EyeScreen: A Vision-Based Desktop Interaction System[1]

Yihua Xu, Jingjun Lv, Shanqing Li and Yunde Jia

School of Computer Science, Beijing Institute of Technology
Beijing 100081, PR China
{yihuaxu, kennylanse, shanqingli, jiayunde}@bit.edu.cn

Abstract. EyeScreen provides a natural HCI interface with vision-based hand tracking and gesture recognition techniques. Multi-view video images captured from two cameras facing a computer screen are used to track and recognize finger and hand motions. Finger tracking is achieved by skin color detection and particle filtering, and is greatly enhanced by the proposed screen background subtraction method that removes the screen images in advance. Finger click on the screen can also be detected from multi-view information. Gesture recognition based on binocular vision is presented to improve the recognition rate. The experimental results show that EyeScreen is able to perform natural and robust interaction in desktop environment.

Keywords: Vision-based interaction system, Human computer interaction, Finger tracking, Gesture recognition

1 Introduction

Keyboards, mice, and joysticks are most commonly used in traditional HCI applications. However, these devices are not sufficiently convenient for natural interaction, because they are difficult to supply 3D and high degree of freedom inputs [1]. Vision-based gesture interaction technique realizes non-contact interaction by analyzing the captured image sequences of human hand as well as extracting and recognizing the static and dynamic hand features. This technique offers a natural and direct interaction approach, which can be used in many applications such as virtual reality and intelligent interaction.

Two approaches are widely used in vision-based gesture interaction: 3D model-based approach and appearance-based approach. The former can recover the 3D hand model and represent gesture independent of viewpoint. However, the construction of models needs large amounts of computation, which makes 3D model-based systems incapable of providing real-time interactions. The appearance-based approach uses more simple features like color, histogram, and contour, etc., or projects images to

[1] This work was partially supported by The High-Tech Program and Natural Science Foundation (60675021) of China.

specific space and extracts hand features from projected data. Though a large number of samples are needed to obtain rational features or projection space in training procedure, the recognition process is of computational efficiency that makes real-time interaction possible. Many vision-based interaction system have been developed in recent years [2~5]. Zhang et al. [2] presented a Visual Panel system using an arbitrary quadrangle panel as user interface. The system provides flexible and robust interaction through tracking the panel and a tip pointer. Corso et al. [3] built a 4D Touchpad platform based on the Visual Interface Cues (VICs) framework. The system supplements 3D physical interaction with an additional temporal dimension, in which predefined sequences of VICs are used to trigger interaction events. Malik and Laszlo [4] presented a Visual Touchpad system that allows fluid two-handed interactions with computers. The system acquires the 3D position of a user's fingertip with two cameras, and simulates the mouse clicks by detecting contact of user's fingertips with a panel surface. However, these systems use extra objects as user interface while feedbacks are displayed on the screen, which distracts users' attention from the interactions and limits the naturalness and directness of interactions.

EyeScreen is a vision-based interaction system employing two cameras facing a screen from different views, which is able to detect finger clicking action on screen and recognize hand gestures. The system provides direct interactions because the command input and feedback display are accomplished through a same screen. We achieve robust hand and finger tracking through skin color detection and particle filtering algorithm [6]. To improve robustness, a screen background subtraction method is proposed that removes background screen images in advance of tracking. PCA and Discriminant-EM (D-EM) [7] algorithms are applied to recognize gestures in each of the two views. To overcome the self-occlusion problem, we propose a gesture level analysis method that integrates the recognition results from two views. We have designed a number of applications to verify the usability of our system, such as finger input, gesture paging and vision-based gaming. The results show that EyeScreen is robust enough to provide natural interactions and can be readily applied to many real-world scenarios.

2 System Configuration

An example setup for EyeScreen is shown in Fig. 1(a). Two cameras are mounted in front of a screen to capture multi-view images covering the full screen. The common view of the two cameras and the screen forms an interaction space, Fig. 1(b). In this space, user can issue commands to computer by the hand motion and gestures and obtain instant feedback right on the screen.

The system algorithm consists of three steps: system calibration, finger tracking and gesture recognition. System calibration algorithm computes the two planar homographies between image planes and the screen plane, as well as the homography between two image planes. Finger tracking algorithm realizes robust hand and finger tracking and detects finger's click actions on the screen. Gesture recognition algorithm recognizes certain hand gestures in the interaction space using two-view information. All these algorithms are implemented in real time, and the outputs are used to drive the application programs.

Two homography matrices, which define projective transformations between image planes and screen plane, are calculated in system calibration. Given an arbitrary point (x_i, y_i) in an image plane, its corresponding point (x_s, y_s) in the screen plane can be calculated by

$$[x_i, y_i, 1] \cong \boldsymbol{H}[x_s, y_s, 1] \quad (1)$$

where ≅ means equal up to a scale factor, **H** is a 3x3 homography matrix. A chessboard image is rendered on the screen providing pairs of corresponding points to compute the **H** matrix.

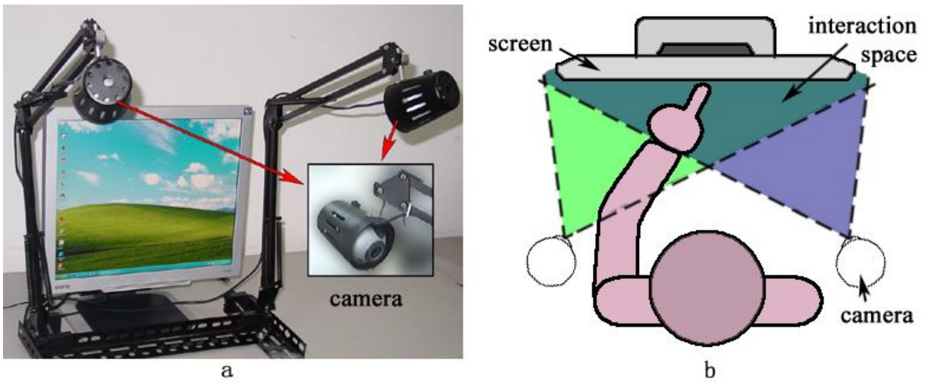

Fig. 1. (a) System configuration (b) Interaction using EyeScreen

3 Finger Tracking

In this paper, finger tracking is achieved through the combination of skin color detection and contour-based particle filtering method [6]. In order to avoid the distraction of skin-like images displayed on the screen, a screen background subtraction approach is proposed to remove the screen images in advance of hand tracking. Click detection is realized using projective transformation.

3.1 Skin Color Detection

Skin color detection is used to segment hand regions and locate the fingertip's position approximately. A Gaussian color model is built in HSV color space to represent the distribution of skin color, and a probability image will be generated in which each pixel value indicates the probability of being skin color. Then hand region can be extracted by thresholding method. Fingertip's position can be calculated approximately using the hand model proposed by Takao [8].

3.2 Screen Background Subtraction

Skin-color-like regions displayed on screen interferes skin-color-based hand segmentation severely. Screen background subtraction can efficiently alleviate the problem by utilizing projective transformation between screen plane and image planes represented by homographies obtained in system calibration. Given an arbitrary scene point P_S on the screen with its image pixels P_L and P_R in the left and right image planes, the projective transformations from the screen plane to the left and right image planes can be given with homography matrices $\mathbf{H}_{LS}$ and $\mathbf{H}_{RS}$ respectively by

$$P_L = \boldsymbol{H}_{LS} P_S \qquad (2)$$

$$P_R = \boldsymbol{H}_{RS} P_S \qquad (3)$$

From Equation (2) and (3), we get

$$P_L = \boldsymbol{H}_{LS} \boldsymbol{H}_{RS}^{-1} P_R \qquad (4)$$

For every pixel in right image, its corresponding point P_L' can be calculated by $P_L ' = \boldsymbol{H}_{LS} \boldsymbol{H}_{RS}^{-1} P_R$. As shown in Fig.2, P_L' coincides with P_L only if the scene point is on the screen. So we transform all points in right image to left image plane by Equation (4), and calculate the difference image with left image, where the region corresponding to screen display has much lower pixel value than other regions. So, screen background can be easily subtracted by a simple thresholding method, which can efficiently improve the robustness of skin-color-based hand region segmentation, as shown in Fig. 3.

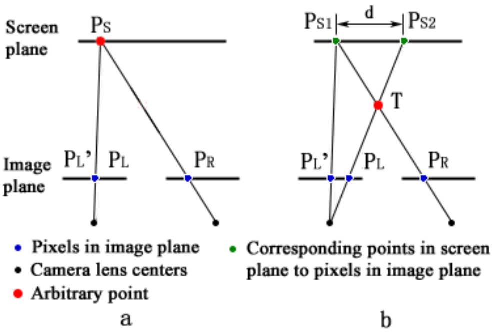

Fig. 2. (a) When P_S is in the screen plane, the point P_L', which is the corresponding point of P_R in left image, coincides with P_L; (b) When T is not in the screen plane, P_L' does not coincide with P_L.

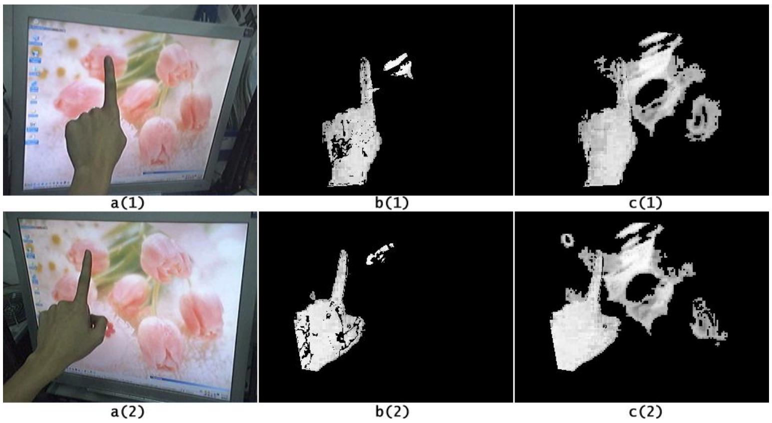

Fig. 3. Row 1: Left images; Row 2: Right images; (a) Source images (b) Results of skin color detection after applying screen background subtraction (c) Results of skin color detection on source images

3.3 Particle Filter Tracking

In this paper, we integrate the contour and color cues to achieve robust finger tracking in the framework of particle filter, which offers a statistical method for dynamic state estimation. In the standard particle filtering algorithm, the knowledge about tracked object at time t is predicted only from its state at time t-1, and then contour-based observation step is used to calculate the weights of new samples. It proceeds iteratively to obtain more accurate tracking results. In our system, samples with various finger shapes are generated by applying rigid transformations to a finger shape model [9], which is represented by a B-spline curve composed of 7 control points. The model of uniform motion is adopted to predict the movement of the finger. To further improve the prediction efficiency, importance sampling is introduced to provide a way of direct searching when auxiliary knowledge is available. The coarse finger position derived from skin color detection is used to generate importance function, which describes the regions with high probability of finger occurrence. The accurate prediction, which combines the previous finger state and auxiliary knowledge extracted at current time, makes the tracking performance very robust.

3.4 Click Detection

To detect clicking actions of a fingertip, we follow the approach described in [4] by projective transformation. As shown in Fig. 2 (b), the corresponding points in screen plane of fingertip T are P_{S1} and P_{S2} respectively, and d is the distance between P_{S1} and P_{S2}. Since the value of d will reduce when T approaches to screen, clicking action can be detected by thresholding method.

3.5 Experimental Results

We use our method and the method combining only skin color detection and particle filtering to track finger under complex screen background. As the results given in Fig. 4, our method can efficiently alleviate the interference caused by complex screen background and improve the robustness of tracking.

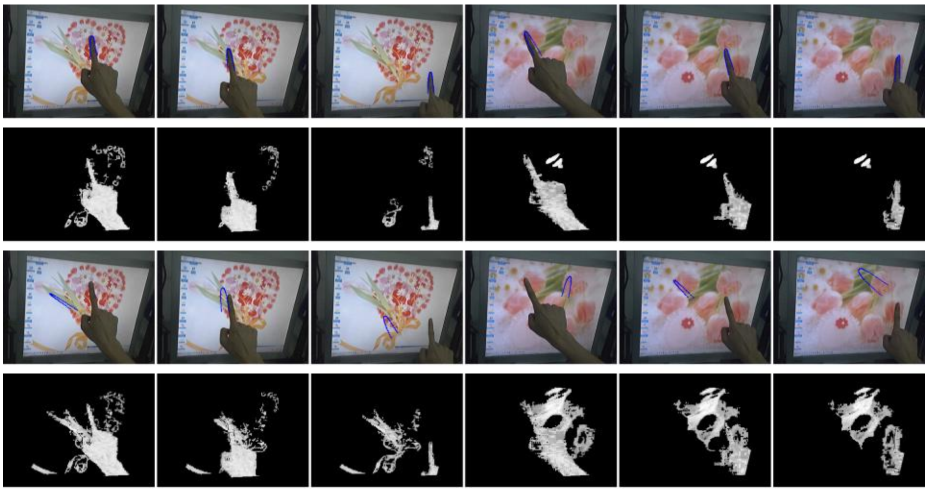

Fig. 4. Comparison of finger tracking methods, blue curves are the finger tracking results: Row 1: Tracking results using our method; Row 2: Extracted skin color region after applying screen background subtraction; Row 3: Tracking results directly using skin color detection and particle filtering; Row 4: Results of skin color detection on source images

4 Gesture Recognition

Monocular gestures can be robustly recognized by firstly using PCA to obtain principle components and then applying D-EM algorithm to classify gestures [7]. However, the problem of self-occlusion still makes gestures ambiguous. A gesture level analysis method is proposed in this paper, which combines pattern information captured by left and right cameras to recognize gestures. By applying PCA and D-EM algorithms to images from each camera, we obtain the recognition results in two different views. Then gesture level analysis is used to select the proper gesture as the final result. Experiments show that our method can efficiently overcome the problem of self-occlusion and improve the accurate rate and speed of recognition.

4.1 Gesture Level

The number of fingers is a significant appearance feature for hand gestures, which is used to assign different levels to gestures in our system. We define the fist gesture as level zero, and gesture with one finger as level one, etc., as shown in Fig. 5. When self-occlusion occurs, gesture recognition is subjected to the following constraint: compared with the ground truths, gestures might be recognized as lower levels, but never the higher levels. The constraint can be used to make a correct decision when recognition results derived from two views are different.

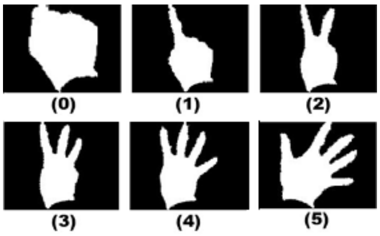

Fig. 5. Levels of gestures

4.2 Modifying Recognition Results

As the difference between visual angles of two cameras in our system is about 40°~ 60°, the gesture without self-occlusion can be captured in at least one of the two views. Based on the constraint mentioned in 4.1, the gesture level might decrease when self-occlusion occurs. We recognize a gesture using PCA and D-EM algorithms in two views separately, and then select the class of recognized gesture with higher level than the other as the final output result, which is described by

$$C_{output} = \{C_i \mid L_i = \max\{L_l, L_r\}, i \in \{l, r\}\} \qquad (5)$$

where C_{output} is the output gesture class and C_i is the recognized gesture class in certain view, while L_l and L_r are the gesture levels in left and right view respectively.

4.3 Experimental Results

We calculate the recognition rate for 6 classes of gestures which are shown in Fig. 5. 1800 pairs of images are collected in our data set including gestures with self-occlusion for one camera. We use 240 pairs of images as training data, and the others as testing data. The recognition rate is 98.53% when using our method, and 84.13% for the left camera, 87.53% for the right camera, respectively.

The linked-image method [9], which also meets the problem of self-occlusion, uses only one recognizer while the method presented in this paper uses two. However, the dimension of a linked image increases compared with source image, which causes the increase of computation. We compare the two methods still using training data and testing data mentioned above. Recognition accuracy and time cost of recognizing all samples are calculated. As shown in Table 1, our method improves the recognition rate and speed efficiently.

Table 1. Comparison of two recognition methods

Resolution	Method	Accuracy	Time (second)
64x48	Gesture Level	98.20%	16.79
	Linked Image	96.82%	21.77
160x120	Gesture Level	98.53%	99.23
	Linked Image	97.05%	141.09
320x240	Gesture Level	98.85%	624.25
	Linked Image	97.33%	810.81

5 Application

In this section, we discuss three applications based on our system: finger input, gesture paging and vision-based gaming.

5.1 Finger Input

Relying on the accurate finger tracking and click detection of our system, user can input text on screen with finger. Two virtual input devices are provided here: screen handwriting pad and screen keyboard. When using screen handwriting pad, the trajectory of fingertip motion on the screen will be recorded and then recognized, as shown in Fig. 6 (a); when user's fingertip touches any key of the screen keyboard, the click event of corresponding button will be generated to input the corresponding character, as shown in Fig. 6 (b).

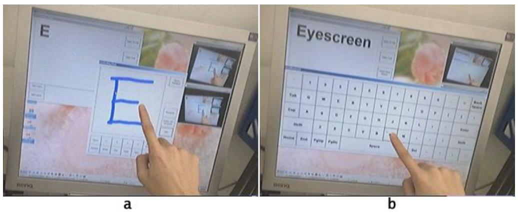

Fig. 6. (a) "E" is written and recognized; (b) "n" key of the keyboard is pressed

5.2 Gesture Paging

We designed a gesture driven document reader, which allows user to control document display by simple gestures. As shown in Fig. 7, an opened document is shown as a book on the screen, and user can turn the document pages naturally as doing that for a real book.

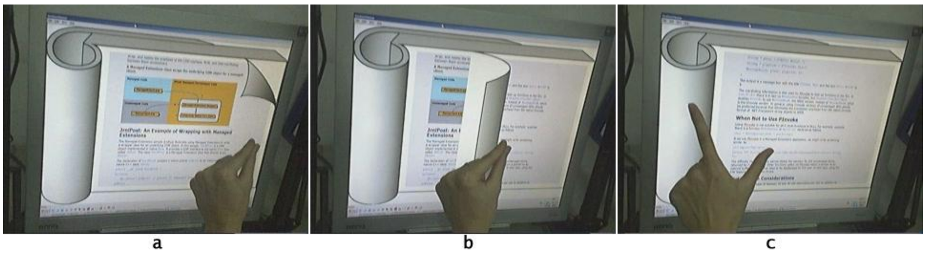

Fig. 7. (a) Ready to turn a page; (b) Paging forward; (c) Exit paging

5.3 Vision-based Gaming

Applying click detection and gesture recognition techniques, we designed a vision-based interaction game called "Visual Pilot", which allows user to manipulate a virtual fighter naturally by hand. Finger pressing on buttons can be detected to control the game, and the actions of fighter are manipulated by hand motions and simple gestures. Some snapshots of playing the game are given in Fig. 8.

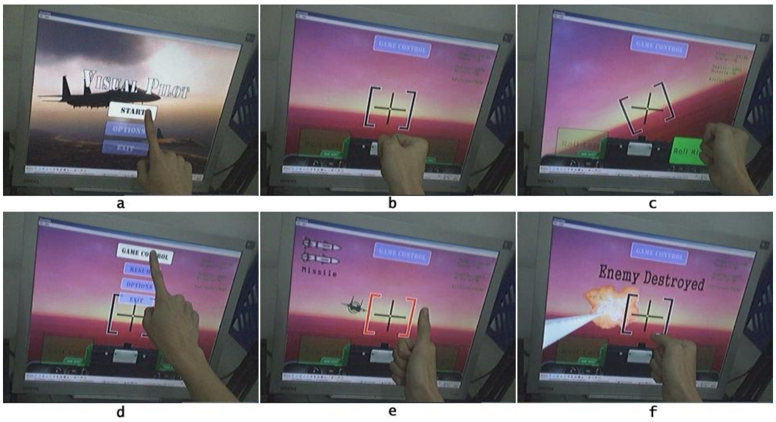

Fig. 8. (a) "Start" button is pressed; (b) Manipulating the fighter to fly horizontally; (c) Manipulating the fighter to roll clockwise; (d) "Game Control" button is pressed; (e) Ready to launch a missile; (f) Manipulate to launch a missile.

6 Conclusion

EyeScreen is a robust vision-based desktop interaction system, which utilizes the desktop interaction space efficiently and achieves natural and direct HCI. In this paper, we combine color detection, screen background subtraction and particle filtering to achieve robust finger tracking. A novel gesture recognition method is also presented, which improves the recognition rate and speed. The system has a wide application foreground in intelligent interaction and digital entertainment.

Currently, EyeScreen provides robust tracking and gesture recognition for single hand. In our future work, multi-object tracking and recognition techniques can be adopted to improve the usability, and utilization of 3D information can enhance the robustness of our system.

References

1. Wu Ying, Huang T.S.: Hand Modeling, Analysis, and Recognition. In IEEE Signal Processing Magazine, 18(3). (2001) 51-60
2. Zhang Zhengyou, Wu Ying, Shan Ying and Shafer S.: Visual Panel: Virtual Mouse, Keyboard and 3D Controller with an Ordinary Piece of Paper. In Proc. ACM workshop on Perceptive User Interfaces (PUI). (2001)
3. Corso J. J., Burschka D. and Hager G. D.: The 4D Touchpad: Unencumbered HCI with VICs. In Proc. IEEE Workshop on Computer Vision and Pattern Recognition. for Human Computer Interaction (CVPRHCI). (2003)
4. Malik S. and Laszlo J.: Visual Touchpad: A Two-handed Gesture Input Device. In Proc. ACM Int. Conf. on Multimodal Interfaces (ICMI), USA. (2004) 289-296
5. Oka K., Sato Y. and Koike H.: Real-time Tracking of Multiple Fingertips and Gesture Recognition for Augmented Desk Interface Systems. In Proc. IEEE Int. Conf. on Automatic Face and Gesture Recognition. (2002) 429-434
6. Isard M.: Visual Motion Analysis by Probabilistic Propagation of Conditional Density. D.Phil. Thesis. Oxford University. (1998)
7. Wu Ying. and Huang T.S.: View-independent Recognition of Hand Postures. In Proc IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR). (2000) 2:88～94
8. Takao N., Shi J., and Baker S.: Telegraffiti: A Camera-Projector Based Remote Sketching System with Hand-Based User Interface and Automatic Session Summarization. In Int. Journal of Computer Vision. (2003) 115-133
9. Xu Yihua, Li Shanqing, Jia Yunde: EyeScreen: A Gesture-Based Interaction System. In Proc. IEEE Int. Conf. on Signal and Image Processing. (2006).