A Real-Time Robust Eye Tracking System for Autostereoscopic Displays using Stereo Cameras

Chan-Hung Su[1,2] Yong-Sheng Chen[3,*] Yi-Ping Hung[1,2] Chu-Song Chen[1] Jiun-Hung Chen[1]

[1]Institute of Information Science, Academia Sinica, Taipei, Taiwan
[2]Dept. of Computer Science and Information Engineering, National Taiwan University, Taiwan
[3]Dept. of Computer Science and Information Engineering, National Chiao Tung University, Taiwan
[*]Email: ysc@csie.nctu.edu.tw

Abstract – **Autostereoscopic display systems can provide users a natural 3-D visualization environment by projecting stereo video onto the user's eyes. Eye position localization is a central module in this kind of display system when users are allowed to move freely. This paper presents robust 3-D eye tracking techniques that can provide accurate eye positions in real time. Technical batteries comprise: (1) robust face detection based on eigenspace method; (2) real-time face tracking; and (3) eye detection in the obtained face region. According to our implementation on a PC with a Pentium IV 1.2 GHz CPU, the frame rate of the eye tracking process can achieve 25 Hz.**

I. INTRODUCTION

Human computer interface plays an important role in a virtual reality system. How to provide an effective, robust, friendly, and immersive communication channel has continued being an intensive research issue. Among the channels, stereoscopic display systems can provide users 3-D visualization environments in a virtual reality system. Conventionally, 3-D visualization requires wearing stereo glasses or head-mounted displays, which may lead to inconvenience and uncomfortableness. The current trend is towards the autostereoscopic display system [1], [2] which renders the stereo video with respect to the user's view point and projects the left and right channels of the stereo video to the left and right eyes of the user respectively. Thus, there is no need to wear any special device on the head.

To provide great enjoyment of stereo visualization without posture limitation, the user can move around freely in front of the autostereoscopic display to watch the stereo video from various points of views. Therefore, the autostereoscopic display system usually contains an eye tracking component for tracking both the left and right eye positions of the user. To facilitate user tracking, commercial products usually adopt active sensing methods which require that the user to wear some special sensors, such as infrared sensors or reflectors, ultrasonic wave receivers, and electromagnetic wave sensors. The major drawback of this kind of tracking methods is that these sensors cannot be mounted on the eyes and thus the obtained positions are not exactly the needed eye positions. Therefore, it is preferable to utilize video cameras for tracking the eye positions in a passive manner.

In the interactive graphics system developed by Azarbayejani et al. [3], a video camera is used for control by observing the user, exacting the image features, and tracking the motion of user's head. Instead of the feature-based method as in [3], another kind of tracking methods utilizes skin-color information for tracking the user's facial region [4], [5], [6]. By using an active camera head, Shirai [7] was able to track the 3-D human behaviour from the estimated optical flow and depth information. Recently, Stauffer and Grimson [8] developed an adaptive background subtraction technique for visual tracking in a monitoring system. In the previous study of this work, we have developed real-time eye tracking techniques for autostereoscopic display systems [9] by using a single camera. Without the requirement of wearing sensors or marks, the 2-D eye positions of the user, in addition to the size and rotation of the facial region, can be tracked in the acquired image sequence.

In this work, we develop a 3-D eye tracking system by using stereo cameras. First, the eigenspace-based method [10] is used for face detection. To improve the robustness and accuracy of the located face position, facial components, such as eyes, nose, and lips, are used for verification. Next, fast template matching technique [11], [12] is adopted to track in real time the motion of the user's face in the following images. Within the extracted facial region, the left and right eyes can be located according to the geometric relation between the face and eyes. After repeating the above process for both cameras, 3-D eye positions can be calculated by using triangulation method.

II. THE EYE TRACKING SYSTEM

This section presents the proposed video-based eye tracking techniques. Some design issues considering the eye tracking for autostereoscopic display systems are first addressed. The whole eye tracking process is sketchily depicted in a flowchart. Afterwards, details of the technical corpus are described.

A. Design Issues

In an autostereoscopic display system, it is not necessary to keep track of the user's eyes all the time. Instead, the system has to know the eye positions only when the user is *watching* the autostereoscopic display. That is, correct tracking is required only for the frontal face when the cameras are mounted in front of the user. Considering that the user can look at the display at different positions in the 3-D space while keeping his/her face towards the display, there are totally four motion parameters to be estimated in the image sequences, which includes the translations in the X- and Y-axes, the scaling, and the rotation around the normal axis of the image.

In order to meet the real-time requirement, we adopt a fast template matching algorithm, the winner-update algorithm [11], [12], for rapid visual tracking. Moreover, a multilevel conjugate direction search technique (MCDS) is used to search and track the template image block, considering its scale and rotation motion parameters in addition to its X/Y translation parameters. To increase the robustness and accuracy of the tracking results, we track the face region first in consecutive image frames. Once the face region is located, the left and right eye positions can then be estimated in the upper-left and upper-right parts of the face region, respectively. Excluding the mouth part, the face region comprises salient features of the eyes and nose and hence increases the stability and accuracy. More details of the design issues can be found in [9].

B. Flowchart of the Eye Tracking System

Fig. 1 illustrates the procedure of the presented eye tracking system. In the beginning, we continue acquiring images and detecting the face region in each image until the face detection succeeds. Once found, the face region is recorded as a long-term face template, called a face representative. Next, the eye positions can be located in the face region obtained from face detection or tracking. The face region under tracking is saved as a short-term face template, which is referred in the following as the face template for simplicity. In the tracking cycle, a new image is acquired and the winner-update algorithm [11], [12] is adopted to match the face template in a search range of the newly acquired image. Then, the candidate face position having the minimum matching error is verified and the face position is refined by using the face representative and the MCDS technique. When the verification succeeds, the eye positions are located and the face template is updated by using the newly obtained face image. This iteration is repeated until the verification fails. To rapidly recover the tracking process, we keep the face template without modification and match the face template in a few consecutive image frames.

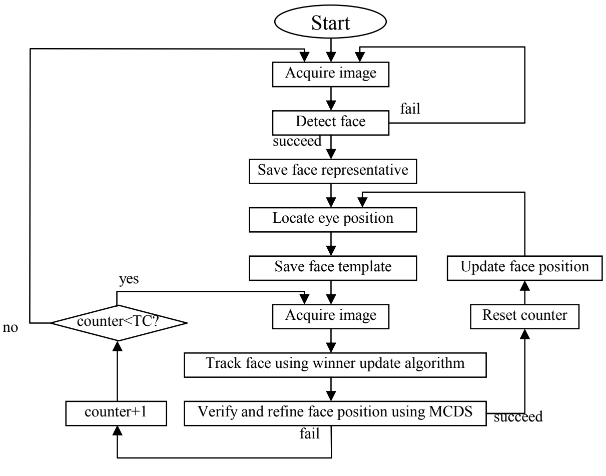


Fig. 1: Flowchart of the eye tracking system.

C. Eye Tracking Techniques

Automatic face detection has to be performed when a user appears in the image or when the tracking process has to be restarted for the recovery from tracking failure. In this work, we adopt a multi-resolution framework for detecting the face in the image. The face detection module is divided into two stages: the candidate selection stage and the candidate verification stage. In the candidate selection stage, we first select some face candidates which are more likely to be face regions in a lower resolution using the eigenface method [10]. These face candidates are then verified in a higher resolution using the eigencomponents, such as the eigeneye and the eigennose, in the candidate verification stage.

A general eigenspace-based method for face detection works as follows. First, a set of training face images are collected. These training images form a matrix with each row representing the pixel values of a training face image. The eigenspace of a smaller dimension can then be calculated by using the PCA. For the image block (with the same size as that of the training face images) at each position in the acquired image, its distance to the eigenspace is calculated to determine whether this image block contains a face image. In this work, if the distance from the image block under examination to the eigenspace is small

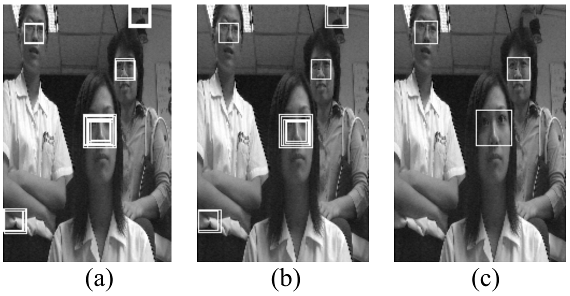


Fig 2: Illustration of candidate merging. (a) All face candidates without merging; (b) face candidate after merging; and (c) the final result verified as faces.

enough, we can conclude that this image block may contain a face. This image block is thus labeled as a face candidate and subjected to further verification.

One of the major difficulties of the eigenface method is the lighting variation problem. If the image under examination is acquired in a lighting condition that is much different from the lighting condition in which the training images of the eigenspace are acquired, the distance from the image block under examination to the eigenspace will be very large, even if it actually contains a face. In this work, we adopt a lighting compensation method, consisting of plane fitting and histogram equalization techniques, to reduce the affection of lighting variation. For each image block, before being subjected to the PCA, we can find a linear plane that best fits the image block. The image block is subtracted by the linear plane and the residual image block is normalized by using the histogram equalization. This lighting compensation processing is performed on each image block which is then subjected to the PCA, during both the training stage and the detecting stage.

As mentioned above, face candidates which are more likely to be face regions are selected in the candidate selection stage. After that stage, there may be many face candidates with various scales and different locations. Before the verification stage, face candidates of the same scale will be merged if they are close enough, and the centroid will be calculated as the new position of the new block. Each block after merging will then be subjected to the verification and assigned an evidence score. For overlapping blocks, all the image blocks except the one with the highest score will be discarded.

As shown in Fig. 2 (b), there may be more than one face candidate after the merging operation. We perform a component-wise verification on each merged face candidate. Fig. 3 shows the facial components that we use in this work. For each components, we collect some training images and calculate the eigenspace with the standard PCA method mentioned above. The obtained eigenvectors are called eigencomponents. We use these eigencomponents to detect the facial features respectively in the corresponding area of the face candidate. A verification score SCR_x is given to the face candidate x according to the following equation:

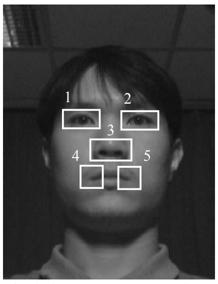


Fig. 3: Illustration of the facial components. The number is the index of the component.

$$SCR_x = \sum_{i=1}^{5} (W_i \times C_i)$$

where W_i is the weighting of the i^{th} component:

$$W_i = \begin{cases} 12 & i < 3 \\ 10 & i = 3 \\ 5 & i > 3 \end{cases}$$

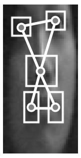
Let b_i be the detected block of the i^{th} component and $\mathrm{MDIS}(b_i)$ be the Mahalanobis distance of b_i. The confidence ratio *Ci* of the block b_i is defined according to the error ratio E_i:

$$C_i = \begin{cases} 0 & \mathit{if}\ E_j > 1 \\ 1 & \mathit{if}\ E_j < 0.1 \\ 1 + \dfrac{E_i - 0.1}{0.1 - 1.25} & \mathit{otherwise} \end{cases}$$

$$E_i = \frac{MDIS(b_i)}{Threshold^{i}_{MDIS}}.$$

Due to the changing expressions, the corners of the mouth are relatively less robust than the other facial features. They are beneficial only to help the verification of those candidates when one eye or the nose is failed to be found.

For face candidate block *x*, if the verification score SCR_x is smaller than the threshold $Threshold_{SCR}$, we claim that *x* is not a human face. In this work, the value $Threshold_{SCR}$ is defined as follows. For those blocks with *SCR* larger than $Threshold_{SCR}$, there are two more criteria to satisfy: first, the arrangement of facial components must be symmetric enough; and second, the intensity of eyes must be dark enough. Let $D_{i,j}$ be the distance from the center of the i^{th} component to the center of the j^{th} component in the candidate block *x*. The candidate block *x* is symmetric enough if the following equations hold:



$$\begin{cases} \dfrac{\min(D_{1,3},D_{2,3})}{\max(D_{1,3},D_{2,3})} \ge 0.8 \\ \dfrac{\min(D_{4,3},D_{5,3})}{\max(D_{4,3},D_{5,3})} \ge 0.8 \end{cases}$$

$$Threshold_{SCR} = \left(\sum_{i=1}^{3} W_i \right) \times 0.6$$

Let AI_upper_x be the average intensity of the upper part of face candidate block x (namely, the average intensity of upper face) and $AI_{x,i}$ be the average intensity of the i^{th} component, the eye image regions are dark enough if the following equations hold:

$$\begin{cases} AI\_\text{upper}_x \ge AI_{x,1} \\ AI\_\text{upper}_x \ge AI_{x,2} \end{cases} .$$

Candidate blocks meet the above-mentioned two criteria will be recognized as human face in the long run. Currently, however, the eye tracking system requires only one face. Thus it only tracks the face with the maximum verification score.

For each image frame, the face image block that is successfully tracked is stored as the face template. This face template will be used for template matching within a search range in the consecutive image. In order to achieve high efficiency while retaining high accuracy, we adopt the winner-update algorithm [11], [12] which guarantee that the global minimum of the template matching can be found efficiently.

For the candidate face position yielding the minimum matching error, the goal of face verification is to determine whether the image block at this position actually contains a face. Decision making can base on the facts that the image block at this candidate face position should "looks" similar both to the face template and to the detected face representatives. Consequently, we make use of the face representative, which is a long-term face template, to perform the verification while refining the face position. To deal with the scaling and rotation of the face image block, we compute various combinations of scaling and rotation for the face representative in advance. MCDS technique is then used to rapidly search for the combination such that the corresponding face representative is similar to the candidate image block. In addition to the verification, MCDS technique is also used to refine the face position along the gradient directions in the X and Y dimensions, one at a time. Further detailed tracking methodology is presented in [9].

Once the face position is determined, we can locate the left and right eye positions within the face region. Instead of the feature-based method [13], we adopted a convolution-based method with an 8x8 mask. The position with the smallest convolution result, which means the darkest region, is considered as the eye position.

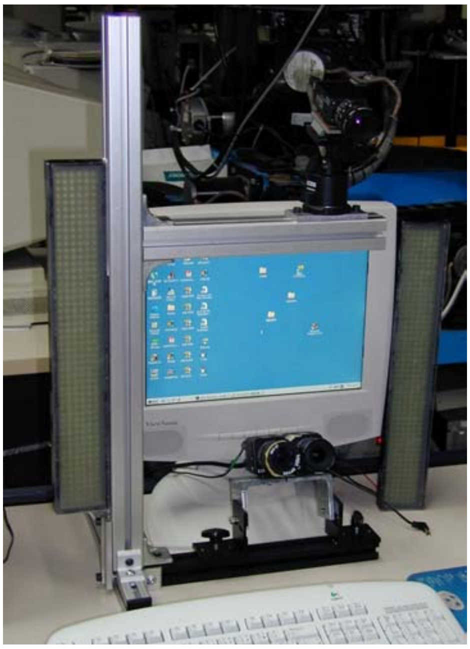


Fig. 4: The stereo eye tracking system.

Since the stereo cameras in the presented eye tracking system are calibrated, all the intrinsic and extrinsic camera parameters are known beforehand. The 3-D eye positions can thus be calculated with triangulation method when the 2-D eye positions are both successfully obtained in the stereo images acquired from the stereo cameras.

III. EXPERIMENTS

The proposed eye tracking system, depicted in Fig. 4, is implemented on a PC with Pentium® IV 1.2 Ghz CPU running Microsoft Windows® 2000. The video frame rate of the image acquisition equipment is 30 Hz, which will limit the maximum frame rate and the minimum latency time of the eye tracking system. In our implementation, the tracking process and the video acquisition proceed concurrently. The tracking process for each frame can be finished in less than 1/30 second. Consequently, the overall frame rate of our eye tracking system is 30 Hz, i.e., the video frame rate, and the latency time ranges from 1/30 to 2/30 second when only one single camera is engaged. When both the stereo cameras are used, the frame rate of the stereo eye tracking system still can achieve 25 frames per second.

Fig. 5 illustrates some face detection results. As shown in Figs. 6 and 7, the proposed tracking system can successfully detect and track user's faces with various expressions or under different lighting conditions.

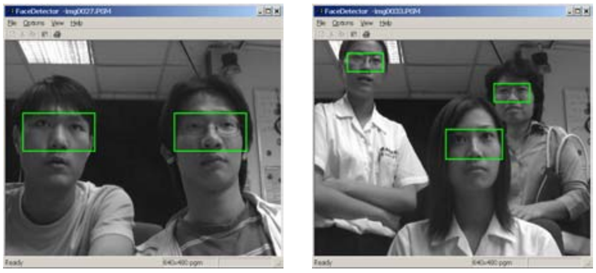


Fig. 5: Examples of face detection results.

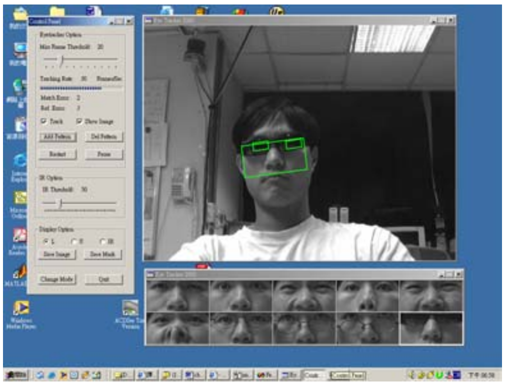


Fig. 6: Results of detecting and tracking faces with various expressions.

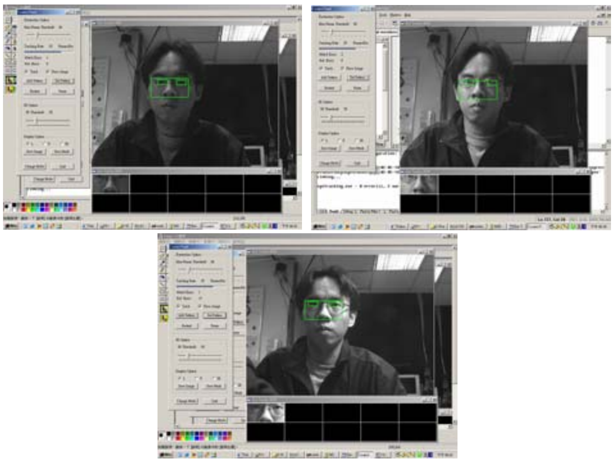


Fig. 7: Result of detecting and tracking faces under different lighting conditions.

IV. CONCLUSIONS

We have presented real-time eye tracking techniques for autostereoscopic display systems. These techniques can accurately, robustly, and efficiently track the user's face and eye positions in 3-D space. According to our implementation, the frame rate of the eye tracking process can achieve 25 Hz.

IV. ACKNOWLEDGMENTS

This work is partially supported by Opto-Electronics & Systems Laboratories, Industrial Technology Research Institute and by the National Science Council under grant NSC-90-2213-E-001-013, Taiwan.

V. REFERENCES

[1] G. Woodgate, D. Ezra, J. Harrold, N. Holliman, G. Jones, and R. Moseley, "Autostereoscopic 3D display systems with observer tracking," *Signal Processing: Image Communication*, vol. 14, 1998, pp. 131-145.

[2] S. Pastoor, J. Liu, and S. Renault, "An experimental multimedia system allowing 3-D visualization and eye-controlled interaction without user-worn devices," *IEEE Transactions on Multimedia* , vol. 1, no. 1, 1999, pp. 41-52.

[3] A. Azarbayejani, T. Starner, B. Horowitz, and A. Pentland, "Visually controlled graphics," *IEEE Transactions on PAMI*, vol. 15, no. 6, 1993, pp. 602-605.

[4] P. Fieguth and D. Terzopoulos, "Color-based tracking of heads and other mobile objects at video frame rates," in *Proceedings of CVPR*, Puerto Rico, 1997, pp. 21-27.

[5] N. Herodotou, K. N. Plataniotis, and A. N. Venetsanopoulos, "Automatic location and tracking of the facial region in color video sequences," *Signal Processing: Image Communication*, vol. 14, no. 10, 1999, pp. 359-388.

[6] T. Darrell, G. Gordon, M. Harville, and J. Woodfill, "Integrated person tracking using stereo, color, and pattern detection," *IJCV*, vol. 37, no. 2, 2000, pp. 175-185.

[7] Y. Shirai, "Estimation of 3-D pose and shape from a monocular image sequence and realtime human tracking," in *Proceedings of the International Conference on Recent Advances in 3-D Digital Imaging and Modeling*, 1997, pp. 130-139.

[8] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Transactions on PAMI*, vol. 22, no. 8, 2000, pp. 747-757.

[9] Y.-S. Chen, C.-H. Su, J.-H. Chen, C.-S. Chen, Y.-P. Hung, and C.-S. Fuh, "Video-based eye tracking for autostereoscopic displays," *Optical Engineering*, vol. 40, no. 12, 2001, pp. 2726-2734.

[10] B. Moghaddam and A. Pentland, "Probabilistic visual learning for object representation," *IEEE Transactions on PAMI*, 1997.

[11] Y.-S. Chen, Y.-P. Hung, and C.-S. Fuh, "A fast block matching algorithm based on the winner-update strategy," in *Proceedings of the ACCV*, vol. 2, 2000, pp. 977-982.

[12] Y.-S. Chen, Y.-P. Hung, and C.-S. Fuh, "Fast block matching algorithm based on the winner-update strategy," *IEEE Transactions on IP*, vol. 10, no. 8, 2001, pp. 1212-1222.

[13] K.-M. Lam and H. Yan, "Locating and extracting the eye in human face images," *Pattern Recognition*, vol. 29, no. 5, 1996, pp. 771-779.