# AUTOMATIC DETECTION AND TRACKING OF HUMAN HEADS USING AN ACTIVE STEREO VISION SYSTEM

CHENG-YUAN TANG* and ZEN CHEN

*Institute of Computer Science and Information Engineering*
*National Chiao Tung University, Hsinchu, Taiwan*

YI-PING HUNG†

**Institute of Information Science*
*Academia Sinica, Taipei, Taiwan*
*E-mail: hung@iis.sinica.edu.tw*

A new head tracking algorithm for automatically detecting and tracking human heads in complex backgrounds is proposed. By using an elliptical model for the human head, our Maximum Likelihood (ML) head detector can reliably locate human heads in images having complex backgrounds and is relatively insensitive to illumination and rotation of the human heads. Our head detector consists of two channels: the horizontal and the vertical channels. Each channel is implemented by multiscale template matching. Using a hierarchical structure in implementing our head detector, the execution time for detecting the human heads in a 512 × 512 image is about 0.02 second in a Sparc 20 workstation (not including the time for image acquisition). Based on the ellipse-based ML head detector, we have developed a head tracking method that can monitor the entrance of a person, detect and track the person's head, and then control the stereo cameras to focus their gaze on this person's head. In this method, the ML head detector and the mutually-supported constraint are used to extract the corresponding ellipses in a stereo image pair. To implement a practical and reliable face detection and tracking system, further verification using facial features, such as eyes, mouth and nostrils, may be essential. The 3D position computed from the centers of the two corresponding ellipses is then used for fixation. An active stereo head has been used to perform the experiments and has demonstrated that the proposed approach is feasible and promising for practical uses.

*Keywords*: Head tracking; maximum likelihood head detector; multiscale; face detection; active vision; stereo vision.

## 1. INTRODUCTION

Tracking human motion with computer vision techniques has been a popular research topic,[1,2,5,7,9,13,14,21,23,28,33] due to potential applications for surveillance, security and human–computer interface. Many researchers are specifically interested in tracking human heads or faces.[1,2,7,9,13,21,28,33] In this paper, we propose an efficient maximum likelihood (ML) head detection method and use it to develop a head tracking algorithm which consists of four modes: the entrance-detection mode, the tracking mode, the fixation mode and the disappearance mode.

[†] Address for correspondence: Institute of Information Science, Academia Sinica, Nankang, Taipei, 115 Taiwan.

International Journal of Pattern Recognition and Artificial Intelligence, Vol. 14, No. 2 (2000) 137–166

Many papers assumed that, at the very beginning, there has already been a human head (body) that can be seen in the image.[1,2,8,9,13] However, a practical tracking algorithm should have the capability to detect the entrance of a target. Vieren *et al.*[31] used the snake to monitor the entrance of an object, and Rehg *et al.*[23] used the motion information to detect the arrival of a human being. In our work, difference images are used for monitoring the entrance of human heads, but are not used for head detection or tracking. The reason is as follows. Image differencing is a commonly-used and effective technique for detecting image change, and is thus useful for motion detection, motion estimation and visual tracking.[5,14,24] However, for tracking human heads, we should not assume that the human head we are tracking is always moving. If the human head stops moving while some other objects are still moving, those methods using only the difference images may fail. Moreover, it may not work when the cameras are allowed to move. Therefore, difference images are only used to monitor the entrance of human heads in our head tracking algorithm.

Face detection[32] is an important research topic for its usefulness in many different applications. The approaches to face detection include the neural network-based approach,[25,28] the feature-based approach,[3,10,34,35] the color-based approach,[17,19,22,26,29] the motion-based approach[17,35] and the hybrid approach. Neural networks are widely used in face detection. Based on neural networks, Rowley *et al.*[25] have presented some face detection systems, among which a fast version can process an image of size $320 \times 240$ within two to four seconds on 200 MHz R4400 SGI Indigo 2. Another interesting face detection system using neural networks was reported by Sung and Poggio.[28] Nevertheless, the computations required by the above two good works are still very expensive in both the training and testing procedures. Some researchers have adopted the feature-based approach to solve the face detection problem. Yow and Cipolla[34] used the perceptual grouping framework to group feature points into face candidates, and then used a probabilistic framework to reinforce probabilities and to evaluate the likelihood of the candidate as a face. They further extended their work to enhance human face detection by using motion and active contours.[35] Burl *et al.*[3] combined a set of local feature detectors with a statistical description of the spatial arrangement of the features to localize quasi-frontal views of faces. The major disadvantage of the feature-based approach is that it cannot deal with relatively small faces, even though it has the ability to detect faces under different scales, orientations and viewpoints. In spite of the large number of face detection methods reported in the literature, face detection is still considered to be a difficult problem, due to the unknown face orientation, the unknown size of a face image, varying illumination, complex background, makeup and glasses in a face, etc. In this paper, we propose a new head detection method which is simple, efficient and very reliable.

Once a human head is detected and located in one frame of a video sequence, it should be tracked in the subsequent frames. The reason for using a head tracking algorithm instead of directly applying head detection to each frame is mainly that the tracking algorithm can fully utilize the inter-frame correlation to improve the algorithm performance, both in speed and reliability. Azarbayejani *et al.*[1] developed a head tracking system for visually controlled graphics. They used an extended Kalman filter formulation to recover six rigid-body motion parameters of an object from a small set of tracked visual feature points. Gee and Cipolla[9] proposed a model-based tracking algorithm for face tracking by using temporal consensus. They used the darkest pixels within local search windows for face tracking. McKenna *et al.*[19,22] used color mixture model for face tracking. Fieguth and Terzopoulos[8] developed a simple and fast head tracker by using five color regions for head tracking and one color region for torso tracking. Yang and Waibel[33] presented another face tracker. They characterized human faces by using a skin-color model, predicted search window by using a motion model, and predicted and compensated for camera motion by using a camera model. Basu *et al.*[2] developed a model-based head tracking system by using motion regularization. They used a 3D ellipsoidal model of the head and interpreted the optical flow in terms of the possible rigid motions of the model. The only requirement is to ensure that each sequence began with a near-frontal view.

Several researchers have modeled the shape of the image of a human head with an ellipse, and then used ellipse fitting methods to locate the human head.[13,26,29] However, it is not easy to reliably extract the contour of a human head when the background is complex. To solve this problem, we have developed a new ML head detection algorithm which utilizes a simple elliptical model and a hierarchical structure to search for the head reliably and efficiently. There have been a few interesting works on ML face detection. Moghaddam and Pentland[20] adopted a ML estimator which used an eigenspace decomposition method to detect human faces and hands. They selected a training set of facial feature templates and estimated the likelihood function. Kervrann *et al.*[15] also used the eigenspace decomposition method to detect human faces and extract the mouth features. They used the generalized likelihood ratio for face detection and ML for estimating the position of the mouth. However, the computation of the eigenspace decomposition and of the likelihood distance measure used are quite expensive. Colmenarez and Huang[6] proposed another ML face detector. They presented a visual learning approach that used nonparametric probability functions and entropy analysis to build a probability model. Then, the detection of face was carried out by searching with the model over several scaled versions of the input image. In this paper, we propose a simpler ML head detector by modeling the human heads with an elliptical template. Comparing with the above works, our ML head detector is simple, fast and reliable.

Another important feature of our head tracker is the use of stereo vision for human tracking. We are not claiming that stereo vision or 3D inference is a necessity for head detection and tracking, but that it helps because no single vision system is perfect by itself. Notice that the 3D structure of a human head can be approximately regarded as a rigid body while its 2D image can be deformed or warped due to head rotation and translation. Once the cameras of a stereo vision system are calibrated, the 3D structure of the observed human head can be easily obtained by using stereo correspondences of facial features (e.g. eyes, nose and mouth). Both the 3D structure and position of the human head can be useful for the prediction of the human head in the next frame. Furthermore, the stereo constraint can also be used for verification in head detection and tracking. Good examples are the works by Rehg *et al.*[23] and Darrell *et al.*[7] In the work of Rehg *et al.*, the vision task requirements for a kiosk interface is characterized. Stereo triangulation, using the image positions detected by the two calibrated cameras, was used to locate the user in the 3D space. Based on stereo information, they divided the distance from the kiosk into three categories for communication purpose. In the other work, Darrell *et al.* applied the census algorithm[7] to obtain the dense depth data (from stereo). Based on the depth data, they can isolate the figure of a user from other objects and people in the background.

Basically, the results obtained by the proposed ellipse-based head detector can only be treated as head candidates, because there can be occasionally ellipse-like objects in the scenes. However, in ordinary scenes, an ellipse-like object is usually indeed a human head if it happens to have roughly the size of a human head. Since we are using a stereo vision system, the size of the detected ellipse-like object may be easily obtained for verification. If more reliable result is required, further verification for eliminating false head candidates may become essential. Existing feature-based methods, which use facial contour and other facial features, such as eyes, mouth and nostrils, can be adopted for head verification. For example, Yow and Cipolla[35] used face boundary and motion consistency to verify the detected face candidates. Han *et al.*[10] proposed a coarse-to-fine process combining neural networks and principle component analysis to verify face candidates.

This paper is organized as follows. Section 2 describes the ML formulation and the method used for detecting the human head. Section 3 describes the details of our head tracking algorithm. Experimental results on detecting and tracking human heads are shown in Sec. 4. Conclusions are given in Sec. 5.

## 2. ML DETECTION OF HUMAN HEADS

An ellipse can be described by the following equation:

$$\frac{(x-x_0)^2}{S_x^2} + \frac{(y-y_0)^2}{S_y^2} = 1\,, \tag{1}$$

where $(x_0, y_0)^T$ is the center of the ellipse. Let $\boldsymbol{r}_0 = (x_0, y_0)^T$ and $s \equiv S_x = \varrho S_y$, where $\varrho$ is a pre-determined constant (see Sec. 4). Then, $\boldsymbol{r}_0$ and $s$ are the parameters that describe the ellipse. Consider Fig. 1(a). Let $\boldsymbol{v}_s^i$, $i = 1, 2, \ldots, N$, denote the points located uniformly on an ellipse of size $s$ centered at $\boldsymbol{r}_0$. Let $\boldsymbol{u}_s^i = \boldsymbol{v}_s^i - \boldsymbol{r}_0$. Then, $\boldsymbol{u}_s^i$ represents the displacement of $\boldsymbol{v}_s^i$ from the reference point $\boldsymbol{r}_0$. In this paper, we define the elliptical template for modeling the human head to be the following:

$$\boldsymbol{T}_{\boldsymbol{r}_0,S}(\boldsymbol{r}) = \sum_{i=1}^{N} \boldsymbol{h}_i \delta(\boldsymbol{r} - \boldsymbol{u}_S^i - \boldsymbol{r}_0)\,, \tag{2}$$

where $\boldsymbol{r} = (x, y)^T$ are the image coordinates, $\delta(\cdot)$ is the delta function, and $\boldsymbol{h}_i = [h_{xi}, h_{yi}]^T, i = 1, 2, \ldots, N$, are the weighting factors. One possible way of choosing the weighting factors is to let $\boldsymbol{h}_i = [1, 1]^T$ for all $i$. A better choice of $\boldsymbol{h}_i, i = 1, 2, \ldots, N$, is described in Sec. 2.1. Here, the elliptical template shown in Fig. 1(a) has the width of one pixel, and is referred to as a *one-pixel-width* elliptical template. By straightforward extension, an elliptical template having $k$ pixels width is referred to as a *$k$-pixel-width* elliptical template (for symmetry, $k$ is an odd number) as shown in Fig. 1(b).

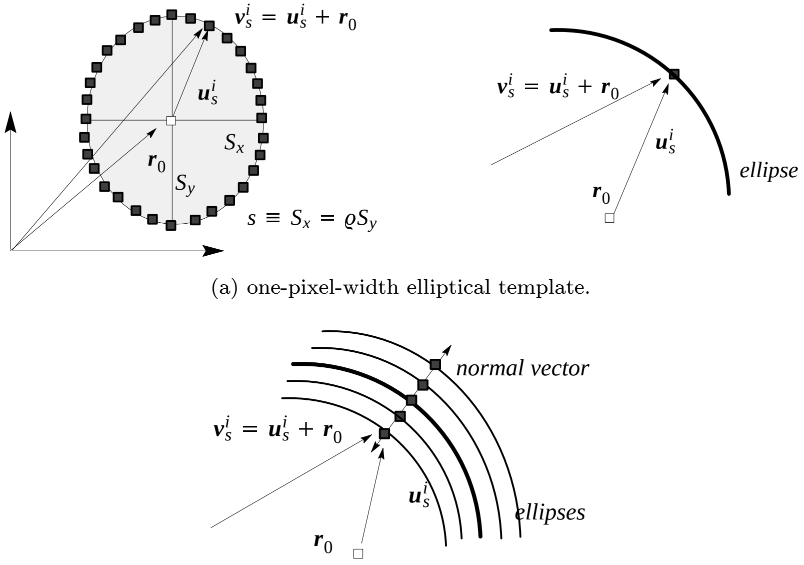
Based on the elliptical template, we formulate the human head detection as a Maximum Likelihood (ML) estimation problem.[16] Let $\boldsymbol{g}$ be an image of intensity gradients defined as the vector function $\boldsymbol{g} : \boldsymbol{D} \to \boldsymbol{R}^2$, where $\boldsymbol{D} = \{(x, y) : x, y = 1, \ldots, M\}$ and $\boldsymbol{R}^2$ is the set of all possible intensity gradient vectors. Let $\boldsymbol{G} = \{\boldsymbol{g}(\boldsymbol{r}), \boldsymbol{r} \in \boldsymbol{D}\}$. Given $\boldsymbol{r}_0$ and $s$, we assume that a noisy gradient image $\{\boldsymbol{g}(\boldsymbol{r}), \boldsymbol{r} \in \boldsymbol{D}\}$ containing an elliptical contour can be modeled as follows:

$$\boldsymbol{g}(\boldsymbol{r})|\boldsymbol{r}_0, s = \boldsymbol{T}_{\boldsymbol{r}_{0'S}}(\boldsymbol{r}) + \eta(\boldsymbol{r}) \,, \tag{3}$$

where $\boldsymbol{T}_{\boldsymbol{r}_0,s}(\boldsymbol{r})$ is defined in Eq. (2), and the $2 \times 1$ noise vector $\eta(\boldsymbol{r})$ is assumed to be Gaussian, i.e. $\eta(\boldsymbol{r}) \sim N(0, \sigma_\eta^2 \boldsymbol{I}_{2\times 2})$. Therefore, the joint probability density function $p(\boldsymbol{G}|\boldsymbol{r}_{0'}s)$ is also Gaussian, and can be shown to be

$$p(\boldsymbol{G}|\boldsymbol{r}_0, s) = \frac{1}{C} \exp\left\{ \frac{2\sum_{i=1}^{N} \boldsymbol{h}_i^T \boldsymbol{g}(\boldsymbol{u}_S^i + \boldsymbol{r}_0) - \boldsymbol{G}^T\boldsymbol{G} - \text{const}}{2\sigma_\eta^2} \right\} , \tag{4}$$

where $C = (2\pi\sigma_\eta^2)^{N/2}$. The ML estimate of the head position, $\boldsymbol{r}_0$, and size, $s$, can be obtained by maximizing $p(\boldsymbol{G}|\boldsymbol{r}_0, s)$ with respect to $\boldsymbol{r}_0, s$, which is equivalent to



(a) one-pixel-width elliptical template.

(b) $k$-pixel-width elliptical template (in this example, $k = 5$).

Fig. 1. The elliptical template used for modeling human heads (each square on the ellipse represents an image point).

maximizing $F(\boldsymbol{r}_0, s)$ with respect to $\boldsymbol{r}_0$ and $s$, where

$$F(\boldsymbol{r}_0, s) \equiv \sum_{i=1}^{N} \boldsymbol{h}_i^T \boldsymbol{g}(\boldsymbol{u}_s^i + \boldsymbol{r}_0)\,. \tag{5}$$

In fact, the operation required by Eq. (5) is similar to that required by template matching where the template$\{\boldsymbol{h}_i\}$ is described in Sec. 2.1.

### 2.1. The Width and the Weighting Factors of Elliptical Templates

In this paper, $\boldsymbol{G}$ is the gradient image which contains $[\boldsymbol{G}_x,\boldsymbol{G}_y]^T$, where $\boldsymbol{G}_x$ is the horizontal gradient image and $\boldsymbol{G}_y$ is the vertical gradient image. It is noted that the directions of edges in the top and bottom portions of the head contour tend to be horizontal and those in the left and right portions tend to be vertical.

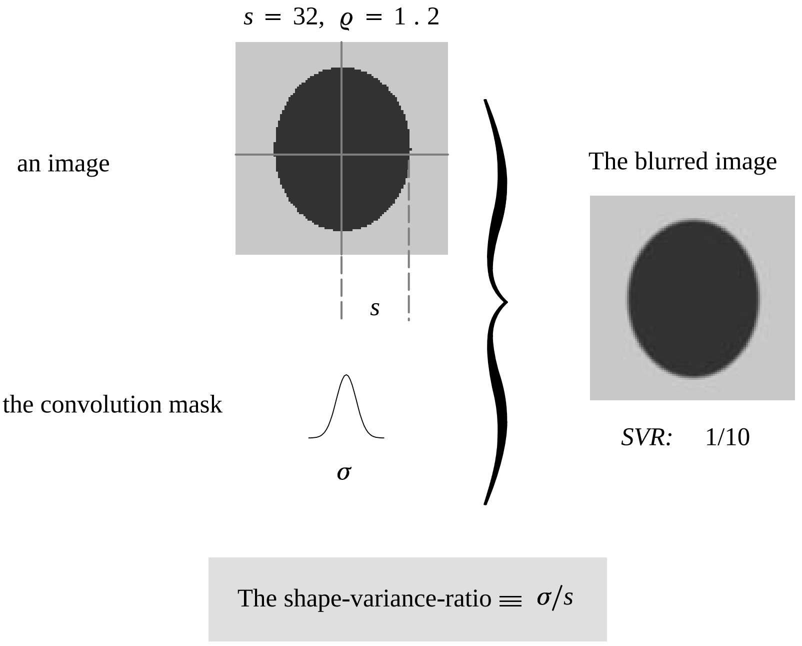
To design an appropriate elliptic template $\{\boldsymbol{h}_i\}$ for human head detection, a straightforward method is to use a large database containing a great many images of human heads. The collection of the head contours of all these images is then served as a training set, and a suitable template can be learned through statistical parameter estimation, neural network or other learning strategies. However, such a strategy requires the head contours to be correctly segmented from each image, which is difficult to be done in an automatic and faultless way.

In our work, instead of learning a template from a large database, we design a procedure to determine the parameters of the template. The procedure is based on the following useful observation. In general, a human head is not an exact ellipse. Nevertheless, a human head can be treated as an ellipse with a little shape variation. Hence, if we want to detect the contour of a human head with an elliptic template, the width of the boundary of the ellipse has to be large enough to cover the shape variation of the contours of human heads. That is, a *k-pixel-width* elliptical template with an appropriate value of $k$ should be used. However, using a *k-pixel-width* elliptical template for matching will take much more computation time for larger $k$, and thus is not an efficient way for implementation. Fortunately, if we subsample the image roughly $k \times k$ (i.e. a $k \times k$ block in the image is subsampled to be a pixel) and obtain a low-resolution image, then the shape variation of the contour of the human head will be reduced to approximately one pixel. Therefore, the contour of the human head can be detected using a *one-pixel-width* elliptical template in the low-resolution image, and hence can be implemented much more efficiently. This important observation inspires us to perform the template matching in a coarse-to-fine manner based on an image pyramid structure (which will be introduced in the next section). From this observation, we can also realize that if the size of a human head contained in an image is small, then a template with thin boundary (i.e. a *k-pixel-width* elliptical template where $k$ is small) is suitable for head detection; on the other hand, if the size of a human head is large, a template with thick boundary should be used. Following this principle, we performed the procedure described below to find appropriate templates.

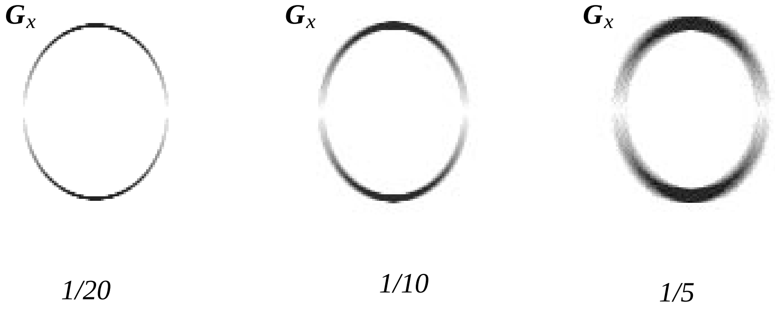
We first generated an image, $\boldsymbol{I}_e$, which contains a black ellipse and a white background. Then, $\boldsymbol{I}_e$ is blurred with a Gaussian convolution mask whose variance is $\sigma^2$. The Gaussian convolution mask was used to simulate the shape variation of the contour of a human head. In principle, a larger $\sigma$ was used to simulate larger shape variation, and vice versa. Also, from the observation mentioned above, the shape variation is proportional to the size of the human head in an image. Therefore, for the simulation purpose, we wanted to use a larger $\sigma$ in the Gaussian blurring process for a larger human head and a smaller $\sigma$ for a small human head, respectively. Instead of repeating the blurring process for many different images containing all of the possible sizes of human heads, we used a hierarchical procedure described above. Here, we denote the shape-variance-ratio (SVR) to be the ratio of to $\sigma$ to $s$.

(1) Set $s = s_{\max}$, the largest possible size of human heads to be handled in our algorithm.
(2) Generate an image $\boldsymbol{I}_e$ which contains a white background and a black ellipse whose $S_x = s$ and $S_y = \varrho s$.
(3) Set the value of the shape-variance-ratio, $SVR$. Usually, we set $SVR = 1/20, 1/10$, or $1/5$. The value of $SVR$ reflects the shape variance relative to the size of a human head.
(4) Blur the image $\boldsymbol{I}_e$ with the Gaussian convolution mask whose $\sigma$ is $SVR \times s$, and obtain a new image $\boldsymbol{I}'_e$.
(5) Use the Sobel operator to compute the gradient images of $\boldsymbol{I}'_e$. Assume that gradient images $\boldsymbol{G}^I_x$ and $\boldsymbol{G}^I_y$ are obtained, where $\boldsymbol{G}^I_x$ is the gradient image in the horizontal direction, and $\boldsymbol{G}^I_y$ is the gradient image in the vertical direction, respectively.
(6) Design the template $\{\boldsymbol{h}^s_i\}$ used for detecting the head with size $s$ to be $\{h^s_{xi} | i = 1, 2, \ldots, N\} = \boldsymbol{G}^I_x$ and $\{h^s_{yi} | i = 1, 2, \ldots, N\} = \boldsymbol{G}^I_y$.
(7) Select a set of subsampled ratios $a_1, a_2, \ldots, a_j, \ldots$, where $0 < a_j < 1$ for $j = 1, 2, \ldots, M$. Let $s_i = a_i \times s$, for $j = 1, 2, \ldots, M$.
(8) Subsample $\boldsymbol{G}^I_x$ and $\boldsymbol{G}^I_y$ using each subsampled ratio $a_j, j = 1, 2, \ldots, M$, and obtain a set of gradient images $\boldsymbol{G}^I_{x_j}$ and $\boldsymbol{G}^I_{y_j}$, where $j = 1, 2, \ldots, M$.
(9) Design the templates $\{\boldsymbol{h}^{S_j}_i\}$ used for detecting the human head with size $s_j$ to be $\{h^{S_j}_{xi} | i = 1, 2, \ldots, M\} = \boldsymbol{G}^I_{x_j}$ and $\{h^{S_j}_{yi} | i = 1, 2, \ldots, M\} = \boldsymbol{G}^I_{y_j}$, respectively.

Using the above procedure, a set of appropriate templates which is useful for detecting the contours of human heads from small sizes to large sizes can be generated. The values of $S_{\max}$ and $SVR$ can be selected according to the requirements of applications. To be clear, an example where $S_{\max}$ is set to be 32 in a $512 \times 512$ image is shown in Figs. 2 and 3. Figure 2(a) shows an image containing a black ellipse with $s = 32$ and $\varrho = 1.2$, and the image was blurred with $\sigma = SVR \times 32$ where $SVR = 1/10$. After computing the gradient images, the $\boldsymbol{G}^I_x$ obtained using some different $SVR$ values are shown in Fig. 2(b). Figure 3(b) shows an example of the subsampling process of the above procedure to obtain the templates for detecting some smaller human heads. Finally, we enlarge the sizes of two smaller templates to be $s = 32$, and show those three templates having the same sizes as in Fig. 3(c).



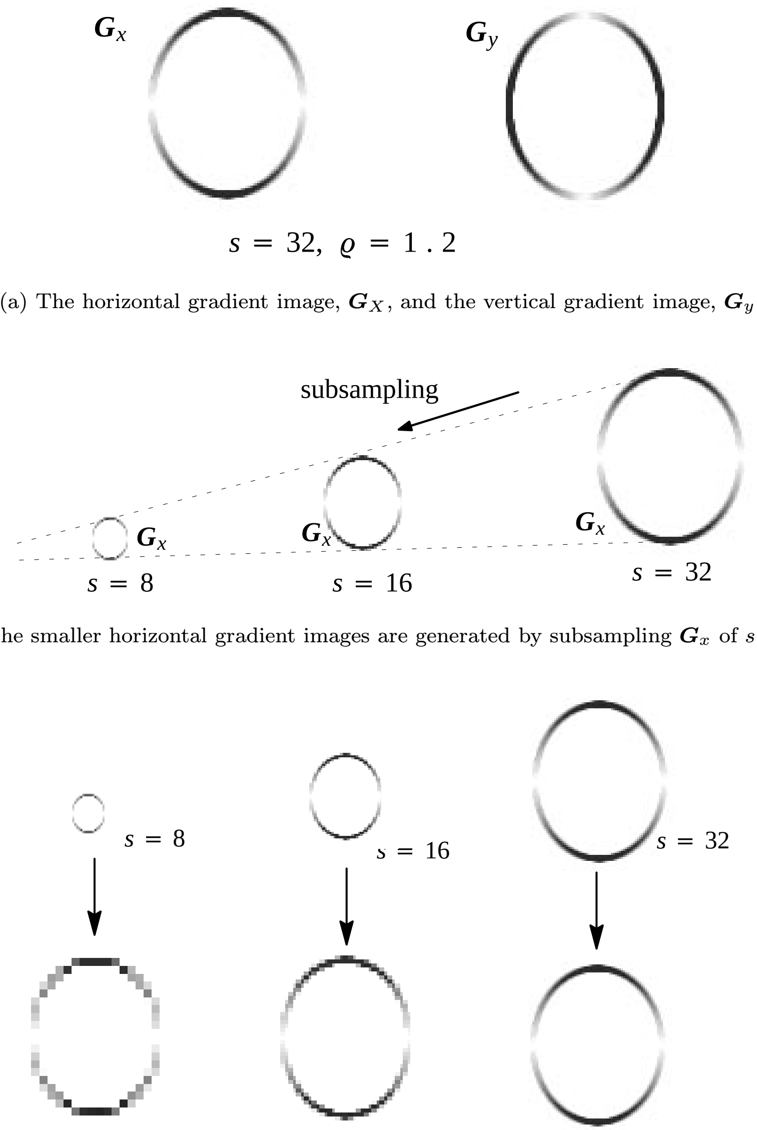
(a) An example for describing $SVR$.



(b) The horizontal gradient images with different $SVR$s.

Fig. 2. The shape-variance-ratio ($SVR$).

The weights of the designed templates are usually real numbers. However, template matching using real numbers includes the computation of both multiplication and addition, which is not efficient in practice. In our implementation, we simply set the weights in the template $\{\boldsymbol{h}_i\}$ to be 1 if the weight value is larger than a given threshold, and otherwise to be 0. Hence, only addition is required in our implementation, and thus is much more efficient. According to our experience, such a simplification of the implementation procedure would not affect the results of human head detection in most cases.



(a) The horizontal gradient image, $\boldsymbol{G}_X$, and the vertical gradient image, $\boldsymbol{G}_y$.

(b) The smaller horizontal gradient images are generated by subsampling $\boldsymbol{G}_x$ of $s = 32$.

(c) The smaller horizontal gradient images are enlarged to the same size for comparison.

Fig. 3. Determining the width of elliptical templates using subsampling.

## 2.2. ML Head Detector: Two-Channel and Multiscale

Our head detection method is executed in an image pyramid structure. That is, a set of images with different resolutions will be generated by smoothing and subsampling the input image. The image of each resolution will be used for human head detection via template matching. Here, we call it the *basic head detector* that the subroutine used for human head detection of a single image in the image pyramid. In the following, we will introduce the basic head detector at first, and then introduce the integration of the basic head detectors into an image pyramid structure for increasing the matching efficiency.

The basic head detector consists of two channels: the horizontal channel and the vertical channel. The gradient images generated by the Sobel operator is decomposed into two components $\boldsymbol{G}_x$ and $\boldsymbol{G}_y$, and are used by the horizontal and the vertical channels, respectively. Each channel is implemented by multiple template matching processes using a set of templates with different sizes. Here, we use $\mathbf{H}$ to denote the set of templates used in the multiple matching processes. Then, the matching scores corresponding to the same size of the elliptical template in the two channels are summed. Finally, the position and the size with the highest score via template matching is served as the result of the head detection. Figure 4 shows an example of the basic head detector. In addition, we can also select a class of the highest-score templates to be the potential candidates (e.g. 50 highest-score templates as shown in Fig. 4) and then use further verification steps (e.g. the steps introduced in Secs. 3 and 4) to find better results.

We hope that the head detection method can detect human heads with variant ranges. Assume that the possible width of a human head required to be handled may vary from $c_{\min}$ to $c_{\max}$ in a 512 × 512 image, where $1 < c_{\min} < c_{\max} < 512/\varrho$. Without lost of generality, we will use $c_{\min} = 12$ and $c_{\max} = 360$ for the following discussion. In fact, the range covered by $c_{\min} = 12$ and $c_{\max} = 360$ is large enough for most practical situations in the human head tracking algorithm. Based on the definition of the size factors introduced in Eq. (1), the range of the ellipses required to be detected is from $s = 6$ to $s = 160$. However, it is very time-consuming to perform multiple template matching processes for all of the necessary values between $s = 6$ to $s = 160$ via the basic head detector. Fortunately, we can integrate the basic head detector into an image pyramid structure to avoid such an exhaustive matching process. We generate an image pyramid containing five levels of images (level 0: 512 × 512, level 1: 256 × 256, level 2: 128 × 128, level 3: 64 × 64 and level 4: 32 × 32). For each image in the pyramid, we only use a set of templates ranging from $s = 6$ to $s = 10$ for matching via the basic head detector. Notice that using a template of sizes for human head detection in the low-resolution image of level $i$ is equivalent to performing a coarse template matching with size $s \times 2^i$ in the highest-resolution image (i.e. the level 0 image). For example, the template matching using $s = 6, 7, 8, 9, 10$ in the 32 × 32 image can roughly handle the matching process of using $s = 96, 112, 128, 144, 160$ in the 512 × 512 image, respectively. Figure 5(a) gives an explanation of the sizes which can be covered by this hierarchical matching process. Figure 5(b) shows the spectrum of the sizes of the ellipses that can be detected. Although there are some empty ranges in the spectrum, it will not affect the matching results for most cases as explained in the following. Remember that the larger the size, the larger is the degree of blur of the elliptical templates designed in our work. Also notice that the width of the empty ranges in the spectrum is increasing with respect to the size, as shown in Fig. 5(b). Hence, the empty sites can still be covered by the blurred region of the templates used for some nonempty sites in the matching process.

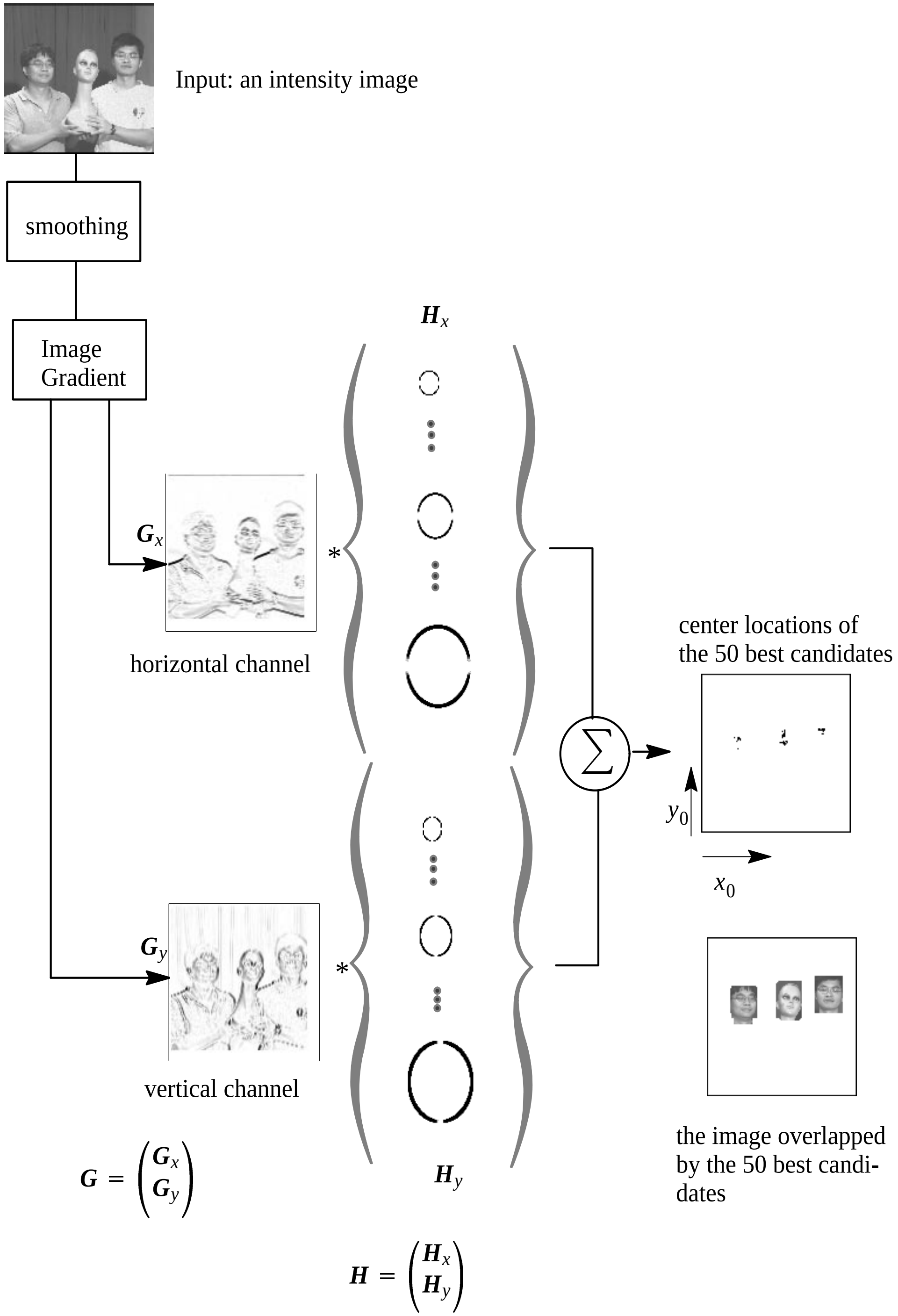
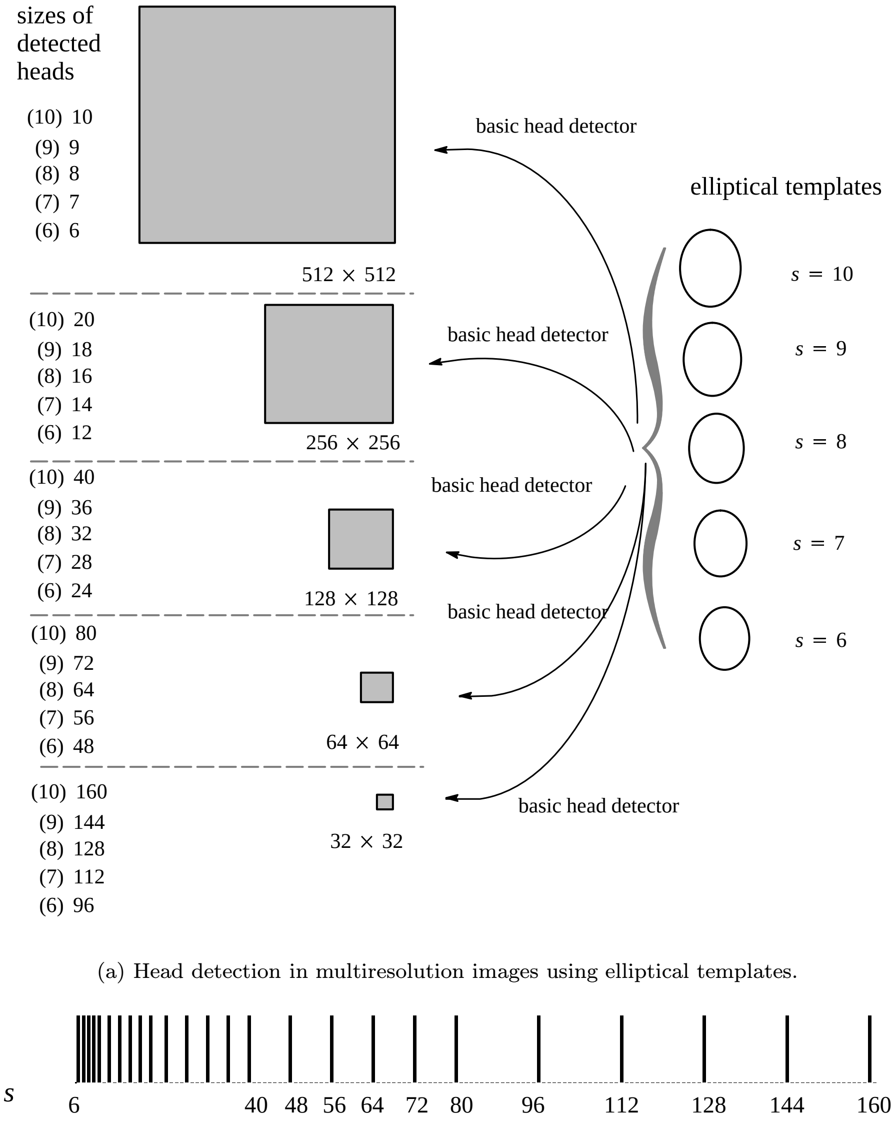

Input: an intensity image
smoothing
Image Gradient
$H_x$
$G_x$
*
horizontal channel
$G_y$
*
vertical channel
$G = \begin{pmatrix} G_x \\ G_y \end{pmatrix}$
$H_y$
$H = \begin{pmatrix} H_x \\ H_y \end{pmatrix}$
$\sum$
center locations of the 50 best candidates
$y_0$
$x_0$
the image overlapped by the 50 best candidates

Fig. 4. The two-channel and multiscale method for implementation of the ML head detector.



(a) Head detection in multiresolution images using elliptical templates.

(b) The spectrum of sizes of the ellipses that can be detected in example 5(a).

Fig. 5. The general framework for detecting human heads using the ML head detector.

Using the above hierarchical method, only five template matching processes are required for each of the five level images. Therefore, totally $5 \times 5 = 25$ times of template matching is necessary to detect a human head with possible widths from 12 to 320 pixels. Moreover, because the templates used here are small ones (the maximal value of $s$ is only 10), the width of the boundary of the ellipse is usually small (or even is only one pixel). Matching with thin-boundary elliptical templates allows the matching process to be more efficient. Because there is some limitation on the distance from the human head to the camera set, the position and the size of the human head in the image are limited to fall within a small search range (e.g. from $s = 48$ to $s = 80$). A limitation of the above hierarchical matching procedure is that only a rough head position can be detected if it is detected by using a low-resolution image. Nevertheless, the precision is enough for some applications of the visual surveillance and human–computer interface. If a higher precision is required, one can use the following refinement process to find a more precise position in a coarse-to-fine manner.

Generally, in this paper, the hierarchical structure (image pyramid) plays two roles: one is for reduction of the computation time and the other is for refinement of the detection results of the general framework. First, we discuss the issue of reduction of the computation time. As mentioned above, if the size of a human head is not small (e.g. $48 \leq s \leq 80$) and is known in advance, the human head can be directly detected in the low-resolution (it is not necessary to detect the human head with all templates whose sizes vary from $s = 6$ to $s = 160$). And then use the hierarchical structure to find the size and position of this human head. Compared to detecting a human head using the basic head detector, the computation time of the basic head detector using the hierarchical structure can be greatly reduced. We give an example and show it in Fig. 6. There are four levels in this example. In level 3, we find the position and size of a human head, $\boldsymbol{r}_0$ and $s$, within a large search region (e.g. $50 \times 50$) in the $64 \times 64$ image, and obtain $\hat{\boldsymbol{r}}_0(1)$ and $\hat{s}(1)$. In level 2, the search region centered at $2\hat{\boldsymbol{r}}_0(1)$ is set to be $3 \times 3$, and the search region centered at $2\hat{s}_0(1)$ for the elliptical size is set to be $3 \times 3 \cdot \hat{\boldsymbol{r}}_0(2)$ and $\hat{s}(2)$ are obtained. Because it only detects a very small region, instead of detecting human heads in the whole image, the computation cost is very low. Similarly, the results of level 2 are used for finding the human head in level 1, and also the results of level 1 are used for finding the human head, $\hat{\boldsymbol{r}}_0(4)$ and $\hat{s}(4)$ in level 0. $\hat{\boldsymbol{r}}_0(4)$ and $\hat{s}(4)$ are the final results in the $512 \times 512$ image. By using such structure, the computation time can be greatly reduced. Also, for the template matching operation of each level involved in the head detector we adopt the adaptive EJO technique[12] to further speed up the computation (see Sec. 4). Second, we discuss the issue of refinement. If the sizes of human heads in an image are large, the results using the general framework described above are not very accurate. For example, there are only five sizes of human heads to be detected in the range of $s = 90$ to $s = 160$. That is, if we use the general framework of head detector to find the human heads, we have to refine the detection results of large human heads. The refinement strategy is similar to that shown in Fig. 6. Because of adopting the hierarchical structure, it is very fast to refine the detection results obtained in the low-resolution image.

So far, the head detector is used to detect the human head in a single image. In our work, we apply the head detector not only in a still image but also in an image sequence. In the beginning of head tracking, the size and position of a human head are unknown. The search region in the first image should be set as a large search region to find the human head. However, when the tracker is tracking a human

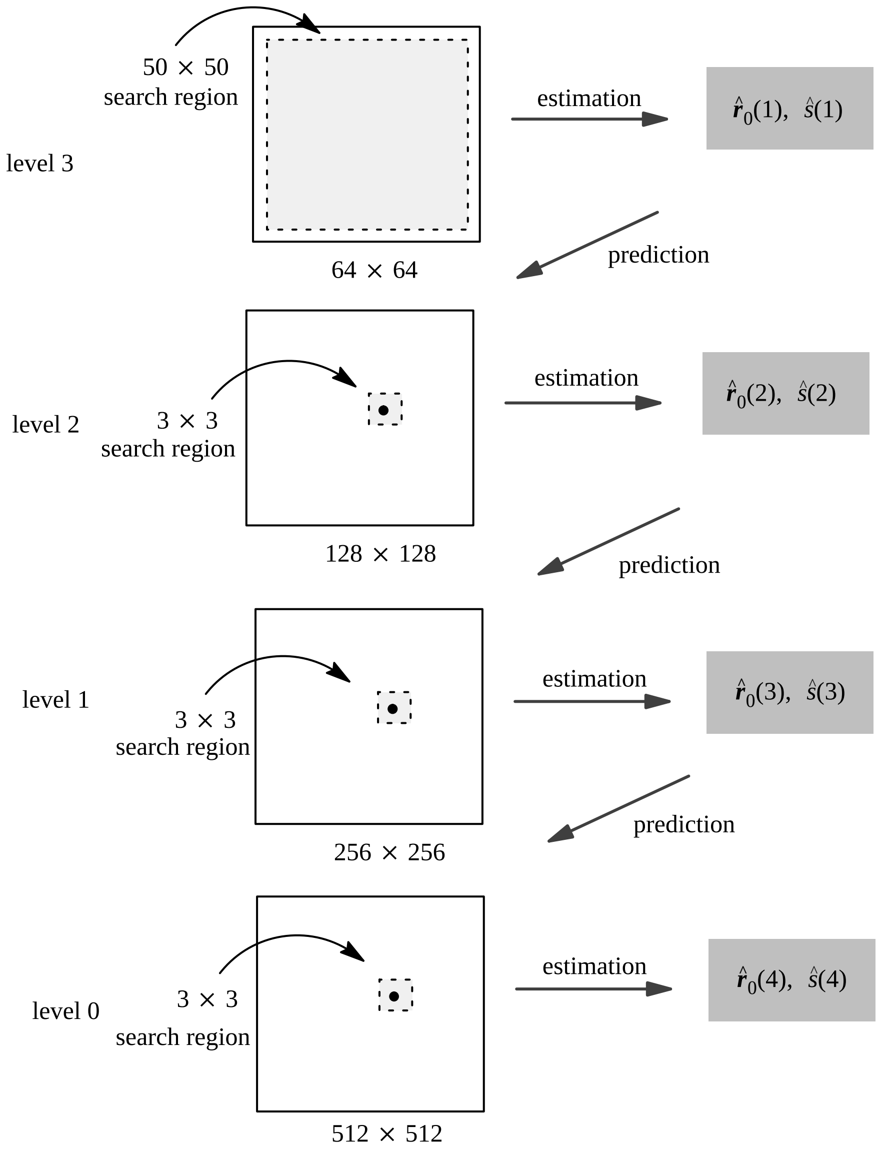


Fig. 6. Hierarchical structure is used for refining the detection results obtained in low-resolution and for reducing the search space in ML head detector. There are four levels in this example.

head, the position and size of the head found in the previous images can be used to predict the size of the elliptical template used for the next image.

## 3. THE HEAD TRACKING ALGORITHM

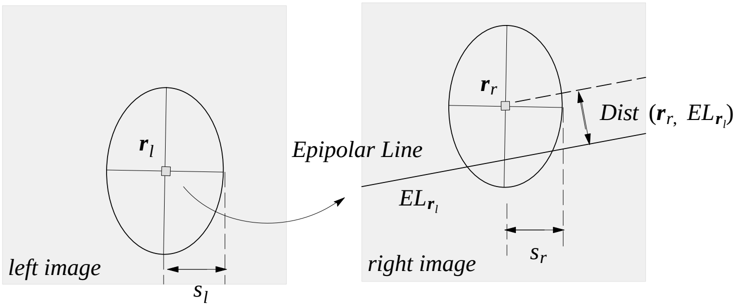
In this work, the head tracker is built upon an active stereo vision system which has been calibrated so that all the camera parameters and kinematic parameters are available. The advantages of using such a well-calibrated active stereo vision system for head tracking include the following. First, it can provide high-resolution images for face detection, reconstruction and recognition, by using a telescopic lens (i.e. a lens having a long focal length) or a zoom lens, while it still allows the person who we want to track to move around in a wide area. Second, it can use the epipolar geometry (or more precisely, the mutually-supported constraint described in this section) to improve the correctness of head detection. Third, it can simplify the fixation process by providing all the camera parameters and the kinematic parameters needed by the computation of inverse kinematics.

The head tracking algorithm presented in this paper has four modes: the entrance-detection mode, the tracking mode, the fixation mode and the disappearance mode. The entrance-detection mode is used to determine whether a human head is entering into the field of view, so that the head tracker can be completely automatic (i.e. without manual initialization). For this purpose, we use the difference image between the background image (containing no human head) and the present image. For each pixel, if the absolute difference is larger than a threshold, we count one. Let $N_D$ be the number of pixels whose absolute difference values are larger than a threshold. If $N_D$ is larger than a prespecified threshold, which means a target is moving into the field of view, then the tracker will switch to the tracking mode.

In the tracking mode, we use the ellipse-based ML head detector described in the last section to locate the human heads seen in the images. However, because of noise and the modeling error (e.g. the contour of a human head is usually not an exact ellipse and the noise is usually not Gaussian), the estimate of the head position obtained by using the ML detector may occasionally be incorrect. To improve the correctness of head detection, instead of using only the most likely one, we first keep a few candidates of human heads when maximizing Eq. (5) and then adopt a mutually-supported constraint to eliminate those incorrect candidates. The mutually-supported constraint used in our head tracking algorithm is briefly described below.

Consider a pair of stereo images shown in Fig. 7. Let $\boldsymbol{r}_l$ and $\boldsymbol{r}_r$ be the centers of the elliptical candidates found in the left and right images, respectively. Let $s_l$ and $s_r$ be their corresponding elliptical sizes. Since our active stereo vision system is well calibrated, given $\boldsymbol{r}_l$ in the left image, we can determine its corresponding epipolar line, $EL_{\boldsymbol{r}_l}$, in the right image. Similarly, $EL_{\boldsymbol{r}_r}$, the epipolar line in the left image corresponding to $\boldsymbol{r}_r$, can also be determined if $\boldsymbol{r}_r$ is given. Next, let $\mathrm{Dist}(\boldsymbol{r}_r, EL_{r_l})$ denote the 2D distance between the point $\boldsymbol{r}_r$ and the epipolar line $EL_{r_l}$ which can be easily computed if $\boldsymbol{r}_l$, $\boldsymbol{r}_r$ and the camera parameters are given. Similar case holds for $\mathrm{Dist}(\boldsymbol{r}_l, EL_{\boldsymbol{r}_r})$. In our head tracking algorithm, two elliptical candidates specified by $(\boldsymbol{r}_l, s_l)$ and $(\boldsymbol{r}_r, s_r)$ can only form a valid stereo correspondence if they satisfy the following mutually-supported constraint: $\mathrm{Dist}(\boldsymbol{r}_r, EL_{r_l}) < \mathrm{Threshold}_1, \mathrm{Dist}(\boldsymbol{r}_l, EL_{r_r}) < \mathrm{Threshold}_1$ and $|s_l - s_r| < \mathrm{Threshold}_2$, as illustrated in Fig. 7.

Once the best and valid stereo correspondence of the elliptical candidates is obtained, it can be used to infer an ellipsoid in the 3D space since the camera



IF $(\boldsymbol{r}_l, \boldsymbol{r}_r)$ *is a stereo pair,*
THEN *it should satisfy* :

$$Dist\ (\boldsymbol{r}_r,\ EL_{\boldsymbol{r}_l}) < threshold_1,$$
$$Dist\ (\boldsymbol{r}_l,\ EL_{\boldsymbol{r}_r}) < threshold_1,$$
$$|s_l - s_r| \leq threshold_2\ .$$

Fig. 7. An illustration of the mutually-supported constraint.

parameters are available in our algorithm. The center of this ellipsoid can be used to control the fixation of the active stereo cameras when the human head is moving toward the borders of the images. It can also be used to predict the 3D motion of the human head using Kalman filtering. This prediction is useful for reducing the search range of applying the ML head detector in the next image frame. Finally, if the human head disappears (due to tracking failure or out of surveillance range) for a few cycles, the tracking algorithm will first enter into the disappearance mode and then switch back to the entrance-detection mode.

The mode transitional diagram for our algorithm is shown in Fig. 8. The algorithm is summarized below:

*Entrance-Detection Mode* If $N_D >$ Threshold, then Goto *Tracking Mode.*

*Tracking Mode*

(1) Search for heads by using the ellipse-based ML detector in both images and then verify them with the mutually-supported constraint.

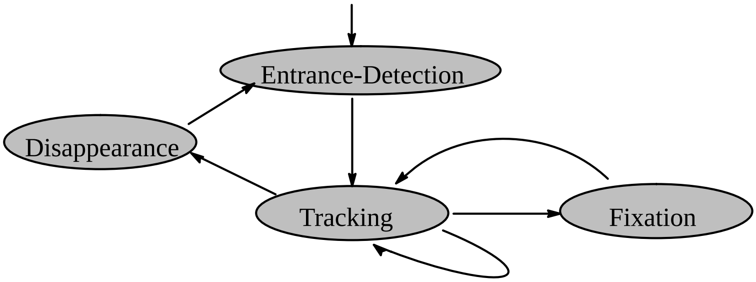


Fig. 8. The mode transitional diagram for the head tracker. There are four modes in this head tracker.

(2) If the human head disappears for a few cycles, then Goto *Disappearance Mode.*
(3) Use the centers of the ellipses to calculate a 3D position.
(4) Estimate the 3D motion parameters and predict the head motion.
(5) If the prediction goes out of the safety-margin, then Goto *Fixation Mode.*

*Fixation Mode*

(1) Control the stereo cameras to focus their gaze on the moving human head.
(2) Goto *Tracking Mode.*

*Disappearance Mode*

(1) Grab a new pair of stereo images which contain only the background.
(2) Goto *Entrance-Detection Mode.*

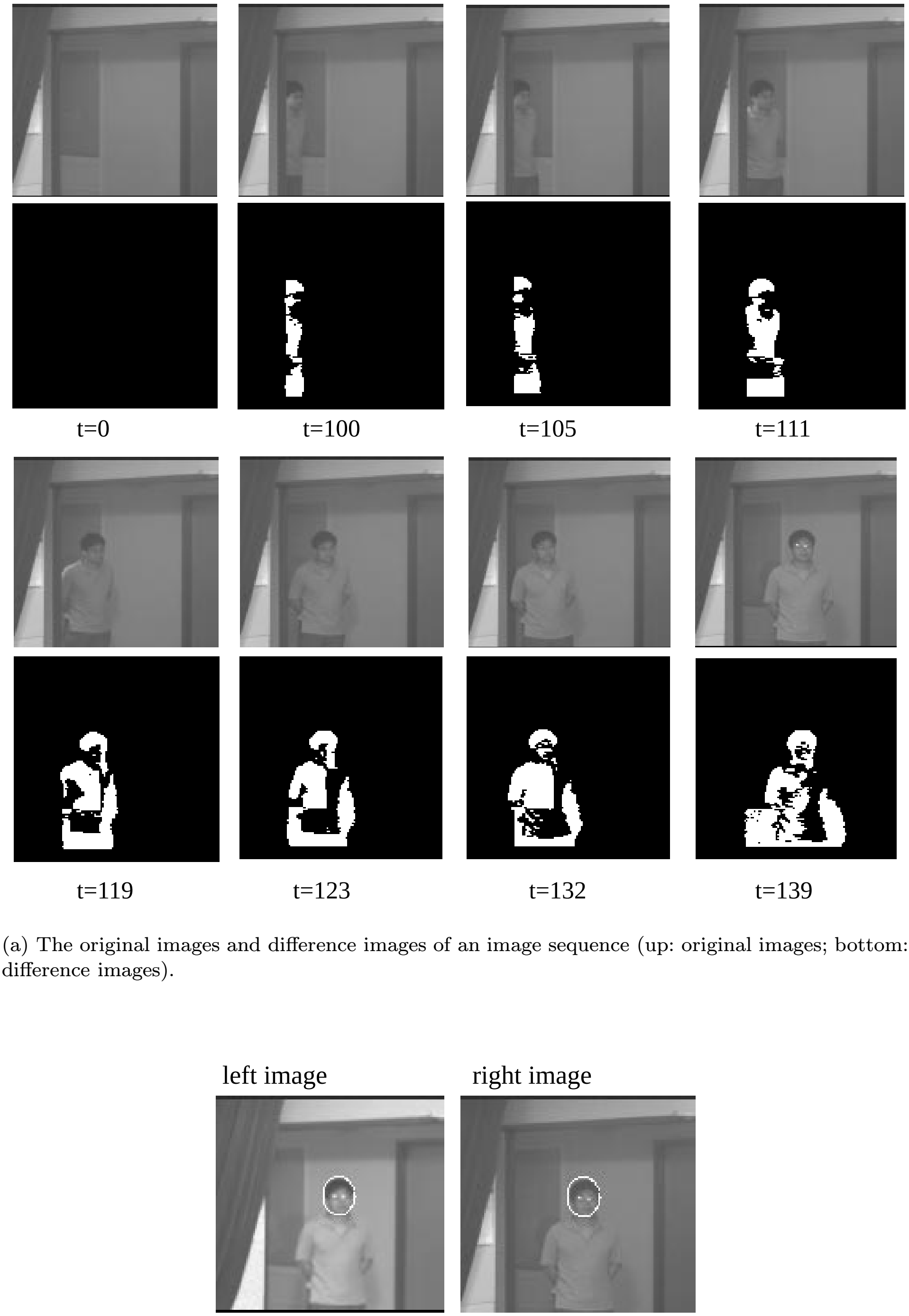
## 4. EXPERIMENTAL RESULTS

In the following experiments, the camera system we used for head tracking is the IIS head, a reconfigurable binocular head that employs eight motors to control its stereo cameras. The IIS head has been calibrated with a simple four-stage method and has achieved the accuracy of one pixel prediction error and 0.2 pixel epipolar error, even when all the eight motors are moving simultaneously.[27] Based on this calibration accuracy, our tracking algorithm not only can use the mutually-supported constraint to verify the candidates of heads found by the ellipse-based ML head detector, but can also easily use the inverse kinematics to control the binocular head to focus their gaze on the moving human head.

In the entrance-detection mode, since the camera is stationary, the difference image can be used to monitor the entrance of a human body. Note that we do not try to detect a human head in the entrance-detection mode. Figure 9 shows an image sequence (only the right image sequence is shown). From $t = 0$ through $t = 138$, $N_D$ is smaller than a prespecified threshold, this algorithm still stays in the entrance-detection mode. In $t = 139$, $N_D$ is larger than a prespecified threshold, then the algorithm switches to the tracking mode.

Next, we use an example to show the use of the mutually-supported constraint in eliminating incorrect candidates for head detection. Figure 10(a) shows that four candidates are found by the ellipse-based ML head detector for each image. It is obvious the most likely candidates (i.e. the #1 candidates) found in both images do not form a valid stereo correspondence. However, after the verification by the mutually-supported constraint, the #3 candidate in the left image and the #3 candidate in the right image are found to be the best stereo pair. Figure 10(b) shows the corresponding vertical and horizontal edge images used by the ellipse-based ML head detector.

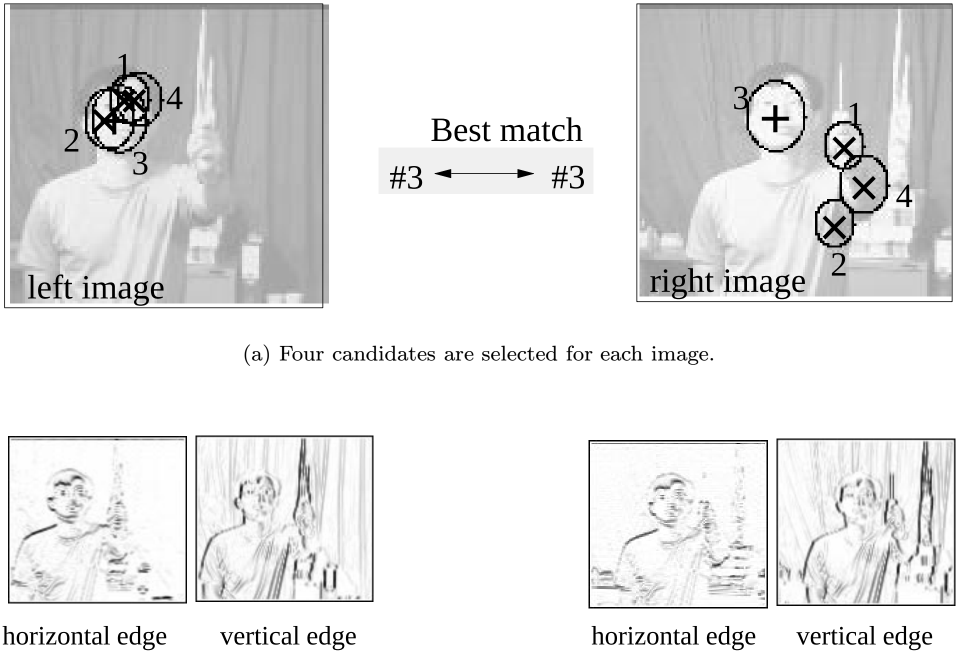
Figure 11 shows an example of applying our head tracking algorithm to detect and track a human head seen in an image sequence. Notice that the background of this image sequence changed most of the time because of the camera movement for fixation. Due to the camera movement, the difference images can hardly be used for detecting the human head in the tracking mode. However, our ellipse-based ML



(a) The original images and difference images of an image sequence (up: original images; bottom: difference images).

(b) The tracking results at $t = 139$.

Fig. 9. An image sequence by using the difference image in the entrance-detection mode.



(a) Four candidates are selected for each image.

(b) The horizontal and vertical edge images computed from the above two images are used by the ellipse-based ML head detector.

Fig. 10. An illustration of using the mutually-supported constraint to eliminate incorrect candidates found by the ML detector.

head detector, together with the mutually-supported constraint, works pretty well. In Fig. 11(c), the size of the head was larger than those in other images because the person moved closer to the camera system at that time instant. In Figs. 11(g) and 11(j), the person tracked purposely rotated his head to test the performance of the tracker. In Figs. 11(e) and 11(f), partial occlusion caused by the hand (or hands) did not prevent our head detector from working. In Figs. 11(h) and 11(i), another person walked into the field of view unexpectedly and passed through behind the person that the tracker was tracking, and the tracking algorithm can still keep track of the target. The other example with a different person and a different background is shown in Fig. 12. The panorama of the environment in this experiment is shown in Fig. 12(a). In Figs. 12(b) and 12(c), $N_D$ is larger than a prespecified threshold, then the algorithm switches to the tracking mode and the results are shown in Fig. 12(d). (The sizes of images we used in image sequences are $128 \times 128$. The image sequence shown in Figs. 11 and 12 and other image sequences containing their source images and corresponding tracking results are available at website ftp://smart.iis.sinic a.edu.tw/pub/HeadTracker.)

Finally, to demonstrate the performance of our head tracking algorithm, we show a few other examples of tracking different persons in different backgrounds. Figure 13 shows some experimental results where the elliptical templates can properly locate the human heads, which are regarded as successful detection. Sometimes, the

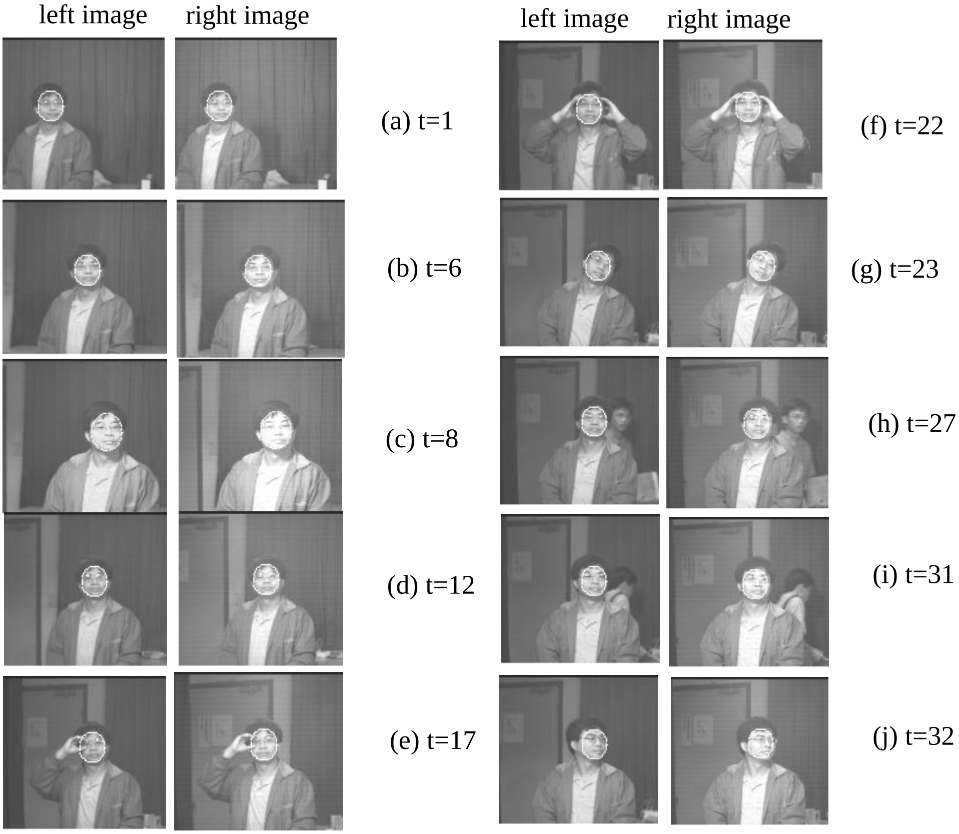


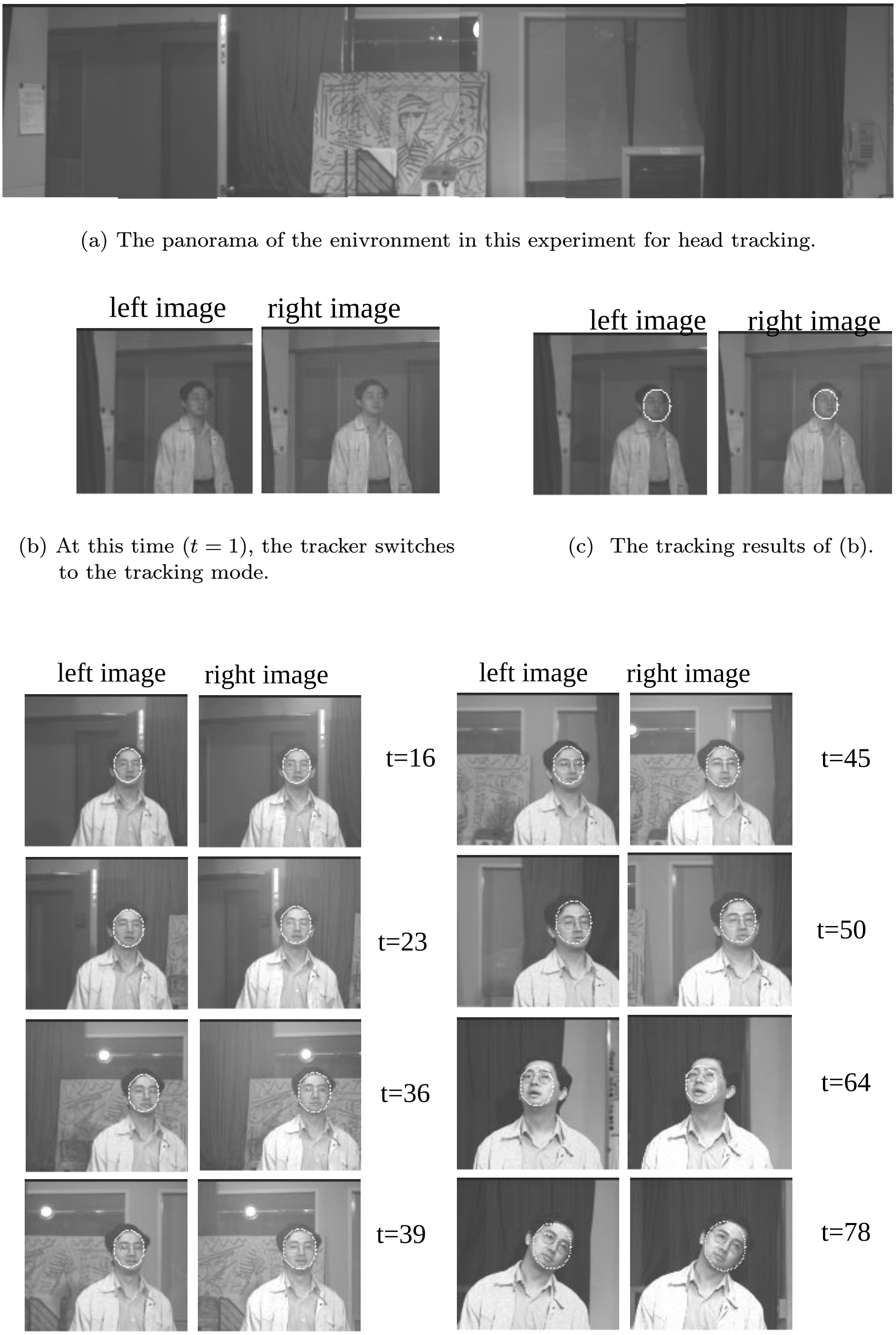
Fig. 11. An example of applying our algorithim to detect and track a human head.

head detection may not be very precise due to occlusion, shadowing and rotation, but the results of tracking may still be acceptable, as shown in Fig. 14. Of course, our tracking algorithm might occasionally fail to detect the correct human head for one or two frames. However, as long as the correct human head can be detected in the next few frames, the previous detection failure can be easily recovered.

Note that, according to our experimental results, our head detector is not sensitive to setting the constant $\varrho$ among $\varrho = 1.2, 1.2, 1.3$ and 1.4. Of course, $\varrho$ can be an estimated parameter in the formulation of our head detector, and then the size of search space is increased by three to four. However, detecting a human head in four-dimensional search space is more time-consuming than that in three-dimensional search space. In this paper, we set $\varrho = 1.2$ in our experiments in consideration of computation time.

### 4.1. The Advantages of our ML Head Detector

In a Sparc 20 workstation, the execution times without image acquisition to detect the human heads in a 512 × 512 image by using the basic head detector are shown in Fig. 15. The procedure of applying the hierarchical structure in the basic



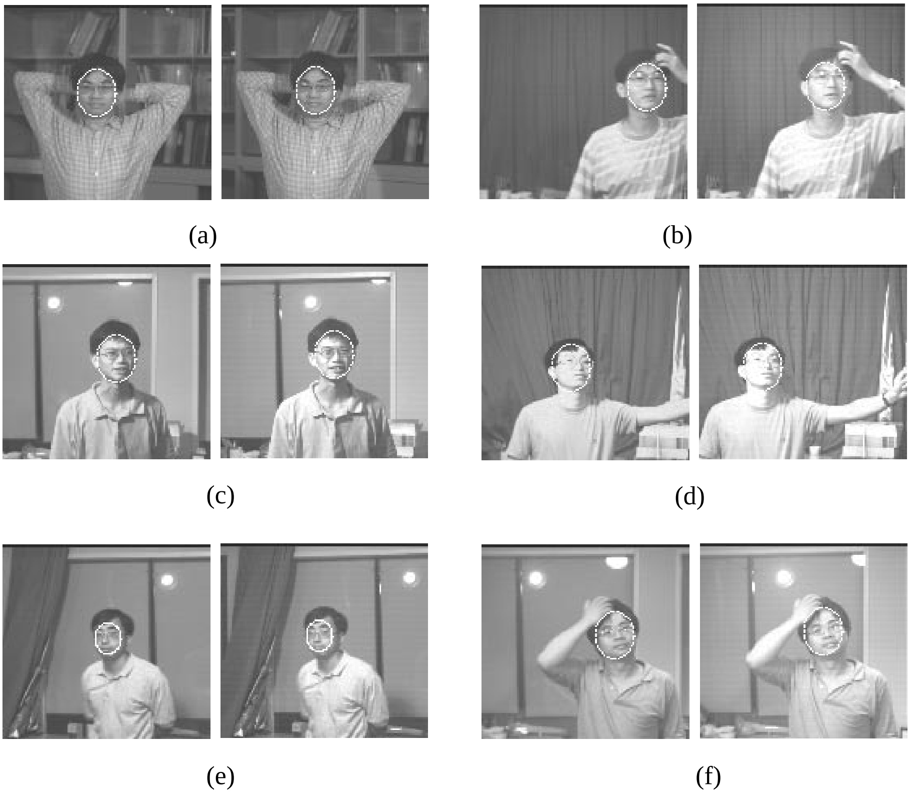
(a) The panorama of the enivronment in this experiment for head tracking.

(b) At this time ($t = 1$), the tracker switches to the tracking mode.

(c) The tracking results of (b).

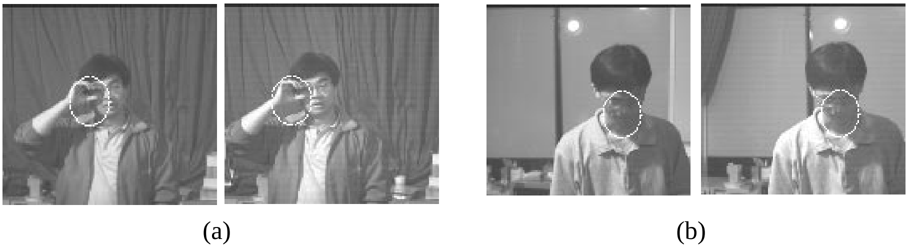
(d) The tracking results for the subsequent images.

Fig. 12. An example of applying our algorithm to detect and track a human head.



(a) (b) (c) (d) (e) (f)

Fig. 13. A few other examples where the detection and tracking of head motion are both successful.



(a) (b)

Fig. 14. Some examples where the results of tracking is acceptable but the estimates of the head position and size are not very precise.

head detector is shown in Fig. 6. Without the AJEO technique and the hierarchical structure, the computation time is 0.806 sec. With both the AEJO technique and the hierarchical structure, however, the computation time is improved to be 0.02 sec. It is obvious that the computation cost can be greatly reduced by using our efficient algorithm which utilizes both the hierarchical structure and AEJO technique. Comparing to the existing methods for human face (head) detection, our ML head detector by using the hierarchical structure and AEJO is very fast. In

| | |
|---|---|
| Detection | 0.802 sec. |
| Detection with AEJO | 0.084 sec. |
| Detection with hierarchical structure | 0.031 sec. |
| Detection with hierarchical structure and AEJO | 0.02 sec. |

Fig. 15. The computation times for detecting a human head in a $512 \times 512$ image with/without the AEJO technique and/or the hierarchical structure.

addition to fast speed, our 0 ML head detector is insensitive both to noise and to scale.

### 4.1.1. Insensitive to noise

In order to demonstrate that our ML head detector is insensitive to noise, we used the ML head detector to detect the human heads in the images having the Gaussian noises. In this paper, we tested one hundred of images having the Gaussian noises (standard deviations are from 0 to 10, in graylevel, were added in the same image) by using ML head detector. The image we used to test is $512 \times 512$ and shown in Figs. 16(a) and 16(b). The results are shown in Figs. 16(c) and 16(d).

From this experimental result, the sample variances including the location, $x$ and $y$ and the size, $s$, of the human head relative to the image size ($512 \times 512$) are quite small. That is, our ML head detector is insensitive to noise.

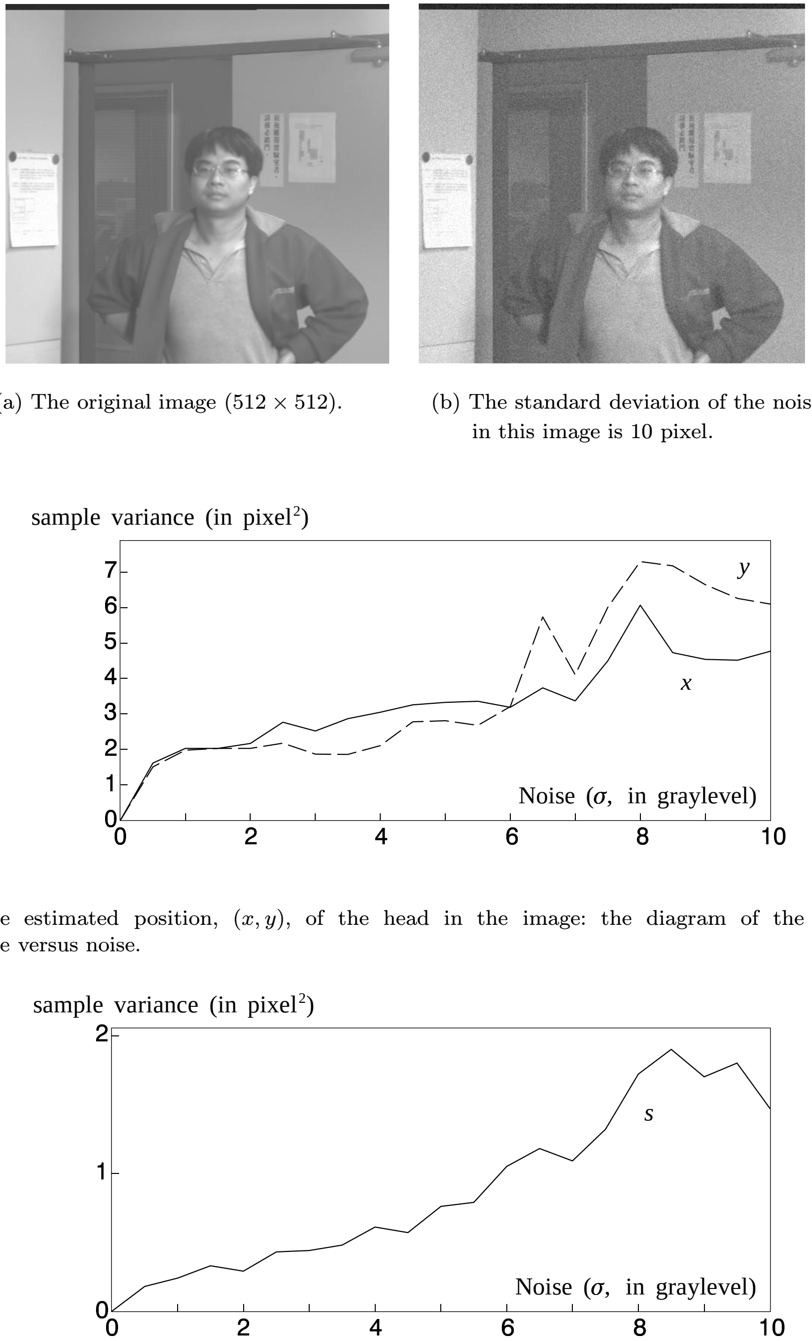
### 4.1.2. Insensitive to scale

We tried to test our ML head detector in the images with different scales. Figure 17 shows an example. We note that, in this example, because the size of the human head in the $32 \times 32$ image is too small (about $6 \times 8$ pixels for a human head in this scale), it is not easy to extract the reliable edge contour of this human head in such scale. Therefore, we do not detect the human head in this scale.

## 4.2. Human Face Detection in a Picture by Using the Verification with the Eyes

While this paper focuses on tracking human heads in a video sequence, this ML head detector proposed can be applied to detect human heads in a single image. As contrasted with the task for tracking a human head in a video sequence, there is no epipolar geometry constraint for detecting human faces in a single image. Therefore, other methods (constraints) should be used. In our experiment, the morphological operators[4] are adopted to extract the eye features. We choose the best one by combining the eyes information. A few candidates (for example, 36 candidates are chosen) are obtained by using the ML head detector. Morphological operators (for example, the *open* operation) and connected component methods are used to extract an eyes pair. If this candidate is a human head, it is required to have an eye pair. Figure 18 shows the results.

Note that morphological operators and the connected component methods can also be used in tracking a human head in a video sequence. However, we do not



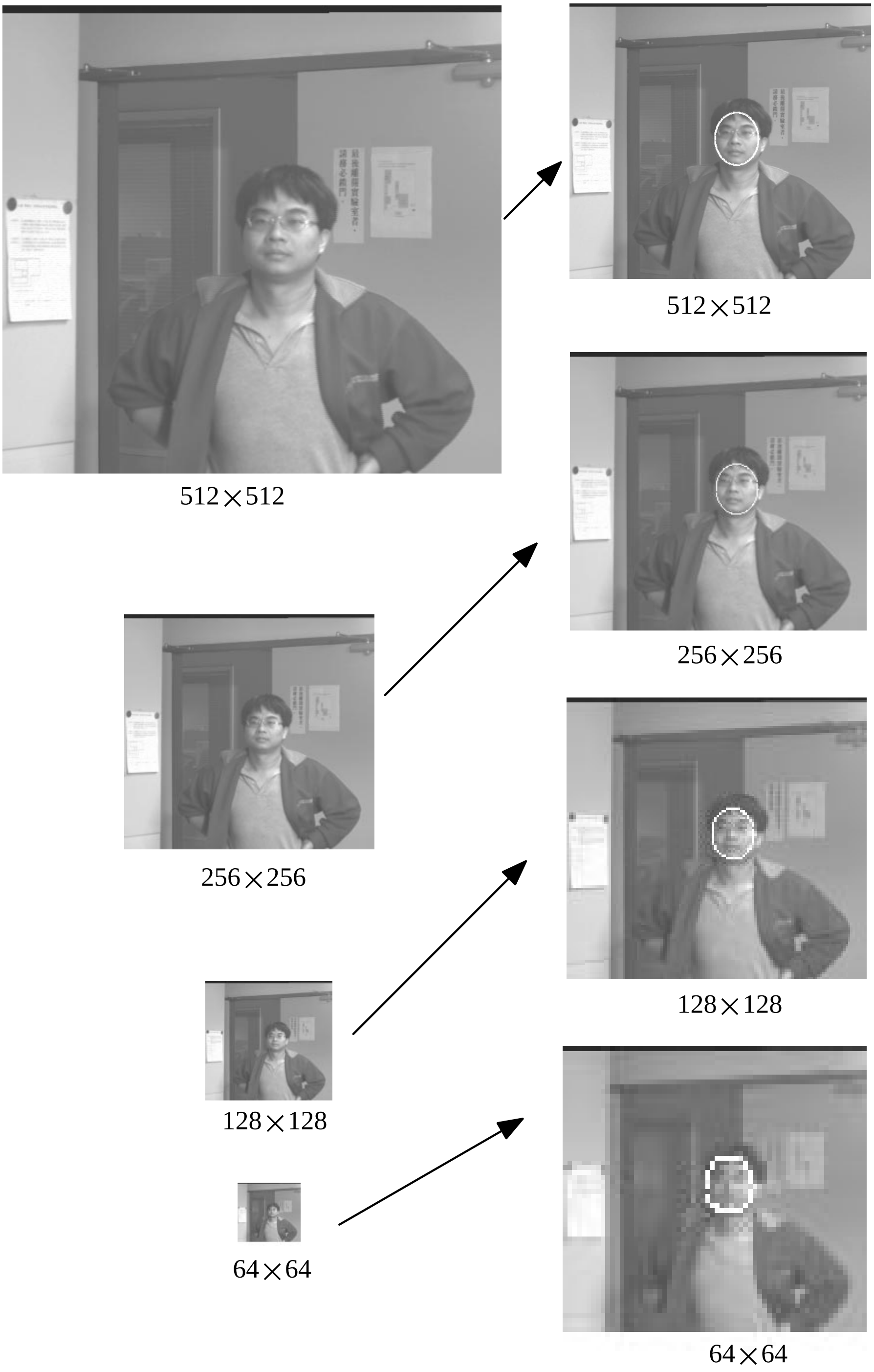
(a) The original image (512 × 512).

(b) The standard deviation of the noise in this image is 10 pixel.

(c) The estimated position, $(x, y)$, of the head in the image: the diagram of the sample variance versus noise.

(d) The size, $s$, of the head in the image: the diagram for the sample variance versus noise.

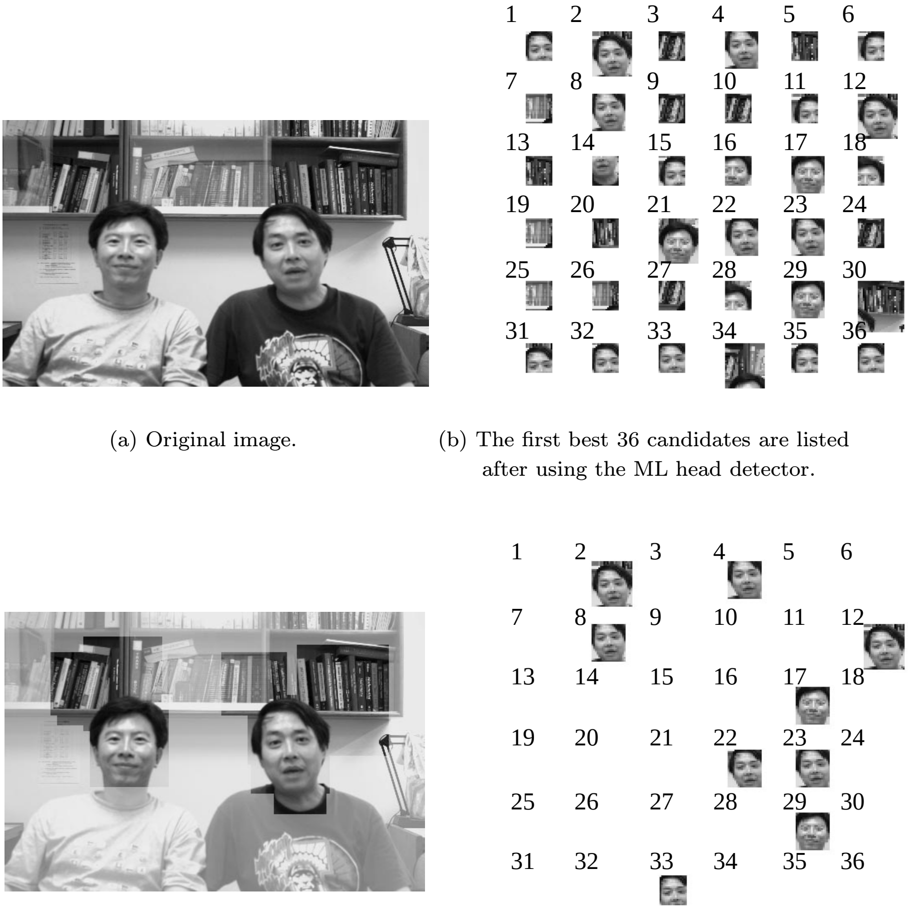
Fig. 16. Sensitivity analysis for ML head detector.



(a) Images with different scales.

(b) The results of head detection.

Fig. 17. An example of head detection in different scales by using the ML head detector.



(a) Original image.

(b) The first best 36 candidates are listed after using the ML head detector.

(c) All candidates in (a) are overlaid in the darker region of the image.

(d) The results after verification with the eyes.

Fig. 18. The results of human head detection (two human heads in this single image).

have to use them for verifying the detection of human faces in stereo video sequences if good results can be readily obtained by using the mutually-supported constraint. After all, the computation time is an important factor for the tracking task, and the extra verification processes can be omitted if they are not critical to the performance.

## 5. CONCLUSION

In this paper, a new head tracking algorithm for automatically detecting and tracking human heads in complex backgrounds is proposed. Our head tracker consists of four modes: the entrance-detection mode, the tracking mode, the fixation mode and the disappearance mode.

The major contribution of this paper is that we proposed a new efficient ML head detection method which utilizes a simple elliptical model and a hierarchical structure to search for the head. Our head detector consists of two channels. Each channel is implemented by multiscale template matching. By using the simple elliptical model for the human head, our ML head detector can reliably locate human heads in images having complex backgrounds, and is relatively insensitive to illumination and rotation of the human heads. The computation of our head detector is similar to that of template matching. However, its computational cost can be greatly reduced by using an efficient algorithm which utilizes both the hierarchical structure and the adaptive early jump-out technique. The reason that the hierarchical structure can be adopted to reduce the search space is because the contour of a human head tends to remain salient in different scales. By using the efficient algorithm in implementing our head detector, the execution time for detecting the human heads in a 512 × 512 image is about 0.02 second in a Sparc 20 workstation (not including the time for image acquisition).

Based on the ellipse-based ML head detector, we have developed a head tracking system that can monitor the entrance of a person, detect and track the person's head, and then control the stereo cameras to focus their gaze on this person's head. Difference images are used to detect the entrance of a human. The ML head detector and the mutually-supported constraint are used to extract the corresponding ellipses in a stereo image pair. The 3D position computed from the centers of the two corresponding ellipses is then used for fixation. The performance of our head tracker can be further improved by combining other informations (e.g. color[19,22], facial features[34]) and further verifications (e.g. neural networks[10,25]).A well-calibrated active stereo head, the IIS-head, has been used to perform the experiments and has demonstrated that our head tracker is feasible and promising for practical uses.

Since there can be occasionally ellipse-like objects in the scenes, the results obtained by the ellipse-based head detector can only be treated as head candidates. However, in ordinary scenes, an ellipse-like object is usually indeed a human head if it happens to roughly have the size of one. Since we are using a stereo vision system, the size of the detected ellipse-like object can be easily obtained for verification. If a more reliable result is required, further verification for eliminating false head candidates may become essential. Once the tracking result (i.e. the detected human heads) is obtained, it can be used as an input for face recognition systems[18,30]. Also, we are currently developing a system which can compute 3D head orientation[11] based on the tracking results obtained by the method proposed in this paper.

## ACKNOWLEDGMENTS

This work was partially supported by the National Science Council, Taiwan, under grants NSC88-2213-E-001-010.

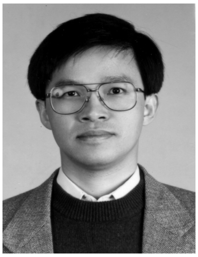
## REFERENCES

1. A. Azarbayejani, T. Starner, B. Horowitz, and A. Pentland, "Visually controlled graphics," *IEEE Trans. Patt. Anal. Mach. Intell.* **15**, 6 (1993) 602–605.
2. S. Basu, I. Essa and A. Pentland, "Motion regularization for model-based head tracking," *Proc. 13th Int. Conf. Pattern Recognition*, Vienna, Austria, 1996, pp. 611–616.
3. M. C. Burl, T. K. Leung and P. Perona, "Face localization via shape statistics," *Proc. Int. Workshop on Automatic Face- and Gesture-Recognition*, Zurich, 1995, pp. 154–159.
4. G. Chow and X. Li, "Towards a system for automatic facial feature detection," *Patt. Recogn.* **26**, 12 (1993) 1739–1755.
5. J. H. Chuang and H. Y. Chen, "A real-time visual tracking system using FPGA," *Proc. Conf. Computer Vision, Graphics and Image Processing*, Taiwan, 1996, pp. 143–150.
6. A. J. Colmenarez and T. S. Huang, "Maximum likelihood face detection," *Proc. 2nd Int. Conf. Automatic Face- and Gesture-Recognition*, Killington, Vermont, 1996, pp. 307–311.
7. T. Darrel, G. Gordon, W. Woodfill, H. Baker and M. Harville, "Real-time people tracking in open environments using integrated stereo, color, and face detection," *Workshop on Visual Surveillenace, Int. Conf. Computer Vision*, Bombay, India, 1998, pp. 26–32.
8. P. Fieguth and D. Terzopoulos, "Color-based tracking of heads and other mobile objects at video frame rates," *Proc. Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, 1997, pp. 21–27.
9. A. Gee and R. Cipolla, "Fast visual tracking by temporal consensus," *Imag. Vis. Comput.* **14** (1996) 105–114.
10. C. C. Han, H. Y. M. Liao, G. J. Yu and L. H. Chen, "Fast face detection via morphology-based pre-processing," *Proc. 9th Int. Conf. Image Analysis and Processing* Florence, Italy, 1997, pp. 469–476; Lecture Notes in Computer Science Series **1311**.
11. T. Horprasert, Y. Yacoob and L. S. Davis, "Computing 3-D head orientation from a monocular image sequence," *Proc. 2nd Int. Conf. Automatic Face- and Gesture-Recognition*, Killington, Vermont, 1996, pp. 242–247.
12. H. C. Huang and Y. P. Hung, "Adaptive early-jump-out technique for fast motion estimation in video coding," *Graph. Model Imag. Process.* **59**, 6 (1997) 388–394.
13. A. Jacquin and A. Eleftheriadis, "Automatic location tracking of faces and facial features in video sequences," *Proc. Int. Workshop on Automatic Face- and Gesture-Recognition*, Zurich, 1995, pp. 142–147.
14. Y. Kameda, M. Minoh and K. Ikeda, "Three dimensional motion estimation of a human body using a difference image sequence," *Proc. 2nd Asian Conf. Computer Vision* **2**, Singapore, 1995, pp. 181–185.
15. C. Kervrann, F. Davoine, P. Prez, R. Forchheimer and C. Labit, "Generalized likelihood ratio-based face detection and extraction of mouth features," *Patt. Recogn. Lett.* **18** (1997) 899–912.
16. K. F. Lai and R. T. Chin, "Deformable contours: modeling and extraction," *IEEE Trans. Patt. Anal. Mach. Intell.* **17**, 11 (1995) 1084–1090.
17. C. H. Lee, J. S. Kim and K. H. Park, "Automatic human face location in a complex background using motion and color information," *Patt. Recogn.* **29**, 11 (1996) 1877–1889.
18. H. Y. M. Liao, C. C. Han, G. J. Yu, H. R. Tyan, M. C. Chen and L. H. Chen, "Face recognition using a face-only database: a new approach," *Proc. 3rd Asian Conf. Computer Vision*, Hong Kong, 1998, pp. 742–749; Lecture Notes in Computer Science Series **1352**.

19. S. J. McKenna, Y. Raja and S. Gong, "Object tracking using adaptive color mixture models," *Proc. 3rd Asian Conf. Computer Vision*, Hong Kong, 1998, pp. 615–622; Lecture Notes in Computer Science Series **1351**.
20. B. Moghaddam and A. Pentland, "Maximum likelihood detection of faces and hands," *Proc. Int. Workshop on Automatic Face and Gesture Recognition*, Zurich, 1995, pp. 122–128.
21. N. Oliver, A. Pentland and F. Bérard, "LAFTER: lips and face real time tracker," *Proc. Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, 1997, pp. 123–129.
22. Y. Raja, S. J. McKenna and S. Gong, "Segmentation and tracking using color mixture models," *Proc. 3rd Asian Conf. Computer Vision*, Hong Kong, 1998, pp. 607–614; Lecture Notes in Computer Science Series **1351**.
23. J. M. Rehg, M. Loughlin and K. Waters, "Vision for a smart kiosk," *Proc. Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, 1997, pp. 690–696.
24. P. L. Rosin, "Thresholding for change detection," *Proc. Int. Conf. Computer Vision*, Bombay, India, 1998, pp. 274–279.
25. H. A. Rowley, S. Baluja and T. Kanade, "Neural network-based face detection," *IEEE Trans. Patt. Anal. Mach. Intell.* **20**, 1 (1998) 23–38.
26. E. Saber and A. M. Takalp, "Face detection and facial feature extraction using color, shape and symmetry-based cost functions," *Proc. 13th Int. Conf. Pattern Recognition* **3**, Vienna, Austria, 1996, pp. 654–658.
27. S. W. Shih, Y. P. Hung and W. S. Lin, "Four stage method for accurate calibration of an active binocular head," *Proc. Workshop on 3D Computer Vision 97*, 1997, pp. 49–56.
28. K. K. Sung and T. Poggio, "Example-based learning for view-based human face detection," *IEEE Trans. Patt. Anal. Mach. Intell.* **20**, 1 (1998) 39–51.
29. K. Sobottka and I. Pitas, "Extraction of facial regions and features using color and shape information," *Proc. 13th Int. Conf. Pattern Recognition* **3**, Vienna, Austria, 1996, pp. 421–425.
30. M. Turk and A. Pentland, "Eigenfaces for Recognition," *J. Cognitive Neurosci.* **3**, 1 (1991) 71–86.
31. C. Vieren, F. Cabestaing and J. G. Postaire, "Catching moving objects with snakes for motion tracking," *Patt. Recogn. Lett.* **16** (1995) 679–685.
32. G. Yang and T. Huang, "Human face detection in a complex background," *Patt. Recogn.* **27**, 1 (1994) 53–63.
33. J. Yang and A. Waibel, "A real-time face tracker," *Proc. 3rd Workshop on Applications of Computer Vision*, Sarasota, Florida, 1996, pp. 142–147.
34. K. C. Yow and R. Cipolla, "Feature-based human face detection," *Imag. Vis. Comput.* **15**, 9 (1997) 713–735.
35. K. C. Yow and R. Cipolla, "Enhancing human face detection using motion and active contours," *Proc. 3rd Asian Conf. Computer Vision*, Hong Kong, 1998, pp. 515–522; Lecture Notes in Computer Science Series **1351**.

**Cheng-Yuan Tang** received the B.Sc. and M.Sc. degrees both in engineering science from National Cheng Kung University of Taiwan in 1989 and 1991, respectively. He received the Ph.D. degree in computer science and information engineering from National Chiao Tung University of Taiwan in 1999. During 1991–1992, he was a research assistant in the Institute of Information Science, Academia Sinica, Taiwan. After graduating from National Chiao Tung University, he joined the Department of Information Management of Hua-Fan University, Taiwan, as an assistant professor.

His research interests include computer vision, visual surveillance, pattern recognition, and virtual reality.



**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 1990, 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 of 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.

Dr. Hung has published more than 70 technical papers in the fields of computer vision, pattern recognition, image processing, and robotics. In addition to the above topics, his current research interests also include visual surveillance, virtual reality, human-computer interface, and visual communication.

**Zen Chen** received the B.Sc. degree from National Taiwan University in 1967, the M.Sc. degree from Duke University, Durham, North Carolina, in 1970, and the Ph.D. degree from Purdue University, West Lafayette, Indiana, in 1973, all in electrical engineering. After graduating from Purdue University, he joined Burroughs Corporation, Detroit, Michigan, where he was engaged in the development of a document recognition system. In 1974, he began to teach at National Chiao Tung University, Taiwan, Republic of China. He served as the Director of the Institute of Computer Engineering from 1975 to 1980. He spent the academic year 1981–1982 at Lawrence Berkeley Laboratory, University of California, Berkeley, California, as a visiting scientist. Later, in August 1989 he spent six months at Computer Vision Laboratory of the Center for Automation Research, University of Maryland, College Park, Maryland, as a visiting professor. Dr Chen is a member of Sigma Xi and Phi Kappa Phi. He is also a member of China Computer Society and Chinese Institute of Electrical Engineering. He was the first president of the Chinese Society of Image Processing and Pattern Recognition founded in 1990.

His current research interests include computer vision, pattern recognition, virtual reality, and parallel algorithms and architectures.