Robust Hand Tracking Using a Simple Color Classification Technique

Miaolong Yuan, Farzam Farbiz, Corey Mason Manders and Tang Ka Yin

*A*STAR Institute for Infocomm Research,*
1 Fusionopolis Way, #21-01, Singapore, 138632

***Abstract*— Skin color is a strong cue in vision-based human tracking. Skin detection has been widely used in various applications, such as face and hand tracking, people detection in the video databases. In this paper, we propose and develop an effective hand tracking method based on a simple color classification. This method includes two major procedures: training and tracking. In the training procedure, the user specifies a region on a hand to obtain the training data. Based on the skin-color distribution, the training data will be classified into several color clusters using randomized list data structure. In the hand tracking procedure, the hand will be segmented in real-time from the background using the randomized lists that have been trained in the training procedure. The proposed method has two advantages: (1) It is fast because the image segmentation algorithm is automatically performed on a small region surrounding the hand; and (2) It is robust under different lighting conditions because the lighting factor is not employed in our effective color classification. Several experiments have been conducted to validate the performance of the proposed method. This proposed method has good potentials in many real applications, such as virtual reality or augmented reality systems.**

***Index Terms*— Color classification, hand segmentation, human computer interaction, real-time tracking.**

I. INTRODUCTION

In virtual reality (VR) or augmented reality (AR) applications, mice, keyboards, joysticks are commonly used devices for human computer interaction. The drawback of such devices is that they lack the flexibility due to spatial constraints from the environments. For example, in a large-scale collaborative product design AR platform, each user is allowed to interact with the virtual product prototype. It is obviously unrealistic to furnish each user a mouse and a keyboard to interact with the virtual prototype in the large-scale AR environment. However, vision-based techniques enable such interaction operations possible by locating one or more cameras looking at the users and analyzing the movements of the users' hand. Vision-based gesture interfaces have a great potential in many interactive systems, with which user inputs can be easily achieved. A key issue is how to separate the hand from the complex background.

In general, the image features or color information often offers different cues for hand tracking. Zhang et al. [15] developed an edge-based dynamical programming method to track a tip pointer for a gesture interface system, which employs an arbitrary quadrangle–shaped panel and a tip pointer as an intuitive input device. To track the tip point, such as fingertip, and the quadrangle–shaped panel, the backgrounds need to be relatively simple. When the tip pointer moves rapidly, the tracking may be lost and re-initialization is required. Furthermore, the edge-based tracking methods are computationally expensive and are prone to causing ambiguities when the backgrounds are complex.

Compared to the edge-based methods, hand tracking methods using color information are more effective. Skin color is a strong cue for vision-based hand tracking, as the skin colors distribute in a small region in a color space and have more difference in intensity than in color. Existing skin-color tracking can be grouped into two categories: parametric methods and non-parametric methods.

Parametric approaches represent the skin-color density in parametric forms, such as the Mixture of Gaussian [12, 10, 16]. Typical Expectation-Maximization (EM) method is often used to fit the probabilistic models. However, the parameters of the skin-color distribution can vary significantly with different people under various lighting conditions. Hence, the major problem is that there is not enough prior knowledge to determine the model order of the Mixture of Gaussian.

The key idea of the non-parametric approaches is to estimate the color distribution from the training data using statistical probability models. One of the non-parametric approaches is based on color histogram. Kjeldsen and Kender [6] used histogram-like structure to segment hand for a gesture-based user interface. However, the system only operated in office environments with typical office lighting conditions and relied on the assumption that skin coloration was relatively unique in the target environment. Sigal et al. [11] proposed a novel histogram-based method using an explicit second order Markov model to predict the evolution of the skin-color histogram over time. Histograms will be dynamically updated based on the current segmentation and predictions of the Markov model. However, the scene changes are not well modeled by this method and the system is only effective for slowly changing dynamic scenes. The method will fail under dramatic or abrupt changes in the scene. In general, histogram-based tracking methods work well when the histograms well quantified and when sufficient training data is available. However, it is

Manuscript received on 30 June 2008
E-Mail: myuan@i2r.a-star.edu.sg

non-trivial to select a good quantization level of the color histogram.

Mean Shift is a robust non-parametric skin-color segmentation method based on region matching [4]. It is used as a way to converge from an initial guess for location and scale to the best match based on the color histogram similarity. CamShift is a continuously adaptive Mean Shift, which detects the mode in the probability distribution image by applying mean shift and dynamically adjusts the parameters of the target distribution [1]. This method is fast and moderately accurate. The accuracy can be possibly improved by using a different color representation. However, there are quite a few parameters, such as the number of histogram bins, the minimum saturation and minimum and maximum intensity, which are not easily to be determined.

Another popular hand tracking method is based on the color classification using related classification techniques [13, 14]. Wu et al. proposed a color segmentation scheme by approximating the color distribution of an image in the HSI color space using a self-organizing map (SOM) in which each output neuron corresponds to a color cluster [13]. However, when the lighting condition changes dramatically, their color segmentation method may fail. In [9], Ong and Bowden employed clustering methods to cluster the training data into similar shapes based on skin-color distribution and build a classifier tree for detection. A database of images is first clustered using a k-means clustering algorithm. After which, a tree structure of boosted cascades is then constructed. The classifier tree is subsequently used to recognize particular gestures. This method is suitable when both training and test databases have fairly simple and similar backgrounds. Some other researchers use face detection results to track hands, as faces and hands have a similar skin-color distribution and the face can be automatically detected using advanced computer vision techniques [2, 7, 8]. Nickel and Stiefelhagen [2] presented a gesture system that uses both face and hand positions for the interaction. Manders et al. presented a robust hand tracking method based on CamShift and a joint probability function [7]. However, in these methods, the hand can be tracked once the face is first detected. This is obviously not practical in some applications.

The purpose in this paper is to propose a simple, but effective color classification technique using randomized lists. This method includes two major procedures: training and tracking. In training, a region is specified on a hand to obtain the training data. Based on the skin-color distribution, the training data will be classified into several color clusters using randomized lists. In tracking, the human hand is segmented from the background in real-time using the trained randomized lists. The method is fast and effective under different lighting conditions.

The remaining sections of this paper are organized as follows: Section 2 describes the mechanism of the color classification. Section 3 describes the details of the real-time hand segmentation method. Experimental results are presented in Section 4 to validate the performance of the proposed method. Conclusions and future work are given in the last section.

II. Color Classification

2.1 L*a*b* Color Space

Skin color segmentation depends on not only the segmentation approaches, but also the color distribution in different color spaces. In most non-parametric skin-color tracking methods, the HSI color space is used, as the color distribution in the HSI color space is more concentrative that in the RGB color space. Different form the RGB and HSI color spaces, the L*a*b* color space is used in the proposed method, where L* represents lightness and a*, b* are the coordinates of chromaticity. It is nearly proportional with visual perception, which means that equal distances in the color space correspond to equal perceived color differences. The values of the L*, a* and b* are computed using the following equations [3]:

$$L^* = \begin{cases} 116\left(\frac{Y}{Y_n}\right)^{1/3} - 16 & if\ \frac{Y}{Y_n} > 0.008856 \\ 903.3\left(\frac{Y}{Y_n}\right)^{1/3} - 16 & f\ \frac{Y}{Y_n} < 0.008856 \end{cases} \quad (1)$$

$$a^* = 500\left[f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right] \quad (2)$$

$$b^* = 500\left[f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right] \quad (3)$$

where

$$f(t) = \begin{cases} t^{1/3} & t > 0.008856 \\ 7.787 * t + 16/116 & otherwise \end{cases} \quad (4)$$

where X_n, Y_n and Z_n are the CIE XYZ tristimulus values of a perfect reflecting diffuser. In our approach, they are set as 250.155, 255.000, and 301.410, respectively. X, Y, and Z are the tristimulus values which are computed from R, G, B information of each pixel based on the following Equations:

$$X = 0.607 * R + 0.174 * G + 0.2 * B \quad (5)$$

$$Y = 0.299 * R + 0.587 * G + 0.114 * B \quad (6)$$

$$Z = 0.066 * G + 1.116 * B \quad (7)$$

2.2 Skin-Color Classification

The purpose of the skin-color classification-based hand segmentation approach is to classify the hand color distribution into several color clusters using randomized list data structure. In most color classification-based tracking methods, the number of the clusters should be specified in advance. Hence, the success of the classification technique depends on the specified number of clusters. In this research, the classifier is represented by randomized list data structure. Using a randomized list data structure, the number of the cluster will be automatically determined, thus making the method more effective. In the proposed method, each color cluster consists of the pixels which have the same characteristics defined by four parameters: class C, weight vector ω, threshold λ and pattern count t. ω

represents the set of weighted connections between the clusters and each of the input signals. λ describes a hyper-spherical region of influence around the clusters in the L*a*b* color space. *t* indicates the number of times that a color cluster has responded to the input color signals obtained from the training data.

As mentioned in Section 2.1, the skin-color distribution is represented in the L*a*b* color space. The input vector, which is usually defined by a feature vector for color classification, represents a characteristic of the incoming pattern and is of importance to the classification problem. In the proposed method in this research, it will significantly affect the classification performance, such as robustness to the lighting conditions. The input vector in this research is defined by the a* and b* components of each pixel, which is represented in the L*a*b* color space. From the experiments that we have conducted, the a* and b* components of the training data is sufficient and effective for the color classification problem. Hence, the L* component, which represents the lightness information in the L*a*b* color space, is not employed for training and tracking. This makes the hand tracking method robust under different lighting conditions. In response to an input signal, i.e., an input vector of each pixel, a distance between the input vector and its corresponding weight in each color cluster will be computed based on the Euclidean distance function. For the first input signal presented to the training procedure, a new color cluster will be created first and this input signal will be loaded as a weight vector of this new color cluster, and the pattern count in this color cluster is set to 1. During training, an input signal will be included in a color cluster i if it falls into this cluster, i.e., its distance in this cluster is less than a pre-defined threshold λ. Subsequently, the pattern counter is incremented by 1. Otherwise, a new color cluster will be created and the current input vector is loaded as a weight vector of this cluster.

The color classification mechanism includes the following steps, as illustrated in Fig. 1.

1) Specify the related region(s) in the hand to be trained.
2) Use the specified region to obtain the training data set in the L*a*b* color space.
3) For an input color signal *X*, calculate the distance $d_i = \sqrt{\sum_{j=1}^{2}(\omega_{ij} - x_j)^2}$. If it does not fall into any existing clusters, a new cluster *i* will be created and *X* is set as the weight vector the cluster *i*. This new cluster is then stored at the end of a randomized list data structure.
4) For an input color signal *X*, if it falls into a cluster *i*, the pattern counter of this existing cluster is incremented by 1. The information of this cluster is further updated in the list structure.

In response to an input color signal *X*, related probability response model are commonly used, in which each cluster will output a probability value to determine which cluster will be activated for the incoming input signal. However, in this research, a fast response mode is used, in which the classifier computes the distance based on a radial basis function i.e. $d_i = \sqrt{\sum_{j=1}^{2}(\omega_{ij} - x_j)^2}$) between the input signal and the weights in the color clusters, and then directly compares whether this distance is less than the predefined threshold, that is:

$$p_i = \begin{cases} 1 & if\ d_i < \lambda_i \\ 0 & if\ d_i \geq \lambda_i \end{cases} \quad (8)$$

If d_i is less than or equal to a pre-defined threshold of a color cluster, the cluster will become active to trigger its associated color class C. Otherwise, the cluster will not respond this input signal.

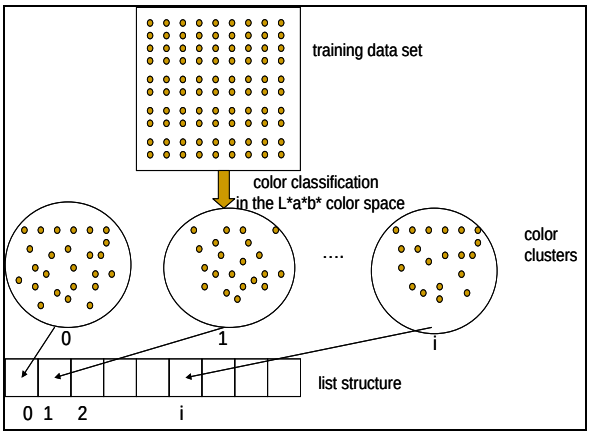


Fig. 1. Color classification

III. REAL-TIME HAND SEGMENTATION ALGORITHM

3.1 Flowchart

For each incoming frame in a live video, the system can directly identify the pixels which may belong to the hand using the randomized lists-based classifier that is obtained during the training procedure. Next, some post-processing procedures, such as pixel labeling, group connectivity, are required to accurately localize the hand from the background in the video. The flowchart is shown in Fig. 2.

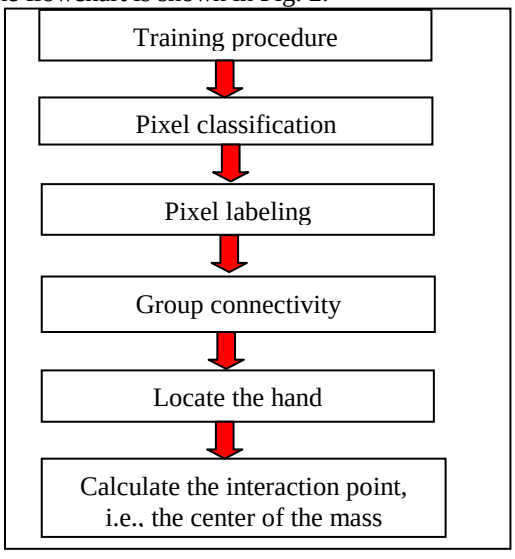


Fig. 2. Flowchart of the hand segmentation

3.2 Initialization

At the beginning, the user is required to specify one or more regions on a hand to obtain the training data to generate a color classifier. The classifier is stored into a randomized data structure. The user is only required to execute the training procedure once. When the training procedure is completed, the training results will be saved automatically as a data file. Subsequently, when the user initiates the hand tracking algorithm using the similar physical environments, the system will automatically load the training results and execute the hand tracking procedure. This provides the advantage of automatic initialization and save processing time.

3.3 Hand Localization

Given an incoming frame from a live video, the key problem is how to accurately localize the hand region. For the first incoming frame, the system acquires the entire color images to be segmented and convert them into the L*a*b* color space. The a* and b* components of each pixel is fed into the randomized lists-based classifier. The randomized lists are used to directly label which pixels belong to the hand. According to Equation (8), the pixels with the output value 1 represent the hand segmentation result. After the pixel labeling process, some standard growing schemes, such as [13] are used to group the labeled pixels into several connected regions. At this stage, there may be many regions to be segmented which includes a large number of small regions from the background which is similar to the hand color distribution. The region with the largest number of the labeled pixels is retained as the hand region. Finally, a feature point, called the interaction point, will be further extracted, which is used an input device. In our system, we use the center of the tracked hand region. This interaction can be directly used to obtain user input for human computer interaction. From the second incoming frame, the system acquires a bounded region surrounding the tracked hand in the previous frame, and repeats the same processes to segment the hand, thus making the segmentation faster. More details will be addressed in Section 4.

3.4 Refining the training data

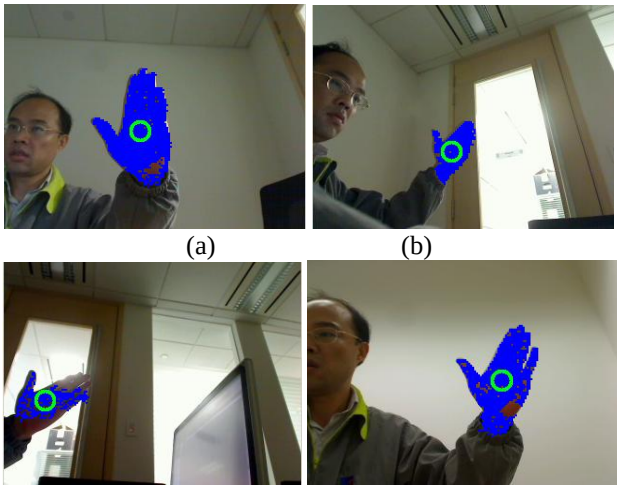
When the user specifies region(s) on the hand to obtain the training data, the selected region(s) may include some pixels which do not belong to the hand, as shown in Figure 3(a). For such a case, some of the generated color clusters may corresponds to those pixels which do not belong to the hand region. However, based on our observation in the experiments that we have conducted, the pattern numbers of these color clusters are often small. Hence, in our system, these color clusters will be pruned according to a given threshold and the trained randomized lists will be re-organized. This refinement process gives the users more flexibility to select a rough large area to obtain the training data. Even though the selected regions include some small areas of the background, the method can still segment the hand accurately. For example, in Fig. 3(a), the specified region includes some small parts of the ceiling and the cubicle, as indicated by the two blue circles. The hand can still be reliably tracked using the refined training results, as shown in Fig. 3(b). Otherwise, the parts of cubicle and the ceiling (from the background) will also be segmented.

(a) (b)

Fig. 3. Example of refining the trained randomized lists

IV. EXPRIMENTS

The proposed hand tracking method has been implemented using visual C++ and several experiments have been carried out to verify this method. Any general web cameras can be used to capture the video sequence. The image size in our testing is 320×240. The user is first required to specify a region on the user's hand to obtain the training data in the L*a*b* color space. The system will execute the training procedure to obtain a color classifier using randomized lists. Next, the trained randomized lists are used for hand segmentation in the live video. The above segmentation method can be used to track the hand in real-time, as the randomized lists-based classifier will be automatically performed on a small region surrounding the hand. We can assume continuity of the position of the hand during tracking. We have tested the speed of tracking the hand, as shown in Figure 4. The average elapsed time for tracking the hand is 0.03147s, which can meet the real-time performance for real applications. Fig. 4 shows some hand tracking results in a live video. The green circles indicate the centers of the mass of the tracked hand region which can be used for interaction operations. In this experiment, the hand moved arbitrarily, such as very fast movement, or the camera was handheld by a user and had also been moved through large changes of viewing angles and volumes. Furthermore, the lighting conditions had also been changed. These results show that the hand can be accurately segmented in real-time under large rotation, scale changes, and various lighting conditions.



(a) (b)

(c) (d)

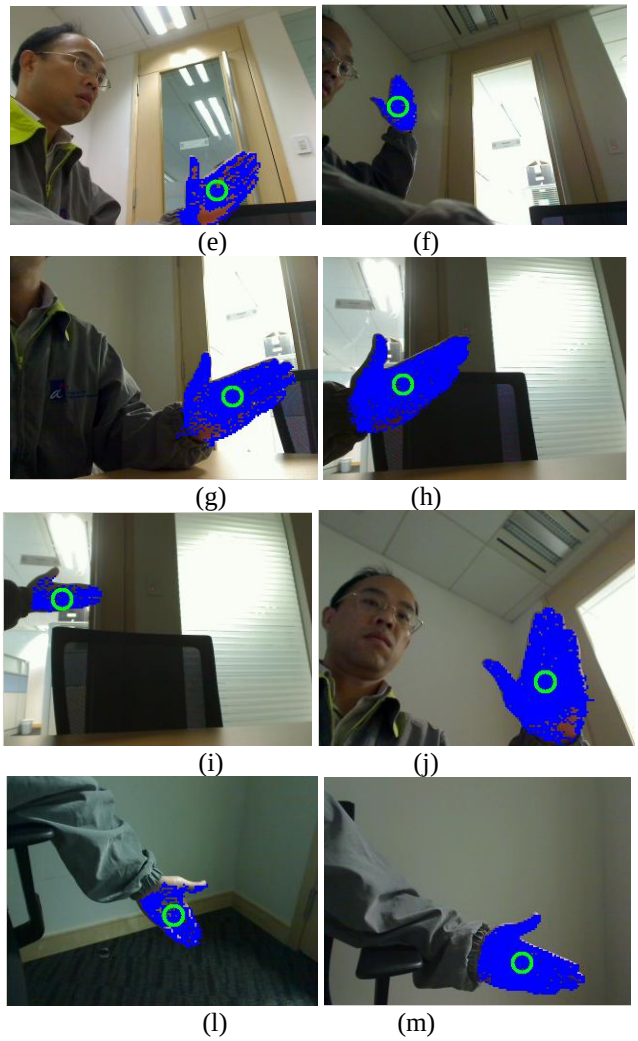


Fig. 4. Examples of the tracking method

Beside the effectiveness of the proposed method, there are two major issues addressed below to further show the robustness of the proposed technique:

(1) *Rapid movement*

Normally, the user moves his/her hand at a normal speed. However, when the use moves his/her hand very fast, the tracking results may be lost. Thus, the previous interaction point is preserved and the user can move his/her hand near the position of the previous interaction point to easily re-track the hand without any manual re-initialization.

(2) *Lighting condition*

In the proposed method, only the a* and b* components in the L*a*b* color space are used for training and hand tracking. This makes the tracking robust under various lighting conditions. Figure 5 shows some tracking results under different lighting conditions from a live video. In fact, in many real applications, only part of the hand is required to be tracked to extract the interaction point. For example, in a virtual keyboard system, it is sufficient if only part of the hand is tracked for interaction purposes. As long as at least one point of the hand is visible and is detected (considering a worse case, for example a sudden lighting change), this point can still be used for interaction operations.

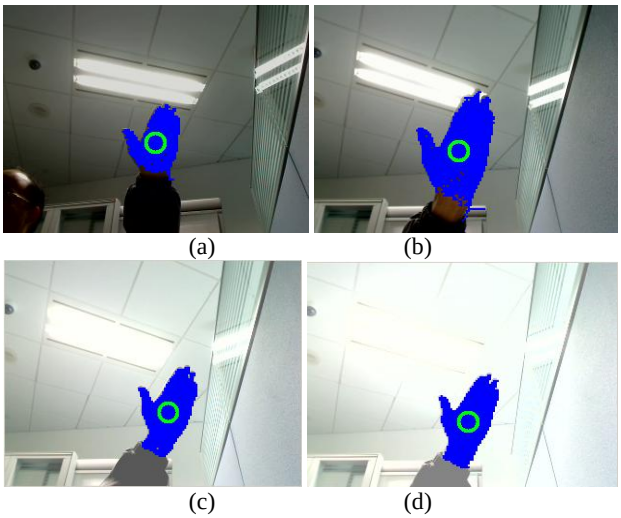


Fig. 5. Tracking results under different lighting conditions

V. CONCLUSIONS

In this paper, a fast and effective hand tracking method is proposed based on a simple color classification technique using randomized lists. The method is simple to operate by moving the user's hand freely for interactive operations. Another advantage is that the proposed method is robust under different lighting conditions. Some experiments have been conducted to validate that the proposed method is robust and stable. Potentially, the proposed hand tracking method can be used as a wireless mouse, keyboard, or a remote control. Examples are video games in VR or AR applications. In future, more research and work will be conducted to apply this method in the real VR or AR-based game applications.

REFERENCES

[1] G. R. Bradski. Computer video face tracking for use in a perceptual user interface, *Intel Technology Journal*, Q2, 1998.

[2] N. C. Cabral, H. Carlos H. Morimoto and K. Z. Marcelo. On the usability of gesture interfaces in virtual reality environments, In *Proceedings of the Latin American conference on Human-computer interaction*, 100-108, 2005.

[3] CIE. 1986. Colorimetry, second edition. Vienna, Austria. Publication CIE No. 15.2.

[4] D. Comaniciu, V. Ramesh. and P. Meer P. Real-time tracking of Non-rigid objects using Mean Shift, In *Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, pp. 142-149, 2000.

[5] M. Jones M. and J. Rehg. Statistical color models with applications to skin detection, *Compaq Cambridge Research lab*, TP CRL 98/11, 1998.

[6] R. Kjeldsen. and J. Kender. Finding skin in color images. *International Conference on Automatic Face and Gesture Recognition*, pp. 312–317, 1996.

[7] C. Manders, F. Farbiz, J. H. Chong, K. Y. Tang K, G. G. Chua and M. H. Loke. Robust hand tracking using a skin tone and depth joint probability model, *International Conference on Automatic Face and Gesture Recognition*, 2008.

[8] K. Nickel and R. Stiefelhagen. Pointing gesture recognition based on 3D tracking of face, hands and head orientation, In *Proceedings of the Fifth International Conference on Multimodal Interfaces*, pp. 140-146, 2003.

[9] E. J. Ong and R. Bowden. A boosted classifier tree for hand shape detection, In *Proceedings of the 6th IEEE Conference on Automatic Face and Gesture Recognition*, pp. 889-894, 2004

[10] Y. Raja, S. J. Mckenna and S. G. Gong. Colour model delection and adaptation in dynamic dcenes, *5th European Conference on Computer Vision*, pp. 460-474, 1998.

[11] L. Sigal, S. Sclaroff and V. Athitsos. Skin Color-based video segmentation under time-varying illumination, *IEEE Trans on Pattern Analysis and Machine Intelligence*, 26(7), pp. 862-877, 2004.

[12] Y. Wu and T. S. Huang. Color tracking by transductive learning, In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 133-138, 2000.

[13] Y. Wu, Q. Liu and T. S. Huang. An adaptive self-organizing color segmentation algorithm with application to robust real-time human hand localization, *In Proceedings of Asian Conference on Computer Vision*, pp. 1106-1111, 2000.

[14] X. M. Yin and M. Xie. Hand image segmentation using color and RCE neural network. *Journal of Robotics and Autonomous Systems*, 34(4), pp. 235-250, 2001.

[15] Z. Zhang, Y. Wu, Y. Shan and S. Shafer. Visual panel: Virtual mouse keyboard and 3D controller with an ordinary piece of paper, In *Proceedings of the Workshop on Perceptive User Interface*, pp. 1-8, 2001.

[16] X. Zhu , J. Yang, and A. Waibel . Segmentation hands of arbitrary color, In *Proceedings of the IEEE Conference on Automatic Face and Gesture Recognition*, pp. 90-95, 2000.

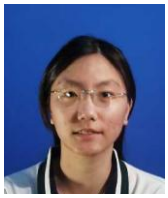
Miaolong Yuan received his BS degree in Mathematics from Hangzhou Normal College in 1992 and his PhD degree in Mechanical Engineering from Huazhong University of Science and Technology (HUST), China in 1997. During 1997-1999, he worked in HUST as a lecturer. During 1999-2000, he worked as a senior software engineer in Asahi Hi-tech Co, Ltd, Japan. From 2000 to 2005, he was a research fellow in Singapore-MIT Alliance, Singapore. From 2005 to 2008, he was a research fellow in Department of Mechanical Engineering, at the National University of Singapore. From 2008, he has been working at A*STAR Institute for Infocomm Research as senior research fellow. His research interests include computer vision, human computer interaction, augmented reality and its applications in manufacturing.

Farzam Farbiz (M'02) was born in Tehran-Iran 1970. He received his Bachelor degree in 1992, Master degree in 1994 and his Ph.D. degree in Electrical Engineering, Electronics in 1999 from Amirkabir University of Technology, Tehran-Iran. His Ph.D. thesis was on computational intelligence filters for image enhancement. He received the first rank of the national young researcher award in 1999 as the best Iranian young researcher due to his research work.

Dr. Farbiz worked as an assistant professor at Zanjan University, Zanjan-Iran from 1999-2001 teaching undergraduate and post graduate courses. Then he joined Electrical and Computer Engineering Department of National University of Singapore, NUS, as research fellow working in areas of augmented and virtual reality. From 2006 he has been working at A*STAR Institute for Infocomm Research as senior research fellow and principle investigator on multimodal game engine and mixed reality system for home application projects. He is also collaborating with A*STAR Data Storage Institute on developing laser holographic display systems. He has published more than 60 papers in international conferences and journals and has served as technical reviewer and program committee in many international journals and conferences.

Dr. Farbiz is the member of IEEE and ACM SIGGRAPH and currently he is one of the technical committee members of SIGGRAPH Singapore Chapter, SSC

Corey Mason Manders currently works as a Senior Research Fellow at the Institute for Infocomm Research, A*STAR, in Singapore. He received his Ph.D. in Computer Engineering from the University of Toronto, in Canada in 2006. Previously, he complete a Master's of Applied Science (M.A.Sc) from the University of Toronto in Computer Engineering in 2002. He holds two undergraduate degrees, one from the University of Toronto in Computer Science (2001), and one from York University in Canada in Fine Arts (B.F.A 1991).



Tang Ka Yin is a Research Officer in A*Star's Institute for Infocomm Research. She received a BEng in Electrical Engineering degree and MEngSc in Signal Processing from the University of New South Wales, Australia in 2004 and 2005. Her research interests include computer graphics, gesture tracking and virtual reality.