In Proceedings of the Fifth Conference on Artificial Intelligence and Applications, Taipei, Taiwan, Nov. 2000, pages 188–193

Video-based Realtime Eye Tracking Technique for Autostereoscopic Displays

Yong-Sheng Chen†‡, Chan-Hung Su†‡, Jiun-Hung Chen‡,
Chu-Song Chen†, Yi-Ping Hung†‡,* and Chiou-Shann Fuh‡
†Institute of Information Science, Academia Sinica, Taipei, Taiwan
‡Department of Computer Science and Information Engineering,
National Taiwan University, Taipei, Taiwan

Abstract

An autostereoscopic display system can provide great enjoyment of stereo visualization without the uncomfortable and inconvenient drawbacks of wearing stereo glasses or HMD. In order to render the stereo video with respect to the user's view point and to accurately project stereo video onto the eyes of the user, who is allowed to move around freely, the left and right eye positions of the user have to be obtained when the user is watching the autostereoscopic display. In this paper, we present a realtime eye tracking technique which can track the eye positions of the user in the video acquired with a camera. The user does not have to wear any sensor or mark and there is no restriction on the background. We employ a fast template matching technique to track the four motion parameters (X and Y translation, scaling, and rotation) of the user's face in each image. The left and right eyes can then be located in the obtained face region. According to our implementation on a PC with Pentium III 500, the frame rate of the eye tracking process can achieve 30 Hz.

1 Introduction

Virtual reality systems become more and more attractive in the applications of education, exhibition, training, and entertainment. One of the major components of virtual reality system is the stereoscopic display for providing the user with stereo visual environment. Conventional stereoscopic displays require users to wear on their heads stereo glasses or Head-Mounted Displays (HMDs), which will make the users feel less comfortable and thus can not immersively enjoy the virtual environment. Recently, researchers began to develop autostereoscopic display systems which can provide great enjoyment of stereo visualization without the requirement of wearing any special device [3]. As shown in Figure 1, an user can move around freely in front of the autostereoscopic display to watch the stereo video from different points of view. The eye tracking component of the autostereoscopic display system is used for tracking the left and right eye positions of the user. The autostereoscopic display system can then render the stereo video with respect to the view point of the user and project the left and right channels of the stereo video onto the two corresponding eyes of the user. In this kind of display system, the eye tracking component is a very important module for the accurately and fluently rendering and projecting the stereo video.

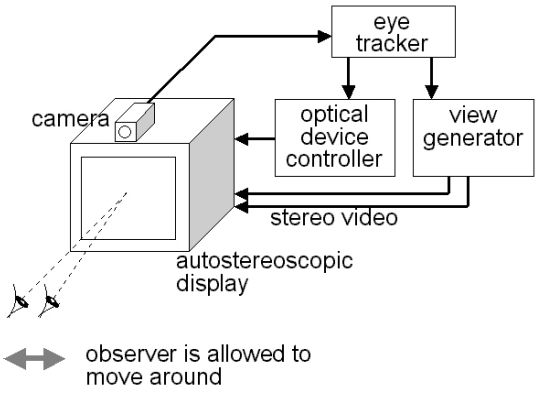

Figure 1. A look around system.

One kind of the tracking methods require that the user wears some special sensors, such as infrared sensors or reflectors, ultrasonic wave receivers, and electromagnetic wave sensors. However, this kind of active sensing methods may cause uncomfortableness

*All the correspondences should be sent to Yi-Ping Hung, Institute of Information Science 20, Academia Sinica, Nankang, Taipei 115, Taiwan. Tel.: +886-2-27883799 ext. 1718. Fax: +886-2-27824814. E-mail: hung@iis.sinica.edu.tw.

and inconvenience. As a result, video-based tracking techniques are often adopted in an autostereoscopic display system to track the user in a passive way [4]. For example, Pastoor et al. built an experimental multimedia system [3] which can be controlled by eye movement. Their system use a camera to track the head position, eye position, and gaze direction in the acquired video. Morimoto et al. also proposed a pupil detection and tracking technique [2]. Two sets of LEDs were mounted on-axis and off-axis with the camera lens. Images with and without "red eyes" can be acquired by alternatively lighting up these two sets of LEDs. Therefore, pupil position can be obtained by using simple image difference.

In this work, we developed a realtime eye tracking technique for autostereoscopic display systems. To avoid the drawback of wearing sensors or marks, we used a camera observing the user for tracking the eye positions of the user in the acquired image sequence. In each image, we employ the template matching technique to track the four motion parameters (X and Y translation, scaling, and rotation) of the user's face. Then, the left and right eyes can be located in the obtained face region. One of the major difficulties of applying the template matching technique is that the computational cost is extremely large. The reason is that the search space of tracking in these four degrees of freedom is very large. To meet the realtime requirements, we applied a fast template matching algorithm, called the winner-update algorithm [1], to speed up the tracking process.

2 Video-based Eye Tracking

In this section, we will describe the proposed video-based eye tracking technique. We will first address some design issues considering the eye tracking technique for autostereoscopic displays. Next, we will depict the flowchart of the proposed eye tracking technique. Then, we will describe each component of the proposed eye tracking technique in detail.

2.1 Design Issues

2.1.1 Consideration of User Behavior

In our eye tracking system, we mounted a video camera on the display for observing the user. When the user is watching the display, the frontal face of the user should appear in the acquired image. In an autostereoscopic display system, it is not necessary to keep tracking the user's eyes all the time. Instead, the system has to track the user's eyes only when the user is watching the autostereoscopic display. This will simplify the tracking problem because correct tracking is required only for the frontal face. Moreover, we assume that the user intends to watch the autostereoscopic display in a comfortable way. That is, the user will not look askew at the display on purpose. The user can look at the display at different positions in 3-D space but keeping his/her face toward the display. Sitting in front of the display, the easiest way to change the horizontal viewing position is to rotate the body around the frontal axis of the user. Consequently, the user may also undergo rotation around the normal axis of the image plane in the image. Moreover, the size of the user's face in the image varies with the distance between the user and the camera. Thus, a scaling parameter which describes the size of the user's face in the image has to be updated when the user moves nearer toward or farther away from the display. To sum up, there are four parameters to be estimated, which are the translations in the X and Y axes, the scaling, and the rotation around the normal axis of the image.

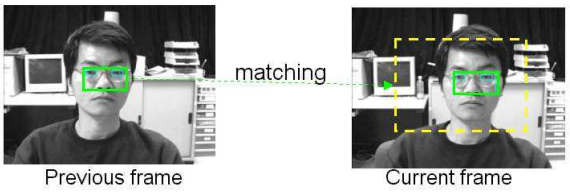

Figure 2. An example of face tracking by using template matching.

2.1.2 Tracking Using Template Matching

Template matching technique has been frequently used for visual tracking due to its simplicity and robustness. As shown in Figure 2, the face of the user can be tracked by using the face image in the previous frame as a template and match it within a search range in the current frame. However, there are two major disadvantages of the template matching technique. The first one is that the computational cost of the template matching is so large that its applicability is restricted, especially for realtime systems. The other disadvantage is that it can only track rigid object undergoing X/Y translation motion. If the object in the image is rotating or changing the scale, the template matching technique may fail to keep the object in track. In order to meet the realtime requirement, we used a fast template matching algorithm, the winner-update algorithm [1], for computational speedup. Moreover, a multilevel conjugate direction search technique (MCDS) is proposed to search and track the template with different scale and rotation.

2.1.3 Eye Tracking and Face Tracking

The eye positions can be tracked by using the eye image as the template and matching in each image frame. However, it is not robust due to the following two reasons. The first one is that the images of the left and right eyes are similar. It is difficult to determine whether the tracked eye position belongs to the left eye or to the right eye. The second reason is that the images of the eyes are relatively small and the ambiguity of matching is larger because of less information content. To overcome these two problems, we first use the face image of the user as the template and match the face template in each image frame to track the face position. Once the face region is located in the image, the left and right eye positions can be obtained in the upper-left and upper-right parts of the face region, respectively. The face region includes eyes and nostrils, which are salient features in the face image. Hence, the stability and accuracy can be greatly enhanced by tracking the face first.

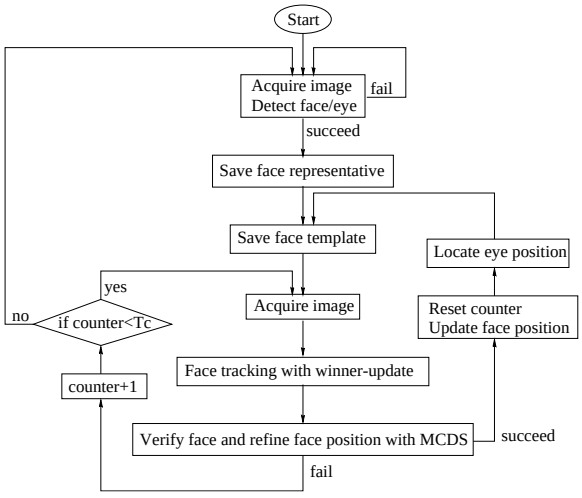

Figure 3. Flowchart of the proposed eye tracking technique.

2.2 Flowchart

Figure 3 shows the flowchart of the proposed eye tracking technique. The system first detects the face and eye positions in the acquired image. Then, the face image is stored as the face representative if detection succeeds. For each acquired image, the face template obtained in the previous image frame is used for matching in a search range by using the winner-update algorithm. Then, the candidate face position with minimum matching error is verified with the face representatives to determine whether it is actually a face position or not. Meanwhile, the face position is also refined by using MCDS technique. When the verification succeeds, the eye positions are detected and the face image is stored as the new face template. When the verification fails, on the other hand, the eye tracking system will recover as soon as possible by matching in a few consecutive image frames with the same face template. If this recovery also fails, the system will proceed face detection again and add a new face representative when the detect succeeds.

2.3 Components of the Eye Tracking Technique

This section will present in detail each component of the proposed eye tracking technique.

2.3.1 Image Preprocessing

The CCD camera we adopted is a conventional one with interlace scanning. That is, the even field scanlines and odd field scanlines in each image frame are exposed at different times. This may influence the tracking result if both fields are used for tracking process. For this reason, we discard the odd field scanlines and use only even field scanlines for tracking in each image frame. Furthermore, we perform averaging sub-sampling operation (4:1) for each scanline to smooth the image and reduce the noise. As a result, each 640×480 image frame acquired with the frame grabber becomes a 160×240 miniature. Because of the smaller image size, the search region of the template matching is also reduced thus will speed up the tracking process. Besides, we only track the upper part of the face including the eyes and nose because the upper part is relatively more rigid than the lower part of the face, which contains the mouth that may be speaking, laughing, or eating during the tracking process. After averaging sub-sampling operation, the rectangular upper face image is reduced to a square block, which can facilitate the winner-update algorithm for fast template matching.

2.3.2 Face Detection

When a new user appears in the image or when the tracking process has to be restarted because the user has been out of track for a certain period of time, the eye tracking system should be able to automatically detect the position of the user's face in the image. The face representative and face template can then be stored for the following tracking process. In this work, we adopt the eigenface method to detect the position of the user's face in the image. A set of training face images can form a matrix with each row representing the pixel values of a training face image. The eigen

space of smaller dimension can then be obtained by using KL transform. For the image block (with the same size as the training face images) at each position in the acquired image, it is first projected on the eigen space and its distance to the eigen space is calculated to determine this image block is a face image or not. If the distance between the image block under examination and eigen space is small enough, we can conclude that this image block contains a face and its position is reported. This face image block is stored as the first one of the representative face, which will be used for face verification and face position refinement during the following tracking process.

When the user is out of track, the eye tracking system can recovery from tracking failure by detecting the user again in the image. For faster recovery, however, the eye tracking system first uses all the available face representatives in turn as the template to match in the image, instead of using eigenface method. Because the template matching by using the winner-update algorithm is faster than the eigenface method, the system can then recover from tracking failure when one of the available representative face appears again in the image. That is, once the tracking process fails, it can be recovered soon when the user stop moving to return to normal pose. If all the face representatives are not successfully matched in the image, the eigen space method is then used for detecting the face position. A new face representative is stored if eigenface detection succeeds.

2.3.3 Face Tracking by Using Winner-Update Algorithm

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. The matching error criterion we used is the Sum of Absolute Difference (SAD):

$$\mathrm{SAD}(u,v) \equiv \sum_{j=0}^{B-1}\sum_{i=0}^{B-1} |T(i,j) - I_t(u+i, v+j)|,$$

where T is the template image with size $B \times B$, I_t is the current image, and (u, v) is the position in the search range $[-R, R] \times [-R, R]$. The position, $(\hat{u}, \hat{v})$, with minimum matching error can be found by calculating and comparing the SADs for all the search positions, $\{(u, v)\}$, in the image, I_t. That is,

$$(\hat{u}, \hat{v}) \equiv \arg \min_{(u,v)\in S} \mathrm{SAD}(u, v),$$

where $S = \{(u, v)| - R \leq u, v \leq R\}$ is the search region and R is an integer which determines the search range. This time-consuming operation can be greatly accelerated by using the winner-update algorithm. The obtained minimum matching error and the corresponding image position are used for further verification and refinement to determine this position contains a face image or not.

2.3.4 Face Verification

For each candidate face position with minimum matching error, the goal of face verification is to determine whether the image block at this position actually contains a face or not. Two criteria can be used for making this decision. The first one is that this minimum matching error should be small enough, that is, the image block at this candidate face position should "looks" similar to the face templates stored in the previous image frame. The second one is that this image block should "looks" similar to at least one of the face representatives. The minimum matching error between this image block and the face representatives should be small enough to justify that this image block is indeed a face image.

The second criterion of face verification is necessary because the tracking error might accumulate during the tracking process. For example, if A is similar to B and B is similar to C, A may be not similar to C. That is, the face position tracked might drift away slowly after a period of time. Consequently, we make use of the face representatives to perform the verification of the second criterion while refining the face position.

Because the user is allowed to rotate and change the scaling (moving near or far away from the display), the rotation and the scaling of the candidate face position may be different from those of the face representatives. We propose a MCDS technique to overcome this problem. When the face image is detected, not only the original face image is stored as the face representative but also the face images which are obtained by rotating and changing the scale of the original face image. Figure 4 illustrate an example of face images obtained with five different rotations and five different scalings. Image pyramid structure is constructed for each face image. During the matching process, the rotation and scaling dimensions are searched by using conjugate direction search. For each combination of rotation and scaling under examination, the image pyramid structure is used for speeding up the search process in a hierarchical manner (from the top level to the bottom level).

2.3.5 Face Position Refinement

Because of the accumulation error, the candidate face position may has drifted away from the correct face position. To improve the accuracy of the tracked face

Figure 4. Face representatives with different rotation and scaling.

Table 1. Convolution mask for eye detection.

1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	2	2	2	2	1	1
1	1	2	4	4	2	1	1
1	1	2	4	4	2	1	1
1	1	2	2	2	2	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1

position, we use the conjugate direction search technique to refine the face position along the gradient direction in the X and Y dimensions. Along the search path, the matching error using the face representatives is calculated. The position with minimum matching error is reported to be the refined face position. According to our experimental results, this method can effectively solve the drifting problem.

2.3.6 Eye Location

After obtaining the position of the user's face, we divide the face image block into four parts. According to the relative geometry between the eyes and the face, we perform eye detection in both upper-left and upper-right sub-blocks by using convolution operation with an 8×8 mask, as shown in Table 1. The position with smallest convolution result are considered as the eye position.

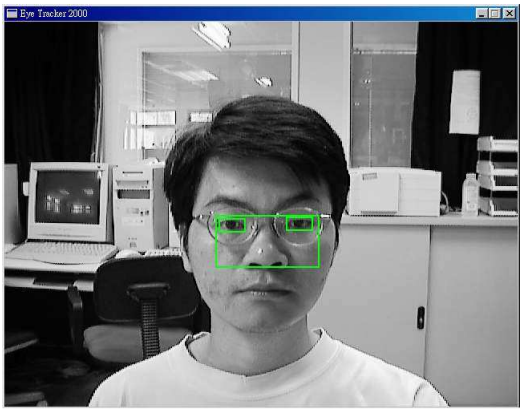

Figure 5. Face detection result.

2.3.7 Temporary Out of Track

When the verification of the candidate face position fails, the face position of the user is out of track and the eye tracking system has to recover from this failure. This situation occurs because of temporary occlusion. For example, the user waves his hand across his face or turns his face off the display. One way of recovery is to detect the face position all over again. However, this process is more time-consuming and should not be performed frequently. Instead of performing face detection, one can try to recover face tracking by simply acquiring the next image and using the template to match again for a specific period of time. This will be helpful when the user's face temporarily turns away or is temporarily occluded. The eye tracking system can recover from tracking failure faster than face detection.

3 Experiments

3.1 System Specification

In this section, we will describe the specification of our experimental eye tracking system. We implemented the proposed eye tracking technique on a PC with Pentium III 500. The video acquisition equipments we adopted include a Watec CCD camera, a Cosmicar 8.5mm lens, and a Matrox Meter II frame grabber. With these equipments, the frame rate of video acquisition is 30 Hz, which will limit the maximum frame rate and the minimum latency time of our eye tracking system. In our implementation, the eye tracking process and video acquisition are concurrently proceeded. Moreover, the eye tracking process for each image can be finished in less than 1/30 seconds. Consequently, the overall frame rate of our eye tracking system is 30 Hz and the latency time ranges from 1/30 to 2/30 seconds.

Figure 6. Examples of face and eye tracking.

3.2 Experimental Results

In our implementation, the image size of the face template is 32×32 and the search range in each image is 160×160. Notice that the image is preprocessed and reduced to 160×240 before the eye tracking proceeds. Figure 5 illustrate an example of face detection, where the original image (640×480) is shown for clarity. As shown in Figure 6, the face and eyes positions with different X/Y translation, rotation, and scaling have been located. The user can move around freely to different positions in 3-D and rotate his head in 1-D. The proposed eye tracking technique can track the face and eye positions of the user. Figure 7 shows that the system is in temporarily-out-of-track stage. Because the user turn his face off the display, the eye tracking system can not successfully track the frontal face. When this happens, the system does not have to deal with this situation because the user is not watching the display.

4 Conclusions

We have presented a video-based eye tracking technique which can accurately and robustly track the user's eye positions in video in realtime. This technique can be combined with an autostereoscopic system, which can provide the user stereo video without the requirement of wearing any special glasses or sensors. Fast template matching technique has been used to track the four motion parameters (X and Y translation, scaling, and rotation) of the user's face in each image. According to our implementation on a PC with Pentium III 500, the tracking frame rate can achieve 30 Hz.

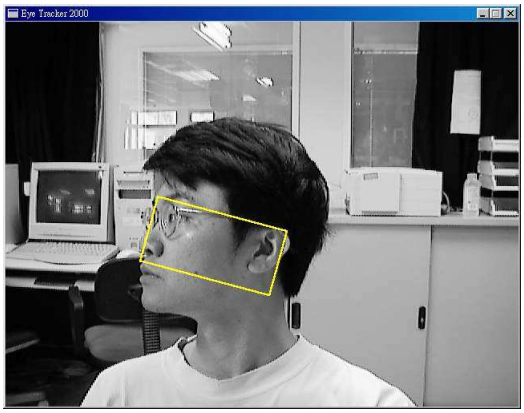

Figure 7. Face is temporarily missed.

Acknowledgements

This work is partially supported by Opto-Electronics & System Laboratory, Industrial Technology Research Institute, Hsinchu, Taiwan.

References

[1] 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 Asian Conference on Computer Vision*, volume 2, pages 977–982, Taipei, Taiwan, Jan. 2000.

[2] C. Morimoto, D. Koons, A. Amir, and M. Flickner. Pupil detection and tracking using multiple light sources. *IBM Almaden Research Center, http://www.almaden.ibm.com/cs/blueeyes/find.html*, 1998.

[3] 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*, 1(1):41–52, 1999.

[4] K. Talmi and J. Liu. Eye and gaze tracking for visually controlled interactive stereoscopic displays. *Signal Processing: Image Communication*, 14(10):799–810, 1999.