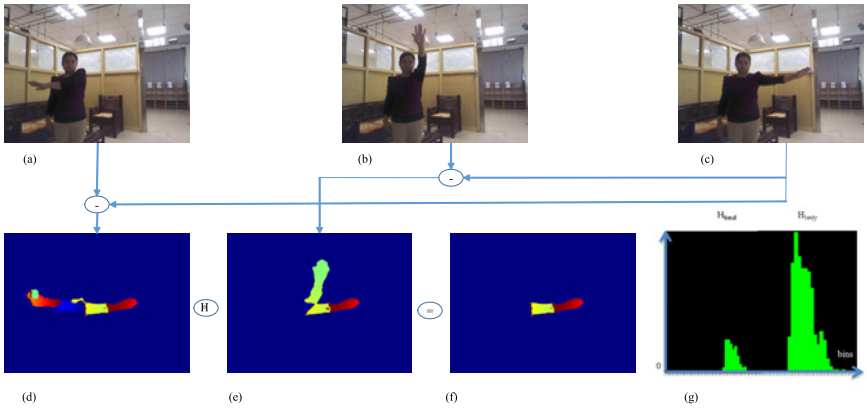


Fig. 3. Result of the learning distance parameter. (a-c) Three consecutive frames; (d) Results of subtracting two first frames; (e) Results of the subtracting two next frames; (f) Binary thresholding operator; (g) A range of hand (left) and of body (right) on the depth histogram

the depth image, the extracted hand regions do not precisely cover area of hand where are boundary of fingers. We propose to use the color distribution of hand skin to correct the hand area. There are many approaches that utilizes color features to segment hand regions (e.g., [24] [12]). Our works are inspired from [24] for learning a model of hand skin color. According to [24], such model (denoted Ω_c) is characterized by two parameters: mean μ_{skin} and covariance δ_{skin} (e.g., by utilizing color vector $[R, G, B]$). In order to estimate (μ_{skin}, δ_{skin}), we design a learning procedure as below.

Firstly, the end-user requires to raise his/her hand so that it is located in a pre-determined rectangle box. For instance, Fig. 4(a,b) shows red boxes. The pixels inside the center regions of these boundary boxes (marked in yellow box in Fig. 4(a-b)) are selected. Feature vectors $[R, G, B]$ of the selected pixels are transformed to HSV color space. We then extract skin color features by using a skin map as defined in [12]. Those features (S_{skin}) are calculated based on HSV color space.

The difference of two feature vectors are calculated by two consecutive frames. Which histograms of S_{t-1} and S_t is compared by a histogram correlation standard ϵ_t as the formula below:

$$\epsilon_t = \frac{\sum_i (S_{t-1}(i) - \bar{S}_{t-1})(S_t(i) - \bar{S}_t)}{\sqrt{\sum_i (S_{t-1}(i) - \bar{S}_{t-1})^2 \times \sum_i (S_t(i) - \bar{S}_t)^2}} \quad (9)$$

A learning time will be stop when $\frac{1}{N}\sum_{t=0}^{N} \Delta_t < threshold$ (N is number of RGB images use for the learning). Fig. 4(c) shows accumulation values of Δ_t along learning time. As shown, the learning procedure is converged after N frames. Parameters $(\mu_{skin}, \delta_{skin})$ are calculated from such sample data.

For pruning candidates of hand, if a $blob[i]$ is large enough, we utilize above color model to verify that's true hand region or not. A Region-of-Interest ($ROI1$) at center of $blob[i]$ (as marked in yellow in Fig. 5(a)) is taken by: $ROI1 = blob[i]/\delta$ in which ($\delta > 1$ is a scale coefficient. A Mahalanobis distance on [R,G,B] data between $ROI1$ region and skin model is calculated. The parameters $\mu_{skin}, \delta_{skin}$ of the skin color model associate to measure similarity of a pixel to the hand skin colors. A ratio between number of the positive pixels (whose

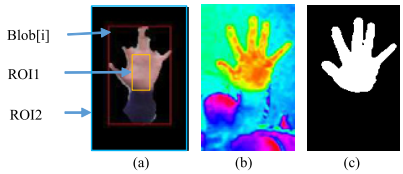


Fig. 5. Results of the hand segmentation. (a) a Candidate of hand; (b) Mahalanobis distance; (c) Refining the segmentation results using RGB features

Mahalanobis distances are large enough) per size of $ROI1$. $blob[i]$ is decided as a true hand region if the ratio is a high value.

For pruning the segmentation results, we create a larger region than original blob[i] by: $ROI2 = blob[i] \times \delta$ (as marked in the light-blue box in Fig. 5(a)). Then a Mahalanobis distance between $ROI2$ and skin model is calculated in order to extract fully hand skin pixels Fig. 5(b). Obviously, hand pixels were updated from original true hand shown in Fig. 5(a). Efficiency of this procedure is shown in Fig. 5(c).

E. Hand posture recognition

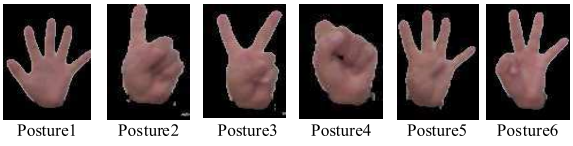
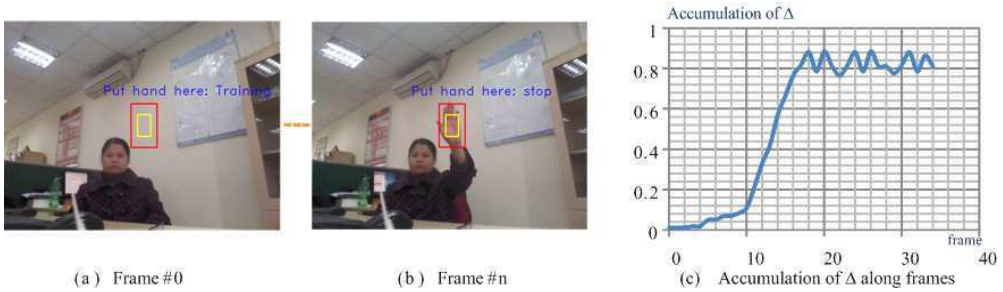


Fig. 6. Left to right: Samples of six hand postures

In this work, we utilize X-Kin library proposed in [25] for recognizing hand posture. Although this is a simple implementation, our main purposes are that even utilizing such algorithms, we successfully recognize hand postures by deploying the proposed an user-guide scheme. To describe a hand posture, we extract hand contours from the segmentation results (as shown in Fig. 5). Then a Fourier transform is applied on points of the hand contour. Each posture is presented by a matrix F contains M rows of the training images by N FFT coefficients per each row. K-Nearest Neighbor (K-NN)



(a) Frame #0 (b) Frame #n (c) Accumulation of Δ along frames

Fig. 4. Result of the learning skin color model

classifier is trained with a data set including 7 hand postures as shown in Fig. 6. To classify a posture, mean and covariant of the matrix F are calculated. We found K cases in the training set closest to the query frames by calculating a Mahalanobis distance between present contour and mean and covariant factors for each posture. A decision of whether it was a label based on the majority vote of the K postures found.

IV. EXPERIMENTAL RESULTS

We evaluate the proposed method in term of the training time versus the accuracy-real time performance. The proposed frame work is warped in a C++ program. The evaluation procedures are implemented on a notebook PC 2.00GHz CPU, 2GB RAM. The Kinect sensor [3] is mounted on a tripod with fixed position and captures data at 15 fps. Six persons implement the proposed learning scheme as well as evaluate performance of the hand posture recognition.

A. Evaluation on training time for end-users

The training background model is same procedure for all trainees. Therefore, this procedure is run one time only. A training time requires ∼2 seconds (30 frames). In order to train the distance from the Kinect sensor to hand parts, trainees raise their hand in few times (∼2 seconds). The most consuming time is to train parameters of the skin color model. We count a duration for each evaluator from starting time when a his/her hand is located and the system automatic stops when the parameters are converged. Table I shows the required time of this step. Totally, the required time per each evaluator is 15±2 sec for the whole training procedures. That is not too long time to train for a effection which is evaluated by the next part.

TABLE I. THE TIME TO LEARN PARAMETERS OF THE HAND-SKIN COLOR MODEL

Criteria	Person1	Person2	Person3	Person4	Person5	Person6
$\sum Frames$	36	24	21	23	31	28
Learning time (sec)	9.2	7.4	7.0	7.3	8.5	7.9
Avg. ± std	8.3 ± 1.2 sec					

B. Hand segmentation evaluation

Figure 7. illustrates the segmented hands that's extracted by image sequences of six end-users. As shown, our proposed method extracts correctly hands with different lighting conditions as well complexity of the background. For quantitative evaluation, we use Jaccard Index that is calculated by:

$$JI = \frac{HDT \cap HGT}{HDT \cup HGT} \quad (10)$$

Where HDT is region of the hand that's segmented by the proposed method. HGT is ground-truth one. The segmentation result is better if the JI index is more closed by 100%. The computational time is also reported. Table II shows JI indexes without/with learning scheme. Obviously, by paying a cost to learning parameters of the skin color model, JI indexes are significantly improved from 63.4% to 86.9%.

TABLE II. RESULTS OF THE JI INDEXES WITHOUT/WITH USER-GUIDE SCHEME

Criteria	Person1	Person2	Person3	Person4	Person5	Person6
$\sum Frames$	128	138	124	116	157	162
Without user-guide scheme						
$JI(\%)$	56.7	52.4	72.1	57.2	68.4	73.8
Avg. ± std	63.4 ± 8.3 %					
With user-guide scheme						
$JI(\%)$	86.2	89.1	85.8	82.4	90.4	87.2
Avg. ± std	86.9 ± 2.6 %					

C. Posture recognition

We evaluate accuracy-real time performance of the proposed system. Seven static hand postures as shown in Fig.

TABLE III. RESULTS OF THE HAND POSTURE RECOGNITION

PostureID	Criteria	Person1	Person2	Person3	Person4	Person5	Person6
Pos 1	Accuracy	90	82.5	89.5	91.8	90.3	88.2
	Time cost	0.57	0.52	0.48	0.51	0.53	0.49
Pos 2	Accuracy	90	88.6	86.7	85	89	82
	Time cost	0.54	0.51	0.59	0.5	0.55	0.51
Pos 3	Accuracy	90	87.4	87.1	81.2	88.3	78.2
	Time cost	0.61	0.6	0.59	0.54	0.56	0.54
Pos 4	Accuracy	96.7	86.7	88	92.7	93.5	90.4
	Time cost	0.51	0.53	0.49	0.55	0.53	0.54
Pos 5	Accuracy	96.7	86.7	88	92.7	93.5	90.4
	Time cost	0.51	0.53	0.49	0.55	0.53	0.54
Pos 6	Accuracy	96.7	86.7	88	92.7	93.5	90.4
	Time cost	0.51	0.53	0.49	0.55	0.53	0.54
Avg±Std	Accuracy	88.1 ± 4.1 (%)					
	Time cost	0.54 ± 0.1 (sec.)					

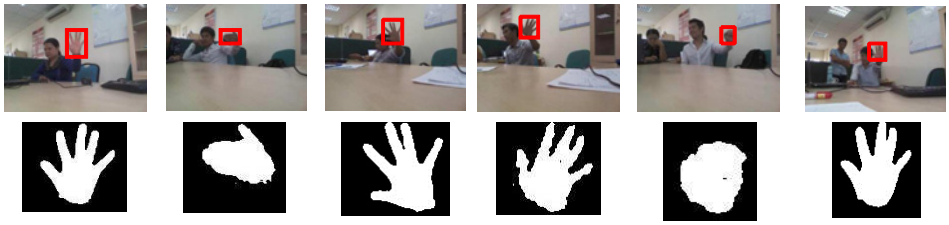


Fig. 7. Some results of the segmented hands on sequences captured image

6 are examined. To build matrix F in Sec. III.D, in the learning phrase, we collect 8 samples for each posture. In the testing phrase, 20 images are collected per each evaluator. Accuracy score is ratio between numbers of the true detection per total postures. The second one is time costs to implement fully steps of the flow-work in Fig. 1(a). Results are given in the Table III. Averagely, the proposed method obtains the accuracy rate at 88.1 $\pm$ 4.1 %, whereas it requires only 0.5 sec/frame for computational time. Performance of the accuracy is comparable with current state-of-art works (e.g., [12] achieved 93%; however its time cost was 2.7 sec/frame).

V. CONCLUSIONS

This paper described a vision-based hand posture recognition system. Our proposed method paid many attentions in a guiding scheme to end-users. This scheme shown successfully in term of robustness and real-time criteria. The learning scheme was designed to focus on estimating heuristic parameters. The proposed method was relatively simple and required a fast learning time. Therefore, it is feasible to deploy practical application, such as to control TV or lighting system in indoor. In the future, we will continue research on learning parameters of the dynamic hand gesture recognition as well as evaluating users behavior in the learning phrase.

VI. ACKNOWLEDMENT

The research leading to this paper was supported by the National Project B2013.01.41 "Study and develop an abnormal event recognition system based on computer vision techniques".

REFERENCES

[1] X. Zabulis, H. Baltzakis, and A. Argyros, *Vision-based Hand Gesture Recognition for Human Computer Interaction*. Lawrence Erlbaum Associates, 2009.

[2] S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: a survey," *Artificial Intelligence Review*, Nov 2012.

[3] M. K. for Windows, "http://www.microsoft.com/en-us/kinectforwindows." Nov 2013.

[4] P. Company. (2013) Pointgrab brings gesture control to home appliances. [Online]. Available: http://www.cnet.com/news/pointgrab-brings-gesture-control-to-home-appliances/

[5] P. Qifan, S. Gupta, S. Gollakota, and S. Patel, "Whole-home gesture recognition using wireless signals," in *Proceedings of the 19th Annual International Conference on Mobile Computing and Networking*, 2013.

[6] J. Choi and B. Seo, "Robust hand detection for augmented reality interface," in *Proceedings of the 8th International Conference on Virtual Reality Continuum and its Applications in Industry*, 2009.

[7] F. Picard and P. Estraillier, "Motion capture system contextualization application to game development," in *Proceedings of Computer Games: AI, Animation, Mobile, Interactive Multimedia, Educational, Serious Games*, 2009.

[8] Q. Chen, C. Joslin, and Georganas.N.D., "A dynamic gesture interface for virtual environments based on hidden markov models," in *Proceedings of IEEE International Workshop on Haptic Audio Visual Environments and their Applications*, 2005.

[9] M. Silva, V. Courboulay, and A. Prigent, "Gameplay experience based on a gaze tracking system," *EURASIP Journal on Applied Signal Processing*, 2007.

[10] A. Mittal, A. Zisserman, and P. Torr, "Hand detection using multiple proposals," in *Proc. of International Conference on British Machine Vision Conference*, 2011.

[11] F. Chen, C. Fu, and C. Huang, "Hand gesture recognition using a real-time tracking method and hidden markov model," *Image and Vision Computing*, vol. 21, Aug 2003.

[12] P. Pisharady, P. Vadakkepat, and A. Loh, "Attention based detection and recognition of hand postures against complex backgrounds," *International Journal Computer Vision*, vol. 101, Aug 2012.

[13] B. Ionescu, D. Coquin, P. Lambert, and V. Buzuloiu, "Dynamic hand gesture recognition using the skeleton of the hand," in *Proceedings of Annual Conference on Communication by Gaze Interaction*, 2007.

[14] M. Bergh, F. Bosch, E. Koller-Meier, and L. Van-Gool, "Haarlet-based hand gesture recognition for 3d interaction," in *Proceedings of the Workshop of Applications of Computer Vision*, 2009.

[15] M. Bergh and L. Van-Gool, "Combining rgb and tof cameras for real-time 3d hand gesture interaction," in *Proceedings of the Workshop of Applications of Computer Vision*, 2011.

[16] M. Park, M. Hasan, J. Kim, and O. Chae, "Hand detection and tracking using depth and color information," in *Proceedings of IPC*, 2012.

[17] P. Doliotis, V. Athitsos, D. Kosmopoulos, and S. Perantonis, "Hand shape and 3d pose estimation using depth data from a single cluttered frame," in *Proceedings of the International Symposium on Visual Computing*, 2012.

[18] P. Viola and M. Jones, "Robust real-time object detection," in *Proceedings of IEEE Workshop on Statistical and Computational Theories of Vision*, 2001.

[19] T. Nguyen and D. Nguyen, "An active boosting-based learning framework for real-time hand detection," in *Proceedings of International Conference on Automatic Face and Gesture Recognition*, 2008.

[20] J. Park and Y. Choi, "On-line learning for active pattern recognition," 1996.

[21] N. Oza and S. Russell, "Online bagging and boosting," in *Proceedings Artificial Intelligence and Statistics*, 2001.

[22] D. Herrera, J. Kannala, and J. Heikkila, "Joint depth and color camera calibration with distortion correction," 2012.

[23] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," in *Proceedings of Computer Vision and Pattern Recognition*, 1999.

[24] M. Jones and J. Rehg, "Statistical color models with application to skin detection," in *Proceedings of IEEE conference on computer vision and pattern recognition*, 1999.

[25] F. Pedersoli, N. Adami, S. Benini, and R. Leonardi, "Xkin - extendable hand pose and gesture recognition library for kindest," in *Proceedings of ACM Conference on Multimedia*, 2012.