

# Video-based Eye Tracking for Autostereoscopic Displays

Yong-Sheng Chen, Chan-Hung Su, Jiun-Hung Chen,  
Chu-Song Chen, Yi-Ping Hung\*, and Chiou-Shann Fuh

\*Corresponding author. Correspondence should be sent to Yi-Ping Hung, Institute of Information Science 20, Academia Sinica, 128 Academia Road, Section 2, Nankang, Taipei 115, Taiwan.

Yong-Sheng Chen is currently a postdoctoral research fellow in the Institute of Information Science, Academia Sinica, Taipei, Taiwan. E-mail: yschen@iis.sinica.edu.tw. Tel.: +886-2-27883799 ext. 2413. Fax: +886-2-27824814.

Chan-Hung Su is currently a graduate student in the Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, and a research assistant in the Institute of Information Science, Academia Sinica, Taipei, Taiwan. E-mail: rey@iis.sinica.edu.tw.

Jiun-Hung Chen is currently a research assistant in the Institute of Information Science, Academia Sinica, Taipei, Taiwan. E-mail: jhchen@iis.sinica.edu.tw.

Chu-Song Chen is an assistant research fellow in the Institute of Information Science, Academia Sinica, Taipei, Taiwan. E-mail: song@iis.sinica.edu.tw.

Yi-Ping Hung is a research fellow in the Institute of Information Science, Academia Sinica, Taipei, Taiwan, and an adjunct professor in the Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan. E-mail: hung@iis.sinica.edu.tw. Tel.: +886-2-27883799 ext. 1718. Fax: +886-2-27824814.

Chiou-Shann Fuh is a professor in the Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan. E-mail: fuh@csie.ntu.edu.tw. Tel.: +886-2-23625336 ext. 327. Fax: +886-2-23628167.

### Abstract

An autostereoscopic display system can provide users the great enjoyment of stereo visualization without the uncomfortable and inconvenient drawbacks of wearing stereo glasses or head-mounted displays. To render stereo video with respect to the user's viewpoints and to accurately project stereo video onto the user's eyes, the left and right eye positions of the user, who is allowed to move around freely, have to be obtained when the user is watching the autostereoscopic display. We present real-time tracking techniques that can efficiently provide the user's eye positions in images. These techniques comprise: (1) face detection by using multiple eigenspaces of various lighting conditions, (2) fast block matching for tracking four motion parameters ( $X$  and  $Y$  translation, scaling, and rotation) of the user's face, and (3) eye locating in the obtained face region. According to our implementation on a PC with a Pentium III 700 MHz CPU, the frame rate of the eye tracking process can achieve 30 Hz.

### Keywords

eye tracking; face tracking; autostereoscopic displays.

## I. INTRODUCTION

Virtual reality systems are becoming more and more attractive in the applications of education, exhibition, training, and entertainment. One of the major components of a virtual reality system is the stereoscopic display, providing users with a stereo visual environment. Conventionally, users of a virtual reality system have to wear on their heads stereo glasses or head-mounted displays (HMDs), which will make them feel less comfortable and thus cannot immersively enjoy the virtual environment.

Recently, researchers began to develop autostereoscopic display systems, [1], [2], [3] which can provide the enjoyment of stereo visualization without the requirement of wearing any special device. In a look-around system, as shown in Fig. 1, a user can move around freely in front of the autostereoscopic display to watch the stereo video from various 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. Thus, the autostereoscopic display system can render the stereo video with respect to the viewpoint 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 fluently rendering and accurately projecting the stereo video.

One kind of tracking method requires the user to wear some special sensors, such as infrared sensors or reflectors, ultrasonic wave receivers, and electromagnetic wave sensors. However,

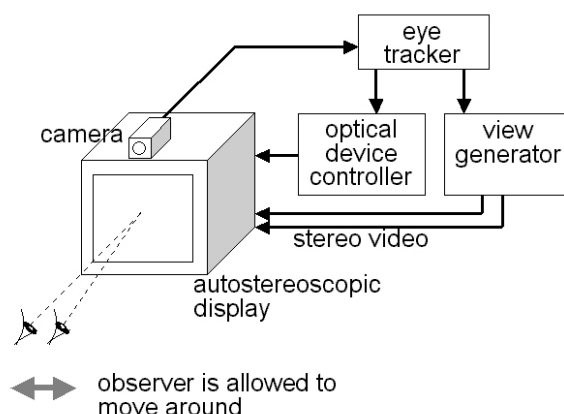


Fig. 1. A look-around system.

this kind of active sensing method is uncomfortable and inconvenient. Moreover, these sensors cannot be mounted on the eyes, thus the obtained positions are not the exact eye positions. Therefore, video-based methods are preferable for tracking users in a passive manner. For example, Azarbajejani et al. [4] developed interactive graphics systems—virtual holography and teleconferencing systems. They used a camera to observe a user. Image features of the user can be extracted and can help to track the motion of the user’s head. The graphics systems can then be controlled according to the tracking results. Instead of using features as in Ref. [4], another option is to use skin-color information for tracking the user’s facial region [5], [6], [7]. Shirai [8] also developed a 3-D human tracking system by using optical flow and depth information observed from an active camera head. Recently, Stauffer and Grimson [9] developed tracking techniques for visual monitoring system based on an adaptive background subtraction method.

Video-based tracking techniques are often adopted in an autostereoscopic display system to track the user’s eye positions in a passive manner [10]. For example, Pastoor et al. [2] built an experimental multimedia system that can be controlled by eye movement. Their system uses a camera to track the head position, eye position, and gaze direction in the acquired video. Morimoto et al. [11] also proposed a pupil detection and tracking technique. Two sets of LEDs were mounted on a camera lens, one set is on-axis and the other is off-axis. 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 image difference.

In this work, we develop real-time eye tracking techniques for autostereoscopic display sys-

tems. To avoid the drawback of wearing sensors or marks, we use a camera observing a user for tracking his/her eye positions in the acquired image sequence. In the initial stage, we use the eigenface method [12] to detect the face position of the user. Multiple eigenspaces are utilized to accommodate various lighting conditions. Then, we employ the block matching technique to track in the following images the four motion parameters ( $X$  and  $Y$  translation, scaling, and rotation) of the user's face. One of the major difficulties of applying the block matching technique is due to its large computational cost. The reason is that the search space of tracking in these four degrees of freedom is very large. To meet the real-time requirements, we apply a fast block matching algorithm, called the winner-update algorithm [13], [14], to speed up the tracking process. Once the face position of the user is obtained, the left and right eyes can be located in the face region.

## II. VIDEO-BASED EYE TRACKING

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

### A. Design Issues

#### A.1 Consideration of User Behavior

In our eye tracking system, we mount a video camera on top of the display to observe 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 the 3-D space while 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

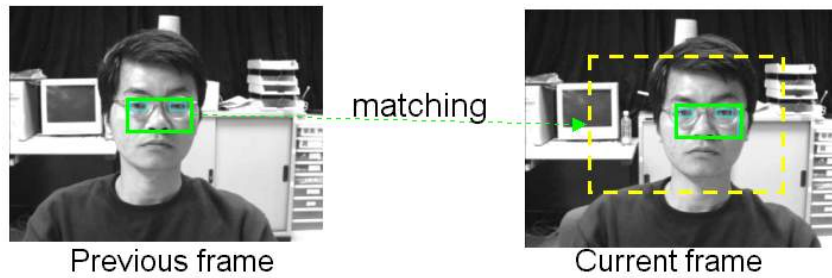


Fig. 2. Illustration of face tracking by using block matching.

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 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.

## A.2 Tracking Using Block Matching

The block matching technique has been frequently used for visual tracking due to its simplicity and robustness. As shown in Fig. 2, the face of the user can be tracked by using the face image in the previous frame as a template image block, and match it within a search range in the current frame. However, there are two major disadvantages of the block matching technique. The first one is that its computational cost is so large that its applicability is restricted, particularly for real-time systems. The other disadvantage is that it can only track a rigid object undergoing  $X/Y$  translation motion. If the object in the image is rotating or changing its scale, the block matching technique may fail to keep the object in track. To meet the real-time requirement, we adopt a fast block matching algorithm, the winner-update algorithm [13], [14], for computational speedup. Moreover, a multilevel conjugate direction search technique (MCDS) is proposed to search and track the template image block, considering its scale and rotation motion parameters in addition to its  $X/Y$  translation parameters.

## A.3 Eye Tracking and Face Tracking

The eye positions can be tracked by matching the eye image (as a template image block) in consecutive image frames. However, this tracking method is not robust due to the following two

problems. 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. One simple way to solve this problem is to apply the geometric constraint of the left and right eye positions. The second problem is that the eye images are relatively small. Hence, the matching results are more ambiguous because of less information content. To overcome these two problems and to increase the accuracy of the tracking results, we match the face image instead, which is a larger image block compared to the eye image. The face positions of the user are first tracked in consecutive image frames by using the block matching technique. 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. The face region comprises eyes and nose, which are salient features in the face image. Hence, the stability and accuracy can be greatly enhanced by tracking the face first.

### *B. Flowchart*

Fig. 3 shows the flowchart of the proposed eye tracking techniques. Initially, we repeat the process of acquiring an image and detecting the face region in the image until the detection succeeds. Once a face region is found, we save the image of this face region as a long-term face template, which is called a face representative. Next, the left and right eye positions can be located within their corresponding area in the face region obtained either through face detection or through face tracking. Also, the image of the face region obtained in both detection and tracking cases is saved as a short-term face template. (In the following, we refer to the short-term face template as the face template for simplicity.) To track the face position in the following stage, we use the face template for matching in the coming image. After a new image is acquired, the winner-update algorithm [13], [14] is adopted to match the face template, very efficiently, in a search range of the newly acquired image. Then, the candidate face position having the minimum matching error is verified by using the face representative (long-term face template) to determine whether the image region located at this candidate position actually contains a face. Meanwhile, the face position is also refined by using the MCDS technique. When the verification succeeds, the eye positions are estimated in the face region. Also, the face template is updated by using the newly obtained image of the face region. This iteration is repeated to track the face and eye positions in consecutive image frames. When the verification fails, on the

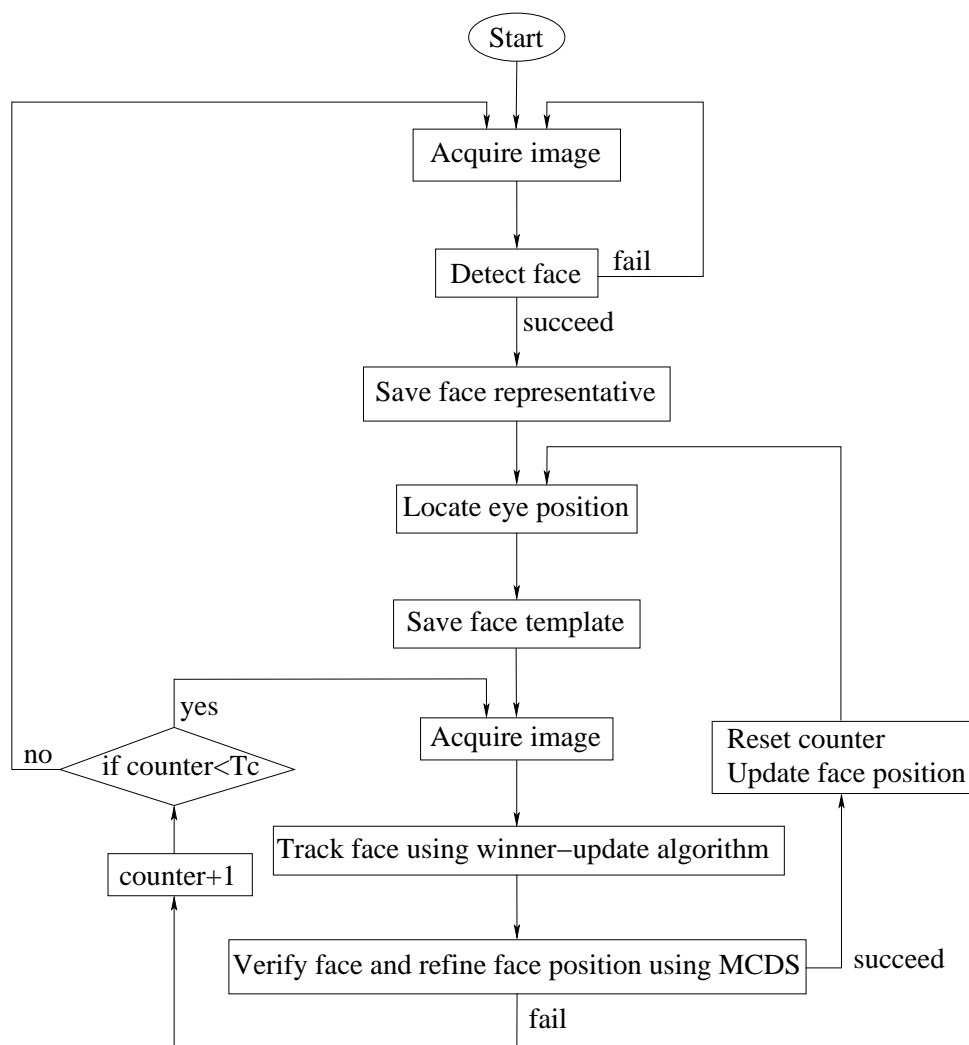


Fig. 3. Flowchart of the proposed eye tracking technique.

other hand, we want to recover the tracking process as soon as possible. Hence, we keep the face template without modification and match the face template in a few consecutive image frames. If this recovery also fails, we reset the tracking process by proceeding face detection and use a new face representative after succeeding in the detection.

### C. Components of the Eye Tracking Technique

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

## C.1 Image Preprocessing

For each acquired image frame, we discard its odd field scanlines and use only its even field scanlines. Also, we perform an averaging subsampling operation (4:1) for each scanline to smooth the image and reduce the noise. As a result, each  $640 \times 480$  image frame acquired from the camera becomes a  $160 \times 240$  miniature. Because of the smaller image size, the search region for the block matching is also reduced thus will speed up the tracking process. 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. By using this averaging subsampling operation, the rectangular upper face image is reduced to a square block, which can facilitate the winner-update algorithm [13], [14] for fast block matching. In general, the winner-update algorithm does not require a square image block. However, it is beneficial to use a square image block due to easier implementation and computation of the winner-update algorithm, which is briefly described later in Sec. II-C.3.

## C.2 Face Detection

Automatic face detection has to be performed 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. In this work, we adopt the eigenface method [12] 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 eigenspace of smaller dimension can then be obtained by using the principal component analysis. For the image block (with the same size as 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. If the distance from the image block under examination to the eigenspace 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 face representative, which will be used for face verification and face position refinement during the following tracking process.

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 very different from the light-



ing 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. One research direction regarding the lighting variation problem is to utilize the illumination invariant formulation, such as in Ref. [15]. For computational simplicity and efficiency, we use another approach to reduce the lighting variation problem in this work. We assume that there is a finite number of ordinary lighting conditions in the environment of the autostereoscopic display system. For each lighting condition, we collect a set of training images and compute their corresponding eigenspace. Each eigenspace comprises its mean and eigenvectors. That is, we compute multiple eigenspaces for various lighting conditions. When determining whether an image block contains a face, we first calculate its distances to the means of all the eigenspaces. Next, we choose the eigenspace yielding the minimum distance for further face estimation, because the lighting condition of the image block under examination is more likely to be the same as that of the chosen eigenspace. The standard eigenface method [12] is then performed by using the chosen eigenspace to determine whether the image block contains a face.

Another difficulty of the eigenface method for face detection is the geometric variation problem, such as scaling and rotation. To reduce the computational cost, we do not consider the rotation variation and only consider the scaling variation of the frontal face. For the image on which the face detection is performed, we scale up and down the image. The previously mentioned face detection method that uses multiple eigenspaces is then performed on every resulted image of different scale. Fig. 4 shows some examples of face detection results for different lighting conditions and scales. Notice that the images are subsampled into a  $\frac{1}{4} \times \frac{1}{2}$  miniature during the image preprocessing stage.

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, we first use the face representative as the matching template to match in the image, instead of using the eigenface method. Because the block matching technique using the winner-update algorithm is faster than the eigenface method, the system can then recover from tracking failure when the face representative appears again in the image. That is, once the tracking process fails, it can be recovered soon when the same user stops moving to return to a normal pose. If the face representative is not successfully matched in the image, the eigenspace method is then used for detecting the face

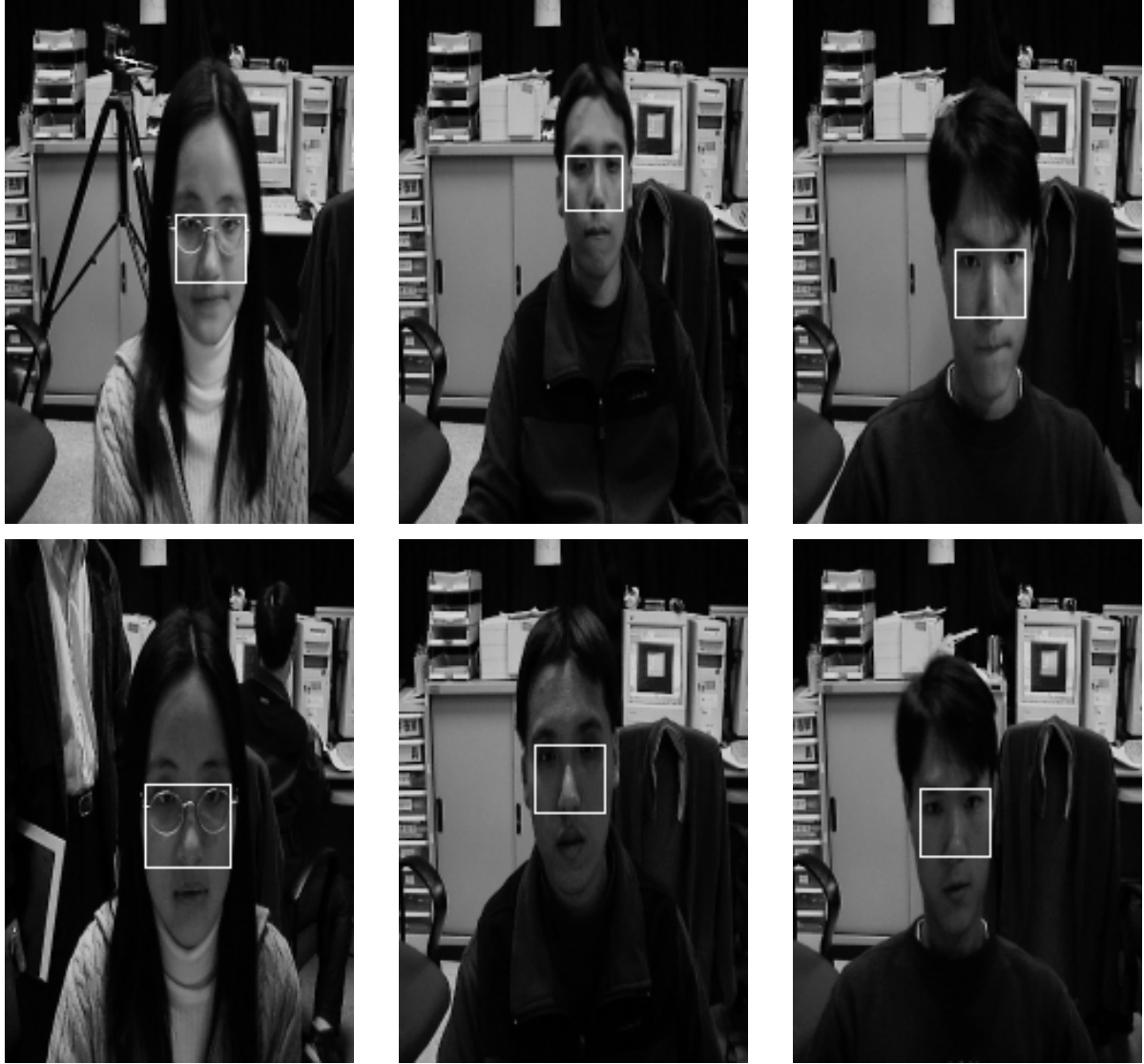


Fig. 4. Examples of face detection results for different lighting conditions and scales.

position. A new face representative is stored if the eigenface detection succeeds.

### C.3 Fast Face Tracking 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 block matching within a search range in the consecutive image. The matching error criterion we used is the sum of absolute difference (SAD):

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

where  $T$  is the template image block of size  $B \times B$ ,  $I_t$  is the current image, and  $(u, v)$  is a position in the search range  $[-R, R] \times [-R, R]$ . The position,  $(\hat{u}, \hat{v})$ , yielding the 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,

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

where  $S = \{(u, v) | -R \leq u, v \leq R\}$  is the search region and  $R$  is an integer that determines the search range. This time-consuming operation can be greatly accelerated by using the winner-update algorithm [13], [14]. The obtained minimum matching error and the corresponding image position are used for further verification and refinement to determine whether this position contains a face image.

Most of fast block matching methods gain their efficiency by reducing the search positions under examination. Criterion for this reduction can be gradient descent [16], multiresolution search space [17], and spatial-temporal correlation [18]. The major drawback for these kinds of method is that they may be trapped at the local minimum, thus reduce the accuracy of the tracking result.

To achieve high accuracy, we adopt the winner-update algorithm, which guarantee that the global minimum of the block matching can be found efficiently. This algorithm first constructs a block sum pyramid (BSP) [19] for the template image block and for each image block in the search region. The SAD calculated by using the upper levels of the corresponding pyramid pairs can be proved [19] to be the lower bound of the desired SAD calculated by using the bottom levels, which contain the original images of the template image block and the image block in the search region. For a search position, once one of its lower bounds of SAD calculated at some upper level is already larger than the global minimum SAD, this search position can be skipped for further consideration. The computational cost of the SAD lower bound is smaller than that of SAD itself, because the lower bound is calculated at the upper level of the pyramid. Moreover, we reduce many of the SAD lower bounds actually calculated by using the winner-update search strategy [13], [14]. This search strategy initially calculates the first SAD lower bounds for all the search positions by using the top levels of their pyramids. The one having the minimum lower bound is selected as the temporary winner, and its next lower bound is calculated by using its lower level. Then, the temporary winner is selected again from all the search positions.

This process is repeated until the minimum SAD lower bound of the newly selected temporary winner is calculated at the bottom level; that is, the SAD lower bound is the desired SAD. In this way, a substantial computation can be saved, and thus speeds up the block matching process. In Refs. [13] and [14], we also proposed an efficient method, in terms of computation and storage, to construct pyramids for all the search positions in the search region. It is beneficial to use the square image block due to the simplicity of the pyramid construction and the SAD lower bound calculation.

#### C.4 Face Verification and Face Position Refinement

For each 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. 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 look similar to the face templates stored in the previous image frame. The second one is that this image block should look similar to the face representative. The matching error between this image block and the face representative should also 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 may accumulate during the tracking process. For example, if  $A$  is similar to  $B$  and  $B$  is similar to  $C$ ,  $A$  may not be similar to  $C$ . That is, the face position tracked may drift away slowly after a period of time. Consequently, we make use of the face representative, which is a long-term face template, to perform the verification of the second criterion while refining the face position.

Because the user is allowed to rotate his/her head and to move 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 representative, which is recorded when the face is detected. 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. A multilevel conjugate direction search technique is then used to search for the combination, such that the corresponding face representative is similar to the candidate image block.

When the face region is detected, not only the original face image block is stored as the face representative, but also the face image blocks that are obtained by rotating and scaling

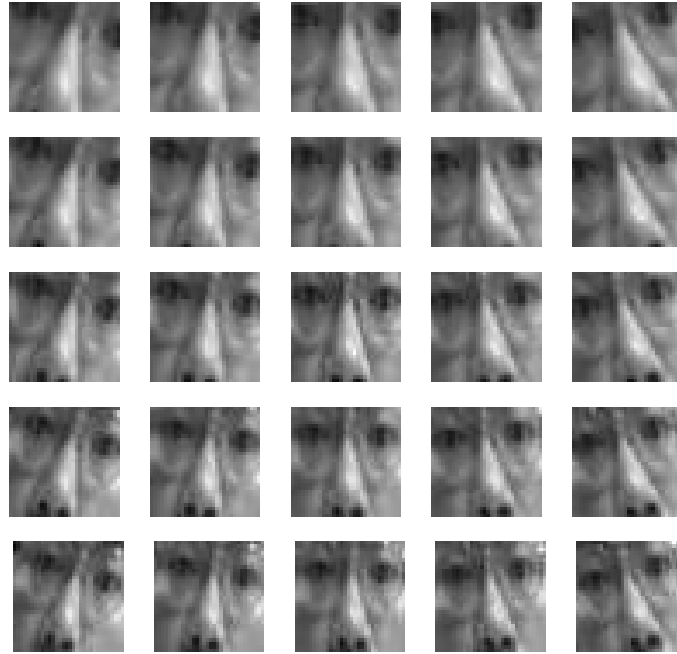


Fig. 5. Face representatives comprising various combinations of rotation (horizontal direction) and scaling (vertical direction).

up and down the original face image block. Fig. 5 illustrates an example comprising 25 face representatives of five different rotations and five different scales. A block sum pyramid structure is then constructed for each face representative.

During the matching process, two dimensions of the scaling and rotation can be searched by using the one-at-a-time search, which is a simplified conjugate direction search [16], on each level of image pyramids. On the top level, we find the combination of scaling and rotation yielding the local minimum of SAD by using the following process. Starting from the initial combination of scaling and rotation, we move along the scaling direction first and find the local minimum of SAD by using the gradient descent search. Then, starting from this local minimum, we move along the rotation direction and find the new local minimum of SAD by using the gradient descent search again. The combination of scaling and rotation yielding the new local minimum is used as the initial combination for the next (lower) level. By repeating this one-at-a-time search from the top level to the bottom level, we can obtain the combination of scaling and rotation yielding the local minimum of SAD at the bottom level. We utilize the pyramid structures because it is more efficient due to the coarse-to-fine search, and we have more chances

TABLE I  
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

to bypass the local minimum and to achieve the global minimum due to the smoother error surface on the upper levels, which contain smoother images.

Because of the accumulation error, the candidate face position may have drifted away from the correct face position. To improve the accuracy of the tracked face position, we utilize the similar conjugate direction search technique to refine the face position along the gradient directions in the  $X$  and  $Y$  dimensions, one at a time. Along the search path, the matching error using the face representatives is calculated. The position yielding the minimum matching error is reported to be the refined face position. According to our experimental results, this method can effectively overcome the drifting problem.

### C.5 Eye Location

Once the face position is determined, we can find the left and right eye positions in the corresponding area within the face region. Instead of feature-based methods [20], we adopted a convolution-based method for simplicity and efficiency.

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 the convolution operation with an  $8 \times 8$  mask, as shown in Table I. The position with the smallest convolution result, which means the darkest region, is considered as the eye position.

### C.6 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 face template to match again for a prespecified period of time. This will be helpful when the user temporarily turns his/her face away or his/her face is temporarily occluded. The eye tracking system can recover from tracking failure faster in this way.

## III. EXPERIMENTS

### A. System Specification

In this section, we describe the specification of our experimental eye tracking system. We implemented the proposed eye tracking technique on a PC with a Pentium III 700 Mhz CPU. The video acquisition equipment we used is a Sony DFW-VL500 digital camera, which adopts the IEEE-1394 standard. The video frame rate of this camera 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; that is, the video frame rate. The latency time ranges from 1/30 to 2/30 seconds.

### B. Experimental Results

In our implementation, the image size of the face template is  $32 \times 32$  and the search range in each image is  $[-80, 80] \times [-80, 80]$ . Notice that the image is reduced to  $160 \times 240$  in the preprocessing stage before the eye tracking proceeds. Fig. 6 illustrates an example of face detection, where the original image ( $640 \times 480$ ) is shown for clarity. As shown in Fig. 7, the face and eye 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

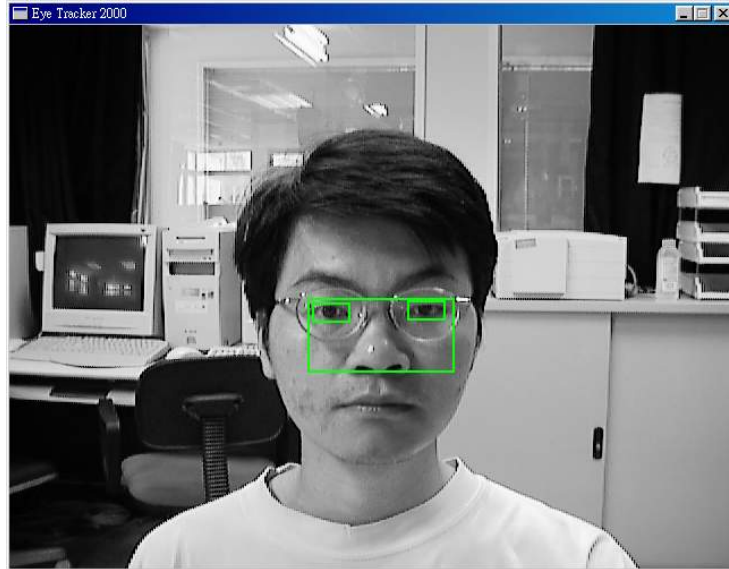


Fig. 6. Example of face detection result.

proposed eye tracking technique can track the face and eye positions of the user. Fig. 8 shows that the system is in the temporary out-of-track stage. Because the user turn his face off the display, the eye tracking system cannot 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.

Fig. 9 shows the tracking results in an image sequence. The  $X/Y$  positions, scales, and rotations of the face and eyes are estimated. When the user's face is occluded by his hand, the tracking process fails. After the user takes his hand off his face, the tracking process is recovered.

#### IV. CONCLUSIONS

We have presented video-based eye tracking techniques that can accurately, robustly, and efficiently track the user's face and eye positions in image sequence. When combined with an autostereoscopic system, the resulted look-around system can provide the user stereo video without the requirement of wearing any special glasses or sensors. A fast block 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 a Pentium III 700 Mhz CPU, the frame rate of the face and eye tracking can achieve 30 Hz.





Fig. 7. Examples of face and eye tracking.

#### ACKNOWLEDGEMENTS

This work is partially supported by Opto-Electronics & Systems Laboratories, Industrial Technology Research Institute, Hsinchu, Taiwan. The authors would like to thank Mr. Iwan Ulrich at CMU for his support of the IEEE-1394 digital camera windows driver.

#### REFERENCES

- [1] Graham J. Woodgate, David Ezra, Jonathan Harrold, Nicolas S. Holliman, Graham R. Jones, and Richard R. Moseley, "Autostereoscopic 3D display systems with observer tracking," *Signal Processing: Image Communication*, vol. 14, pp.

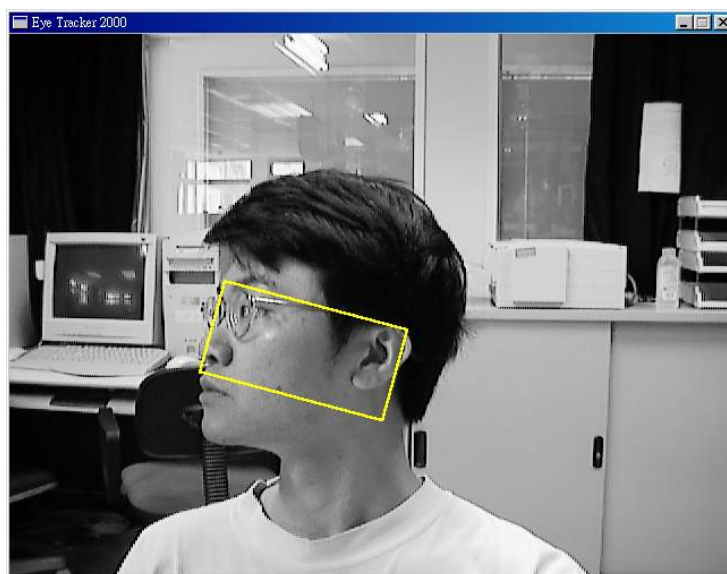


Fig. 8. Face of the user is temporarily missing.

- 131–145, 1998.
- [2] Siegmund Pastoor, Jin Liu, and Sylvain 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, pp. 41–52, 1999.
  - [3] C. Tsai, N. Wang, S. Lee, W. Hsueh, and C. Lee, “A novel viewer-tracking-based autostereoscopic display using horizontal-stripe barriers,” in *Proceedings of the Seventh International Display Workshop*, Kobe, Japan, Nov. 29–Dec. 1, 2000.
  - [4] A. Azarbayejani, T. Starner, B. Horowitz, and A. Pentland, “Visually controlled graphics,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 6, pp. 602–605, 1993.
  - [5] Paul Fieguth and Demetri Terzopoulos, “Color-based tracking of heads and other mobile objects at video frame rates,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Puerto Rico, June 1997, pp. 21–27.
  - [6] 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, pp. 359–388, 1999.
  - [7] T. Darrell, G. Gordon, M. Harville, and J. Woodfill, “Integrated person tracking using stereo, color, and pattern detection,” *International Journal of Computer Vision*, vol. 37, no. 2, pp. 175–185, 2000.
  - [8] 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*, Ottawa, Ontario, Canada, May 1997, pp. 130–139.
  - [9] Chris Stauffer and W. Eric L. Grimson, “Learning patterns of activity using real-time tracking,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 747–757, 2000.
  - [10] Kay Talmi and Jin Liu, “Eye and gaze tracking for visually controlled interactive stereoscopic displays,” *Signal Processing: Image Communication*, vol. 14, no. 10, pp. 799–810, 1999.
  - [11] Carlos Morimoto, Dave Koons, Arnon Amir, and Myron Flickner, “Pupil detection and tracking using multiple light sources,” *IBM Almaden Research Center*, 1998, <http://www.almaden.ibm.com/cs/blueeyes/find.html>.
  - [12] Baback Moghaddam and Alex Pentland, “Probabilistic visual learning for object representation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 696–710, 1997.

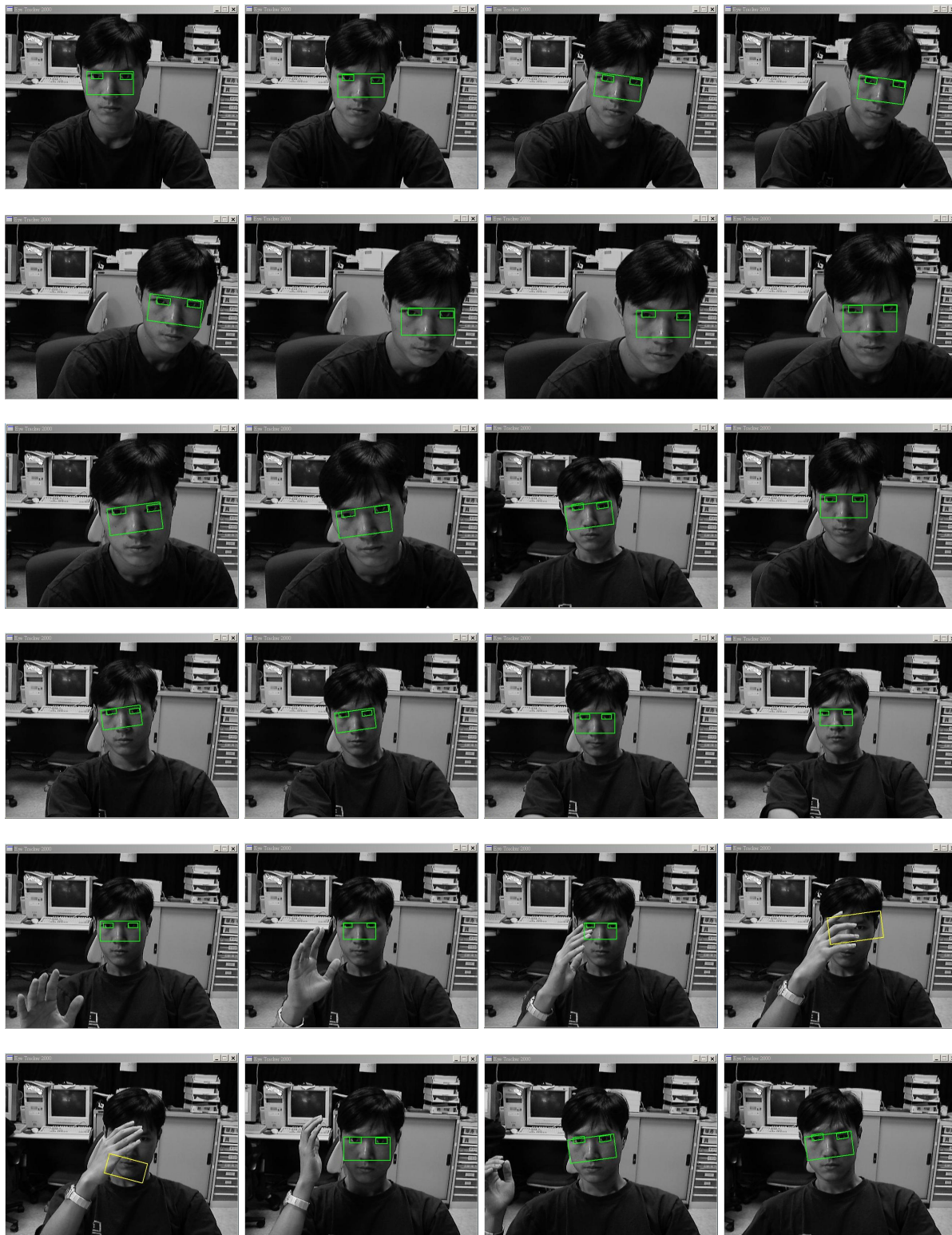
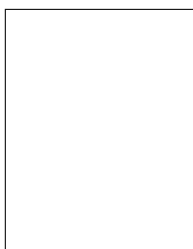
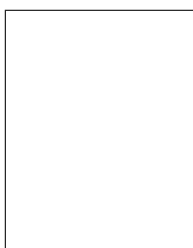


Fig. 9. An example of face and eye tracking in an image sequence of about 6 seconds. Only one eighth of the image frames regularly sampled are shown.

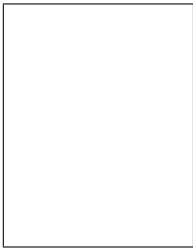
- [13] Yong-Sheng Chen, Yi-Ping Hung, and Chiou-Shann Fuh, “A fast block matching algorithm based on the winner-update strategy,” in *Proceedings of the Asian Conference on Computer Vision*, Taipei, Taiwan, Jan. 2000, vol. 2, pp. 977–982.
- [14] Yong-Sheng Chen, Yi-Ping Hung, and Chiou-Shann Fuh, “Fast block matching algorithm based on the winner-update strategy,” *IEEE Transactions on Image Processing*, vol. 10, no. 8, pp. 1212–1222, Aug. 2001.
- [15] Ronald Alferez and Yuan-Fang Wang, “Geometric and illumination invariants for object recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 6, pp. 505–536, 1999.
- [16] Ram Srinivasan and K. R. Rao, “Predictive coding based on efficient motion estimation,” *IEEE Transactions on Communications*, vol. COM-33, no. 8, pp. 888–896, 1985.
- [17] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, “Motion-compensated interframe coding for video conferencing,” in *Proceedings of National Telecommunications Conference*, New York, 1981, vol. 4, pp. G5.3.1–G5.3.5.
- [18] Junavit Chalidabhongse and C.-C. Jay Kuo, “Fast motion vector estimation using multiresolution-spatio-temporal correlations,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 3, pp. 477–488, 1997.
- [19] Chang-Hsing Lee and Ling-Hwei Chen, “A fast motion estimation algorithm based on the block sum pyramid,” *IEEE Transactions on Image Processing*, vol. 6, no. 11, pp. 1587–1591, 1997.
- [20] Kin-Man Lam and Hong Yan, “Locating and extracting the eye in human face images,” *Pattern Recognition*, vol. 29, no. 5, pp. 771–779, 1996.



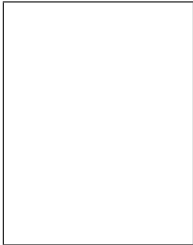
**Yong-Sheng Chen** received the B.S. degree in computer and information science from National Chiao Tung University, Hsinchu, Taiwan, in 1993, and the M.S. degree and the Ph.D. degree in computer science and information engineering from National Taiwan University, Taipei, Taiwan, in 1995 and 2001, respectively. He is currently a postdoctoral research fellow in the Institute of Information Science, Academia Sinica, Taipei, Taiwan. His research interests include fast template matching, pattern recognition, object detection, visual tracking, and video surveillance.



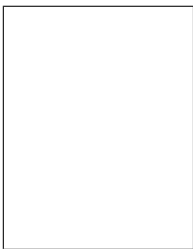
**Chan-Hung Su** received the B.S. degree in computer science and information engineering from National Taiwan University, Taipei, Taiwan, in 2000. He is currently a research assistant in the Institute of Information Science, Academia Sinica, Taipei, Taiwan, and a master student at National Taiwan University. His research interests include computer vision, image processing, and pattern recognition.



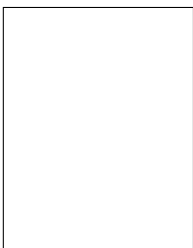
**Jiun-Hung Chen** received the B.S. and the M.S. degrees in computer science and information engineering from National Taiwan University, Taipei, Taiwan, in 1997 and 1999, respectively. He is currently a research assistant in the Institute of Information Science, Academia Sinica, Taipei, Taiwan. His research interests include pattern recognition, image processing, and computer vision.



**Chu-Song Chen** received a B.S. in control engineering from National Chiao Tung University, Hsingchu, Taiwan, in 1989. He received an M.S. in 1991 and a Ph.D in 1996, respectively, both from the Department of Computer Science and Information Engineering, National Taiwan University. From 1997 to 1999, he was a postdoctoral fellow of the Institute of Information Science, Academia Sinica, and has become an assistant research fellow since 1999. He has received both the outstanding paper award from the Image Processing and Pattern Recognition (IPPR) Society and the best paper award from the Image Processing and Application Association (IPAA) of Taiwan, in 1997. He has received the outstanding paper award for the application of computers in ICS'2000 held in Chiayi, Taiwan. His research interests include computer vision, pattern recognition, and computer graphics.



**Yi-Ping Hung** received his B.S. in electrical engineering from National Taiwan University in 1982. He received an M.S. from the Division of Engineering, an M.S. from the Division of Applied Mathematics, and a Ph.D. from the Division of Engineering, all at Brown University, in 1987, 1988 and 1989, respectively. He then joined the Institute of Information Science, Academia Sinica, Taiwan, and became a research fellow in 1997. He served as the deputy director of the Institute of Information Science from 1996 to 1997, and received the Outstanding Young Investigator Award given by Academia Sinica in 1997. He has been teaching in the Department of Computer Science and Information Engineering at National Taiwan University since 1990, where he is now an adjunct professor. He has published more than 70 technical papers in the fields of computer vision, pattern recognition, image processing, and robotics. In addition to these topics, his current research interests also include visual surveillance, virtual reality, and human-computer interface.



**Chiou-Shann Fuh** received the BS degree in computer science and information engineering from National Taiwan University, Taipei, Taiwan, in 1983, the MS degree in computer science from the Pennsylvania State University, University Park, Pennsylvania, in 1987, and the PhD degree in computer science from Harvard University, Cambridge, Massachusetts, in 1992. He was with AT&T Bell Laboratories and engaged in performance monitoring of switching networks from 1992 to 1993. He was an associate professor in the Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan from 1993 to 2000, and was then promoted to a full professor. His current research interests include digital image processing, computer vision, pattern recognition, and mathematical morphology.