

Real-Time Fingertip Detection from Cluttered Background for Vision-based HCI*

Lianwen Jin¹, Caihong Chen², Lixin Zhen³, Jiancheng Huang⁴

(1,2 South China University of Technology, Guangzhou 510640, China;

3,4 Motorola China Research Center, Shanghai 200002, China)

Abstract: Vision-based HCI (Human Computer Interaction) is an important technology to make machine more intelligent. Fingertip tracking and detection is the key for many vision-based HCI systems. In this paper, we propose a new fingertip detection approach from cluttered background, and present a new simple but effective foreground segmentation algorithm based on the characteristics of intensity change over the entire image, and a new background updating method based on foreground connection detection. The fingertip position is finally detected based on template matching after the finger foreground image has been successfully segmented. Based on the accurate detection of fingertip, a novel virtual character recognition system is designed. With this fingertip tracking based system, human can input characters to computer using just the movement of his fingertip, without any additional devices such as keyboard, digital pen, so that it provides a novel wireless character inputting modality for HCI application.

Key words: Moving Fingertip Detection; Background Modeling; Finger-writing; Virtual Character Recognition; Human Computer Interaction

1. Introduction

Vision-based HCI (Human Computer Interaction) is

one of the recently popular research topics in the field of HCI^[1,2,3]. Besides that there are some technologies such as sound, speech recognition, pen-based computing, context-aware devices and conventional WIMP (Windows, Icons, Menu and Pointer) approaches which allow for a much richer, multi-modal interaction between human and machine. One kind of vision-based HCI system is vision-based fingertip tracking. There have been a lot of researches and applications in this field, such as gesture recognition^[4,5], sign language recognition^[6,7,8], finger mouse^[9,10], Augmented Desk Interface Systems^[11,12], window system control (such as WWW navigator, menu control)^[13,14], finger paint system^[10] and so on. The basic techniques include color segmentation^[4,6], infrared segmentation^[5,12], blob-models^[15], contours modeling^[16], correlation or template matching^[11,12] and wavelets^[17]. However, there are various kinds of restrictions for most of these systems mentioned above. For example, colored or special devices such as gloves are needed^[4,8], expensive hardware requirements (such as 2-D/3D camera or infrared-camera)^[5,12,14], no real-time application^[16], and restrictive background conditions (e.g. simple background^[9,11,13], or background should be brighter than hand^[10]).

In this paper, we propose a new real-time fingertip detection approach from clutter background without any restrictions for background or device. Based on the accurate detection of fingertip, a novel virtual character recognition system (VCRS) is built. With the

*Acknowledgments: This work is supported by Motorola Human Interface Lab Research Foundation (No.D84110) and NSFC (No.60275005).

Lianwen Jin, professor; main research fields: character recognition, image processing, computer vision and human-machine interface; E-mail: eelwjin@scut.edu.cn.

Caihong Chen, postgraduate; main research fields: image processing and pattern recognition.

Lixin Zhen, researcher of Motorola China Research Center; main research fields: pattern recognition, image processing and human-machine interface.

Jiancheng Huang, researcher of Motorola China Research Center; main research fields: pattern recognition, signal processing and human-machine interface.

VCRS, human can input characters to computer using just the movement of his fingertip, without any additional devices such as keyboard and digital pen. The main contribution of this paper is that a new background model based on pixel intensity analysis is proposed, a foreground segmentation method based on the overall image intensity analysis is given. Furthermore, the application of our finger-writing based on the system is a novel HCI modality, which is a kind of wireless character inputting method.

This paper is arranged as follows. In section 2, a brief review of several different background models is given, and then we present our background model. A comparison of our model and another famous background model named W4 is carried out. In section 3, we describe an improved foreground segmentation method which can deal with illumination change in indoor situation. After segmentation of the foreground objects, fingertip detection based on template matching is described in the 4th section. In section 5, the basic idea of our fingertip-based virtual character inputting HCI system is described. Conclusions are given in section 6.

2. Background Modeling

Background modeling is one of the important computer vision problems. Although many methods have been proposed to address some of the solutions for this problem^[15,18-22], it may be safe to say that universal model for foreground detection from arbitrary background has not been found. Pixel intensity is one of the most commonly used features in background modeling. The pixel intensity is often modeled with some statistical distribution, in which parameters are gotten after a process of training and recursively updating. In the Pfinder system^[15], Wren has modeled the pixel intensity of a complexly stationary background as a Gaussian distribution $N(m, s^2)$, and the image noise over time is modeled by a zero mean Gaussian distribution $N(0, s^2)$. This

Gaussian distribution model is the baseline model for many background modeling methods^[18-21]. Although single Gaussian based model can adapt to slow changes in the scene by real-time updating the model, it does not adapt to outdoor scenes in which background is changed frequently. Many other methods have been proposed to solve this problem in outdoor scenes^[18-22]. In reference [18], a mixture of three normal Gaussian distributions is used to model the pixel value for traffic, shadow, and vehicle in outdoor environment. In reference [19], a non-parametric background model is used by estimating the density function of observing pixel intensity distribution based on a sample of intensity value for each pixel. Grimson and Stauffer^[21] have characterized the intensity value of each pixel with a mixture of K Gaussian distributions (typically, K is three to five). Besides modeled background with Gaussian distribution, W4^[22] has used temporal deviation to model the background. In W4, each pixel is represented by three values: the minimum intensity values, the maximum intensity values and the maximum intensity difference between consecutive frames observed during training period. Most of the above methods are proposed to solve the situation in outdoor sites, not designed for indoor environment. Moreover, models of mixture Gaussian are computational intensive and not fit for real-time application without the support of special hardware.

In our fingertip tracking based system, ordinary USB camera is used in indoor sites for video capture. Although the background of indoor scene is not as complex as outdoor, there are still many problems that should be taken into consideration, such as shadow, flash light, etc. In order to solve the problems of indoor background modeling, we propose a simple but effective background model to segment human hand from cluttered background. Our background model $M_i(x)$ is given by equation (1) :

$$m_t(x) = \frac{1}{N} \sum_{i=1}^N |I_i(x, t)| \quad (1)$$

$$d_t^2(x) = \frac{1}{N} \sum_{i=1}^N (I_i(x, t) - I_i(x, t-1))^2$$

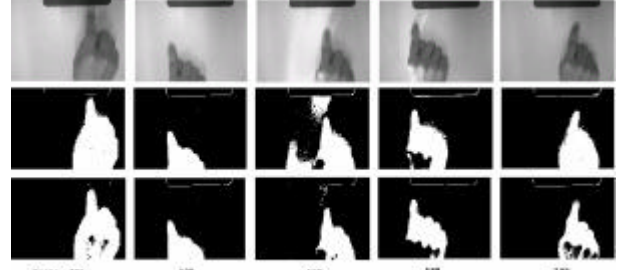
where N is the number of initial frames (typically 20-40) without foreground.

After the initial background model is obtained, we propose the following background model $M_t(x)$ for each pixel x at frame t :

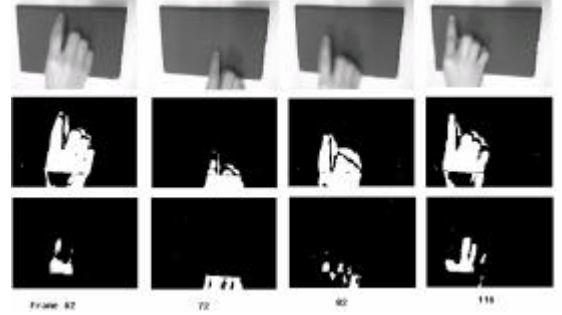
$$M_t(x) = \begin{bmatrix} m_t(x) \\ d_{\text{median}}^t \\ d_{\text{max}}^t \end{bmatrix} = \begin{bmatrix} m_t(x) \\ \text{median}_z \{ |I^t(z) - I^1(z)| \} \\ \text{max}_z \{ |I^t(z) - I^1(z)| \} \end{bmatrix} \quad (2)$$

where z is pixel index over the current frame. Here, $m_t(x)$ is the intensity value of background model of pixel x at time t , $m_0(x) = m_t(x)$. The d_{median}^t and d_{max}^t are the median and maximal value of intensity change over the whole image respectively, and they are updated for each frame. Although $M_t(x)$ is just a simple model without high computation cost, it works very well in indoor scene and can adapt to intensity changes caused by the white-balance and indoor light change.

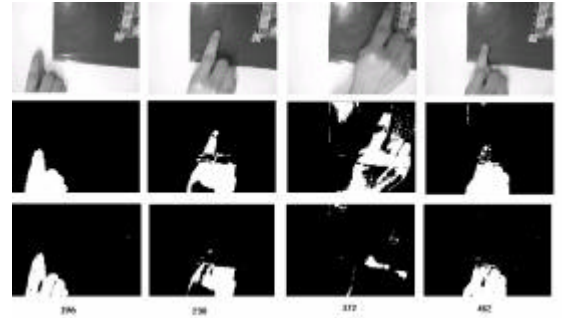
Fig. 1 shows some experimental results of our model and W4 model (a famous background model^[22]) in different situations. From Fig. 1(a) to Fig. 1(c), the 1st row displays origin pictures, the 2nd is foreground extracting results using our background model, and the 3rd is results of W4. In Fig. 1(a), frame 469 and 588 are the situations when a flash light is shining. We can see from Fig. 1(a) that though W4 can deal with wider spectrum of intensity changes, it can not detect the foreground with fewer changes very well (see frame 588). In Fig. 1(b) and Fig. 1(c), it can be seen that our model works better than W4 when the background consists of high intensity contrast objects, or the intensity of some background objects is similar with foreground. The parameters of two models (the minimum and maximum intensity values) used in W4 can tolerate large derivation of pixel intensity, but relatively small intensity change can not be detected by W4.



(a) Illumination Change with Flash Light at Frame 469, 588



(b) Background with High Intensity Contrasts



(c) Background with Similar Intensity of Foreground

Fig. 1 Comparison of Foreground Segmentation Results between Our Method and W4

3. Foreground Segmentation Based on Overall Intensity Analysis

The basic idea of foreground segmentation is: Each pixel intensity is considered as a Gaussian distribution $N(m, s^2)$, different pixels have different median value m , but share the same standard deviation s . So if the deviation from background model m is greater than a certain degree, it will be classified as foreground pixel. From the analysis of video imagery, it can be found that intensity value changes greatly when the foreground hand passes (see

“c” point of Fig. 2), while other pixels (see “a”, “b” points of Fig. 2) change little (just caused by illumination change and noise).

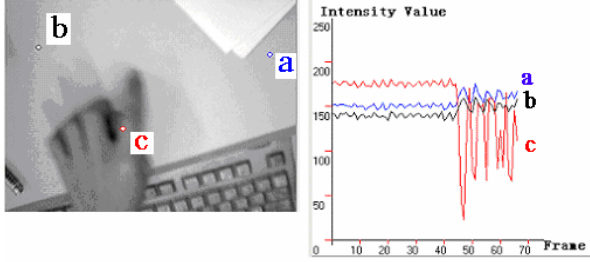


Fig. 2 Pixels Intensity Analysis of Video Imagery

In Fig. 2, pixels *a* and *b* are background points, of which the intensity value changes are only caused by illumination change and white-balance or noise, and pixel *c* is one of the foreground points, of which the intensity values change greatly when foreground object passes.

Based on the observation and the basic idea described above, the most commonly used foreground segmentation method is a kind of threshold method described in equation (3). But it does not work very well for our system. From experiments, we find that the method based on equation (3) will cause lots of noise (for example, see the experimental results in Fig. 5, the 3rd line.)

$$B(x) = \begin{cases} 0 & \text{background, } I^t(x) - m_l(x) < k \cdot d_l(x) \\ 1 & \text{foreground, otherwise} \end{cases} \quad (3)$$

where $I^t(x)$ is the intensity value of pixel x at t time, $m_l(x)$ is the initial median intensity value of pixel x , and $d_l(x)$ is the initial intensity value derivation.

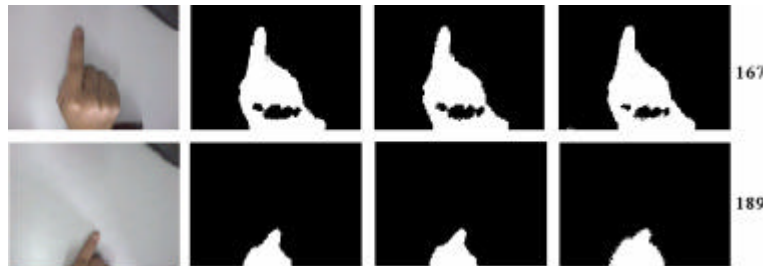
In order to adapt to indoor light change, we improve a segmentation method based on overall pixels intensity change. The threshold value is designed dynamically

according to global intensity derivation d_{median}^t and d_{max}^t . Those pixels, when illumination change is great, will be verified and detected as foreground. Our segmentation method can eliminate white-balance problem, which is given by equation (4):

$$B(x) = \begin{cases} 0 & \text{background if } I^t(x) - m_l(x) < k_{\text{median}} \cdot d_{\text{median}}^t \\ & \vee I^t(x) - m_l(x) < T \vee I^t(x) - m_l(x) < k_{\text{max}} \cdot d_{\text{max}}^t \\ 1 & \text{foreground, otherwise} \end{cases} \quad (4)$$

In equation (4), d_{median}^t is the median value of $I^t(x) - m_l(x)$, d_{max}^t is the maximal value of $I^t(x) - m_l(x)$, k_{median} (typically 1.5-1.8) and k_{max} (typically 0.2-0.3) are constant weight of d_{median}^t and d_{max}^t . T is a constant (typically 25~28). All the condition $I^t(x) - m_l(x) < k_{\text{median}} \times d_{\text{median}}^t$, $I^t(x) - m_l(x) < k_{\text{max}} \times d_{\text{max}}^t$ and $I^t(x) - m_l(x) < T$ contribute to the segmentation of foreground. Detailed experimental results are shown in Fig. 3, Fig. 4 and Fig. 5 respectively.

In Fig. 3, frame 167 is a normal situation. On one hand, when the foreground is only a small part as shown in 189, 196 frame, the d_{median}^t value is relatively small, so the use of strategy $I^t(x) - m_l(x) < k_{\text{max}} \times d_{\text{max}}^t$ can eliminate some shadow disturbance. On the other hand, when the foreground is a big part as shown in 352 and 412 frame, the strategy $I^t(x) - m_l(x) < k_{\text{median}} \times d_{\text{median}}^t$ works better than $I^t(x) - m_l(x) < k_{\text{max}} \times d_{\text{max}}^t$ meanwhile sometime d_{max}^t may be very great because of noise, in this case the strategy $I^t(x) - m_l(x) < k_{\text{max}} \times d_{\text{max}}^t$ will not work, so it is necessary to introduce the strategy of $I^t(x) - m_l(x) < k_{\text{median}} \times d_{\text{median}}^t$.



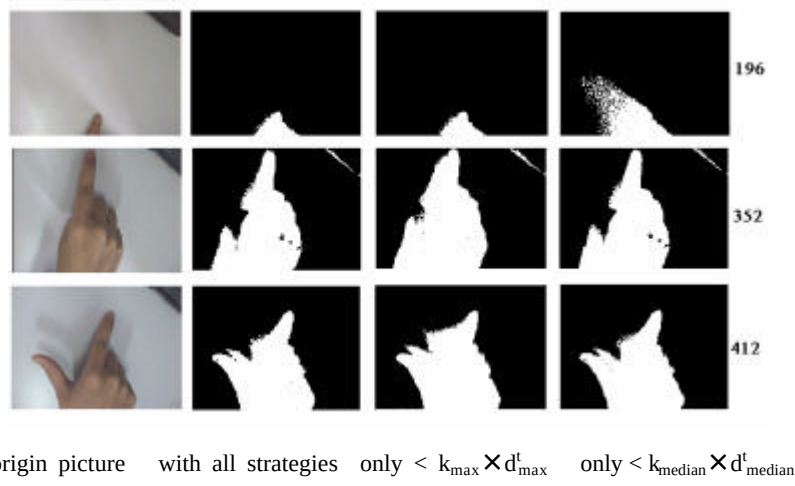
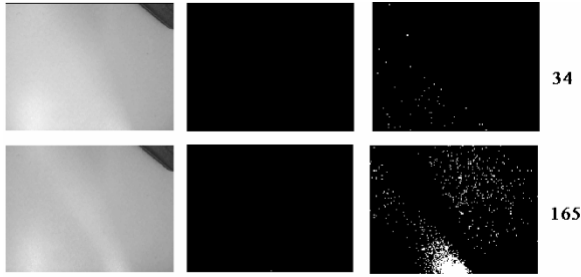


Fig. 3 Comparison of the Judge Strategies in Equation (4)

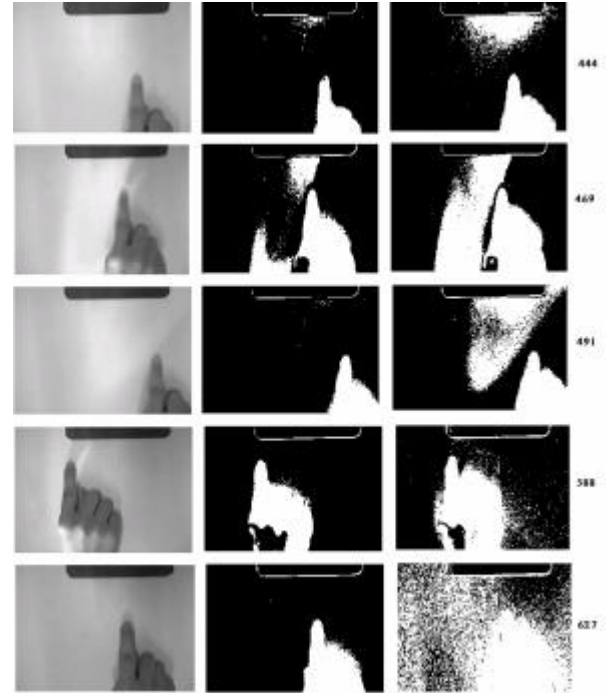
The strategy $\hat{I}(x) - m_l(x) < T$ is introduced to deal with noise when there is no foreground, in this case, $d_{\max}^t$ and d_{median}^t are very small, as shown in Fig. 4.



Original Picture With All Conditions Lack Condition $\hat{I}(x) - m_l(x) < T$

Fig. 4 Use of Strategy $\hat{I}(x) - m_l(x) < T$ in Formula (4)

Fig.5 compares the foreground segmentation results based on equations (3) and (4). Frames 444~588 are the situations when flashlight is shining. From frame 627 it can be seen that the white-balance phenomena is introduced after the flash light passes. It can be seen that our segmentation method based on global derivation is more powerful to eliminate illumination change and white-balance phenomena.



Original Picture Foreground of Our Method Result Based on Eq.(3)

Fig. 5 Comparison of Our Foreground Segmentation Method with Traditional Method

From Fig. 3, Fig. 4 and Fig. 5, it can be seen that our segmentation approach works better than traditional method.

4. Fingertip Detection

The fingertip detection of our system consists of

two stages: rough fingertip region based on histogram analysis and temple-matching stage to detect the accurate position of fingertip.

4.1 Rough Fingertip Region Detection

The rough region is defined by histogram analysis in the detected foreground hand image, as shown in Fig. 6. The rough detection process is accomplished by the following three steps:

Step 1: Calculate x , y coordinate histograms of current image respectively.

Step 2: Detect a region of “beginning $X \sim \text{end } X$ ” where the corresponding histogram value changes significantly in x coordinate. Detect a region of “beginning $Y \sim \text{beginning } Y + \text{picHeight}$ ” in y coordinate.

Step 3: The rough fingertip region is typically defined by a rectangle of (beginning $X - 10$, end X , beginning $Y + 35$, end $Y + 35$), as shown in Fig. 6.

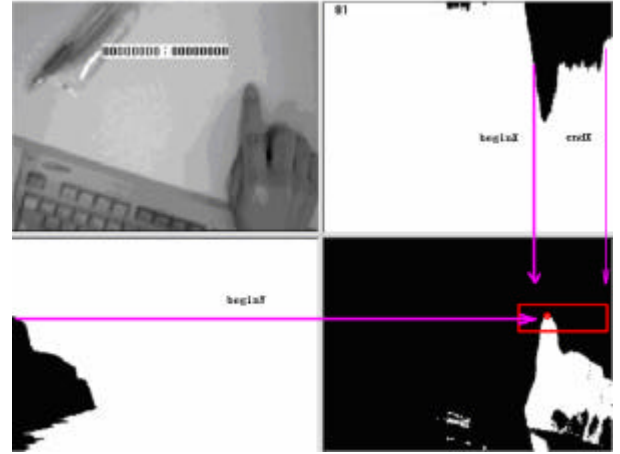


Fig. 6 Detection of Rough Fingertip Region

4.2 Fingertip Detection by Template Matching

The template matching method based on equation (5) is used to find the accurate fingertip position.

$$(i_t^*, j_t^*) = \underset{i, j}{\operatorname{argmin}} \left\{ \sum_{m=0}^{wWidth} \sum_{n=0}^{wHeight} \operatorname{abs} \left(p(i + m + 5, j + n + 3) - w(m, n) \right) \right\} \quad (5)$$

where (i_t^*, j_t^*) is the fingertip position detected at frame t , $p(i, j)$ is the inputting picture, $w(m, n)$ is the fingertip template, $wWidth$ and $wHeight$ are the width and the height of template image.

From experiments we have found this method is good enough to get the accurate fingertip from the rough region given by Fig. 6. Fingertip detection results are illuminated in Fig. 7.

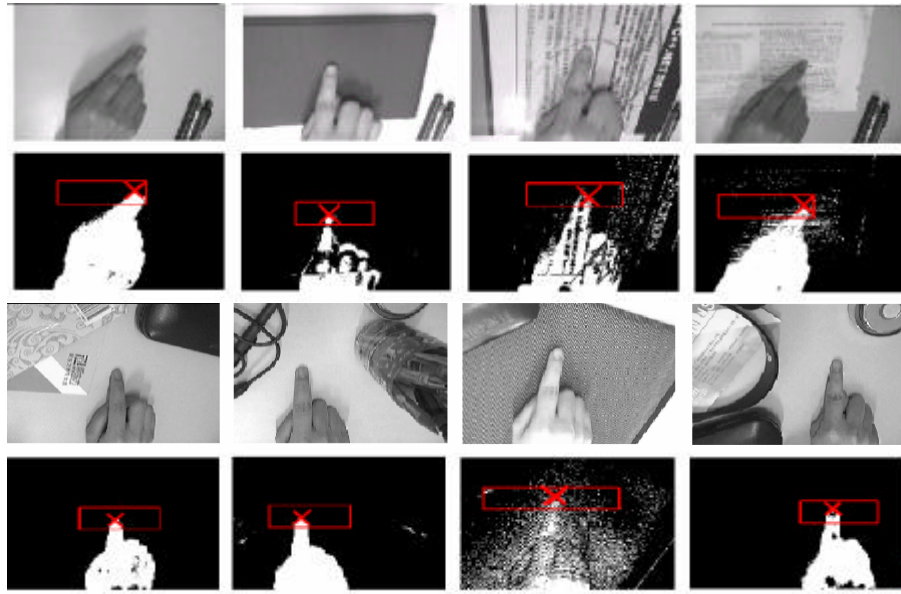


Fig. 7 Some Fingertip Detection Results with Different Cluttered Background

From Fig. 7, it can be seen that our fingertip detection approach is effective and promising. The proposed approach can also deal with some situations when the detection foreground picture is a blur (see the 3rd column of Fig. 7). Experiments on large real world video data suggest that accurate rate for fingertip position detection can be 95% or higher.

5. Fingertip Based Virtual Character Inputting HCI system

The diagram of our fingertip based virtual character inputting HCI system is shown in Fig. 8 The basic idea of the proposed system is as follows: the

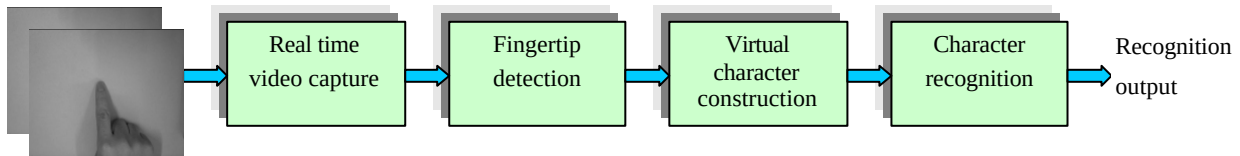


Fig. 8 Diagram of the Proposed Fingertip Based Virtual Character Inputting HCI System

The virtual character reconstruction is implemented by linking all consecutive detected fingertip positions together when human finished “writing” a character using fingertip.

Gabor features are extracted for the reconstructed virtual characters^[23], and then a LDA (Linear discriminate Analysis) based classifier^[24] is employed to give the final fingertip-written recognition result. The demo system is implemented by Microsoft VC++6.0, Windows 2000 OS and DirectX8, on a PC of Pentium IV 1.8G CPU and 256M memory. We use normal consume-grade low resolution USB1.0 digital camera to capture the movement of fingertip. The size of each frame image is 320×240. The speed of the system is about 20fps, and it is suitable for real time application. Experiments show that the correct rate for fingertip-written English letters and digitals recognition is more than 98.5%. Fig. 9 illuminates our finger-written HCI system.

movement of human fingertip is captured by camera, and the trajectories of the fingertip is detected and reconstructed as a kind of character (since there is no ink when we use fingertip “write” with the certain movement of the fingertip, we call the reconstructed character as virtual character). A classifier is then employed to recognize the virtual character “written” by fingertip. By this way, human can input character to computer without any additional devices such as pen, digital pen and keyboard.

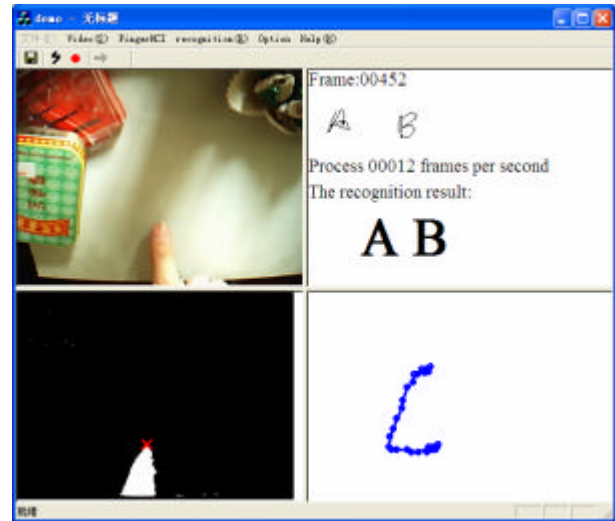


Fig. 9 Overview of Our Fingertip-Written Character Recognition HCI System

Compared with conventional character inputting methods based on keyboard, digital pen, or touch screen, the merit of our vision based character inputting system is that it is a novel inputting method requiring small hardware devices (since the size of camera is much smaller than keyboard or touch screen), so that it is especially suitable for embedded

application, such as applied to smart mobile phone, PDA, etc.

6. Conclusions

In this paper, we have proposed a simple but effective background model and background updating model for vision-based fingertip HCI system. A foreground segmentation method based on overall intensity change is proposed to segment the moving finger object. Real-time experiments show that in indoor environment, the system can effectively model the background and can accurately find the fingertip position. Based on the fingertip detection algorithm, a novel character inputting system based on fingertip-written character recognition is built.

References

- [1] J. K. Aggarwal, Q. Cai, *Human Motion Analysis: A Review*, Computer Vision and Image Understanding, Vol.73, No.3, March, 1999: pp. 428-440.
- [2] V. Pavlovic, R. Sharma, Thomas S. Huang, *Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1997, 19(7): pp. 677-695.
- [3] Liang Wang, Weiming Hu, Tieniu Tan, *Recent Developments in Human Motion Analysis*, Pattern Recognition 36, 2003: pp. 585-601.
- [4] Lien, C. Huang, C, *Model-based Articulated Hand Motion Tracking For Gesture Recognition*, Image and Vision Computing, February 1998, vol.16, no.2: pp. 121-134.
- [5] Kenji Oka, Yoichi Sato, Hideki kOike, *Real-time Tracking and Gesture Recognition*, IEEE Trans. Computer Graphics and Application, December, 2002: pp. 64-71.
- [6] Kazuyuki Imagawa, Shan Lu, Seiji Igi, *Color-Based Hands Tracking System for Sign Language Recognition*, Proc. International Conf. on Automatic Face and Gesture Recognition, 1998: pp. 462-467.
- [7] T. Starner, J. Weaver, et al., *Real Time American Sign Language Recognition Using Desk and Wearable Computer-based Video*, IEEE Trans. PAMI, 1998.
- [8] Chunku Wang, Wen Gao, Shiguang Shan, *An Approach Based on Phonemes to Large Vocabulary Chinese Sign Language Recognition*, Proceedings of the 5th IEEE International Conference on Automatic Face and Gesture Recognition, 2002.
- [9] Quek, F., T. Mysliwiec, M. Zhao, *Finger Mouse: A Freehand Pointing Interface*, in Proc. of International Workshop on Automatic Face- and Gesture-Recognition, Zurich, Switzerland, 1995.
- [10] Christian von Hardenberg, François B  rad, *Bare-Hand Human-Computer Interaction*, Proceeding of the ACM Workshop on Perceptive User Interfaces, Orlando, Florida, USA, Nov., 2001: pp. 15-16.
- [11] Crowley, J., B  rad, F., Coutaz, J, *Finger Tacking as an Input Device for Augmented Reality, Automatic Face and Gesture Recognition*, Z  rich, 1995: pp. 195-200.
- [12] Kenji Oka, Yoichi Sato, Hideki Koike, *Real-time Tracking of Multiple Fingertips and Gesture Recognition for Augmented Desk Interface Systems*, Proc. IEEE Intl Conf. Automatic Face and Gesture Recognition (FG 2002), May, 2002: pp. 429-434.
- [13] R. Kjeldsen, J. Kender, *Visual Hand Gesture Recognition for Window System Control*, Proc. Int'l Workshop Automatic Face- and Gesture-Recognition, Zurich, Switzerland, 1995: pp. 184-188.
- [14] M. S. Lee, D. Weinshall, et al., *A Computer Vision System for On-screen Item Selection by Finger Pointing*, Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, 1997: pp. 1027-1033.
- [15] C. R. Wren, A. Azarbayejani, T. Darrell, A. P. Pentland, *Pfinder: Real-time Tracking of Human Body*, IEEE Trans. Pattern Anal. Machine Intell., Vol.19, July, 1997: pp. 780-785.
- [16] Michael Isard, Andrew Blake, *Condensation—Conditional Density Propagation for Visual Tracking*, International Journal of Computer Vision, 1998, 29(1): pp. 5-28.
- [17] Triesch, J. Malsburg, C, *Robust Classification of Hand Posture against Complex Background*, International Conference on Automatic Face and Gesture Recognition, Killington, 1996.
- [18] N. Friedman, S. Russell, *Image Segmentation in Video Sequences: A Probabilistic Approach*, Uncertainty in Artificial Intelligence, 1997.
- [19] A. Elgammal, R. Duraiswami, D. Harwood, L. S. Davis, *Background and Foreground Modeling Using Nonparametric Kernel Density Estimation for Visual Surveillance*, Proc. IEEE, Vol.90, No.7, July, 2002.
- [20] A. Elammal, D. Harwood, L. S. Davis, *Nonparametric Background Model for Background Subtraction*, in Proc. 6th Eur. Conf. Computer Vision, Vol. 2, 2000: pp.751-767.
- [21] W. E. L. Grimson, C. Stauffer, R. Romano, *Using Adaptive Tracking to Classify and Monitor Activities in a Site*, in IEEE Conf. Computer Vision and Pattern Recognition, 1998: pp. 22-29.
- [22] I. Haritaoglu, D. Harwood, L. S. Davis, *W4:Real-Time Surveillance of People and Their Activities*, IEEE Trans. On Pattern Analysis and Machine Intelligence, Vol.22, No.8, Aug., 2000.
- [23] Daugman, J. G., *Two Dimensional Spectral Analysis of Cortical Receptive Field Profiles*, Vision Research, Vol.20, 1980: pp. 847-856.
- [24] R. O. Duda, P. H. Hard, *Pattern Classification and Scene Analysis*, New York, John Wiley & Sons Pub. Co., 1973.

(Editors: Mei Huang, Ivan, Zidan)