Published in IEEE Transactions on Systems, Man and Cybernetics- Part A, Vol. 36, No. 6, pp. 2146-2151, Nov. 2006

On Automatic Collection of Multi-Angle Face Prototypes

Ping-Hsien Lee, Chu-Song Chen, *Member, IEEE*, and Wen-Yan Chang, *Student Member, IEEE*

***Abstract*—Although extremely tedious, collecting multi-angle facial prototypes is essential for face-related applications such as detection, tracking, recognition, and pose estimation. To reduce the work load, we propose a user-friendly interface that is designed to collect multi-angle facial prototypes in a fast and tractable manner. A user-friendly interface called *dynamic cursor* is designed to guide users in the operation of orientation-enforced video acquisition. A new problem, called the *orientation-enforced face detection problem* is defined and solved for multi-angle facial prototype collection. Our method can help to build a fast and user-friendly face registration module for face-related applications.**

***Index Terms*—Facial prototypes collection, face detection, face tracking.**

I. Introduction

COLLECTING multi-angle facial prototypes is very tedious and time-consuming. The process typically requires a volunteer to orientate his/her head according to the system operator's instructions, while images are taken from various angles. Each image must then be cropped so that only the facial region is retained. Also, to ensure that only the appropriate facial regions are cropped, each image usually has to be cropped manually.

The on-line collection of multi-angle facial prototypes of new users is important in order to build a good face-related application system. For example, in the eigenface method proposed by Turk and Pentland [12], multiple face samples of a person have to be collected to build an eigenspace. It is particularly important if an application system is moved to a different environment, where facial prototype collection must usually be performed again, in order to adapt the system to the new environment. On-line collected facial prototypes can also serve as useful information to tune or re-train application systems. However, a tedious facial prototype collection process hinders the willingness of people to become new users. Hence, an efficient on-line multi-angle human facial prototype collection process is important for real applications and is one of the key issues in making face-related applications popular.

In this paper, we propose a method for the efficient collection of multi-angle facial prototypes. We use an interactive technique called *dynamic cursor*, which is a moving cursor that guides the user's attention. The remainder of the paper is organized as follows. Section II describes the dynamic-cursor system. Section III describes the method developed to solve the orientation-enforced face detection problem in a video. Section IV addresses some implementation issues. In Section V, we give the results of our experiments. In Section VI, an application example is presented. Finally, our conclusions and a discussion are given in Section VII.

II. Dynamic Cursors

Our system framework consists of a mainframe, a monitor screen, and a camera that is mounted on the top or bottom of the screen. A user sits in front of the monitor in a prescribed position and rotates his/her head. We then take a video of the user and crop multi-angle facial prototypes automatically.

In this system, the user may rotate his/her head arbitrarily and an automatic cropping process can be used to find the facial regions in the video sequence. However, facial prototypes collected in this manner may not contain all the required angles. To overcome this difficulty, a moving cursor is used to guide the user's head movement so that video frames can be taken from prescribed angles. Our system configuration is illustrated in Fig. 1(a). To collect a complete set of facial prototypes within a range in which the screen remains visible to the user, the cursor movement is designed to follow a zigzag track, as shown in Fig. 1(b). A user is asked to rotate his/her head by following the cursor while the video is being made. As a result, each frame of the video contains a facial image that corresponds to a known cursor position. An example of the acquisition process and the resulting video is shown in Fig. 2.

However, cropping a facial region in each frame of the video causes a particular face detection problem. Since the user is asked to rotate his/her head so that the moving cursor remains visible to him/her (or more intuitively, the user can keep "facing" the cursor), there is a one-to-one mapping between the cursor positions and the head orientations. We call this specific detection problem the orientation-enforced face detection problem because of the a priori knowledge that each frame contains a facial region in association with a given cursor position (or head orientation). This problem has not been well studied before. We will focus on the orientation-enforced face detection problem and present an effective approach, based on video inputs, for solving it.

Manuscript received July 16, 2004. This work was supported in part by National Science Council, R.O.C. under Grants NSC-93-2213-E-001-011, NSC-93-2752-E-002-007-PAE, and 93-EC-17-A-02-S1-032.

P. H. Lee is with the Institute of Information Science, Academia Sinica, Taipei, Taiwan. He is also with Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan.

C. S. Chen is with the Institute of Information Science, Academia Sinica, Taipei, Taiwan. (phone: +886-2-27883799ext1310; fax: +886-2-27824814; e-mail: song@iis.sinica.edu.tw).

W. Y. Chang is with the Institute of Information Science, Academia Sinica, Taipei, Taiwan. He is also with Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan.

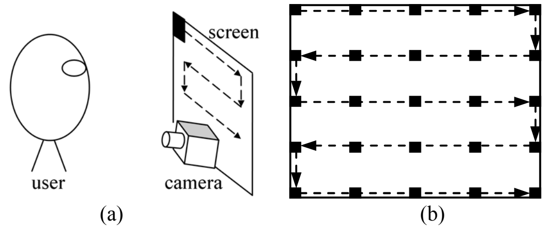


Fig. 1. (a) Configuration of the acquisition system. (b) The zigzag movement track of the dynamic cursor.

III. Orientation-Enforced Face Detection In A Video

Before presenting our detailed algorithm, we analyze two visual clues that are useful for solving the video-based orientation-enforced face detection problem. The first clue is that each single frame in the input video contains a face with a roughly known orientation, while the second clue is that the face location in the video varies continuously. To utilize these visual clues in a consistent and integrated way, we propose the following probabilistic framework. The video-based orientation-enforced face detection problem is formulated as a shortest-path problem in a matching graph, in which the two visual clues are encoded as the node and edge costs of the graph, respectively.

A. A Probabilistic Framework

In our approach, the video-based orientation-enforced face detection problem is formulated as one of maximizing an *a posteriori* probability measure. Given a video containing K frames, I_1, I_2, …, I_K, each frame I_k (k=1,...K) is taken under the k-th orientation (i.e., the k-th cursor position). Our purpose is to find the most probable facial region in each frame. Several scaled input images are generated to find possibly scaled facial regions in each input image. Let a set of ascending scaling factors be $S = \{s_1,...,s_L \mid 0<s_1<s_2<...<s_L\}$. We define the l-th scaled input image, I_k^l, as a $(s_lW)\times(s_lH)$ image generated by linearly scaling I_k, where W and H are respectively the width and height of I_k (k=1,..., K). Since a facial region may occur in any position and scale in an image, we define the facial region candidate as I_{x_k}, where x_k is a position/scale vector (c_k, r_k, l_k) that specifies an $n \times n$ image block with its upper-left point being (c_k, r_k) in the scaled image frame $I_k^{l_k}$, for $0\le c_k<(s_{l_k}W)-n$, $0\le r_k<(s_{l_k}H)-n$, l_k=1,..., L.

Let ω_k be the set consisting of all possible facial prototypes under the k-th orientation. We hope to find the proper facial regions $I_{x_k}^*$ corresponding to the position/scale vector x_k, k=1,…, K, which maximize the following *a posteriori* probability:

$$\arg\max_{x_1,...,x_K} P(x_1,\dots,x_K|I_1,\dots,I_K)$$

$$= \arg\max_{x_1,...,x_K} P(I_1,\dots,I_K|x_1,\dots,x_K)\cdot P(x_1,\dots,x_K)\,/\,P(I_1,\dots,I_K)$$

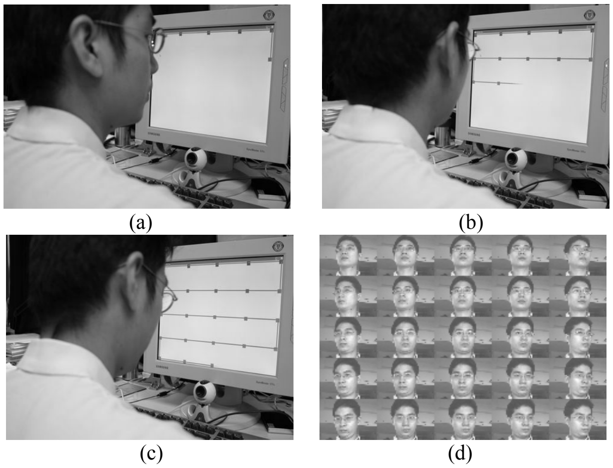


Fig. 2. (a)(b)(c) Illustrations of the video-acquisition process. (d) Image frames of the acquired video.

$$= \arg\max_{x_1,...,x_K} P(I_1,\dots,I_K|x_1,\dots,x_K)\cdot P(x_1,\dots,x_K). \quad (1)$$

Assume that:

(i) $P(I_k|x_k)$, k=1,…, K are independent of each other, and

(ii) $P(x_1,\dots, x_K)$ can be modeled by a first-order Markov chain. That is, $P(x_{k+1}|x_1,\dots, x_k) = P(x_{k+1}|x_k)$ for all k=1,…, K–1.

Then, (1) becomes:

$$\arg\max_{x_1,...,x_K} \prod_{k=1,...,K} P(I_k|x_k)\cdot P(x_1)\cdot \prod_{k=1,...,K-1} \mathrm{P}(x_{k+1}|x_k)$$

$$= \arg\max_{x_1,...,x_K} P(x_1) \prod_{k=1,...,K} P(I_{x_k} \in \omega_k)\cdot \prod_{k=1,...,K-1} \mathrm{P}(x_{k+1}|x_k) \quad (2)$$

In (2), $P(I_{x_k} \in \omega_k)$ is the probability that the cropped region I_{x_k} is a facial image at the k-th orientation, which is referred to as the *probability caused by prototype evidence* (PPE). The other term $P(x_{k+1}|x_k)$ is the transition probability from the current position/scale x_k to the next one x_{k+1}, which encodes motion continuity and is referred to as the *probability caused by temporal continuity* (PTC). To evaluate (2), the above probabilities are specified as follows.

The probability $P(I_{x_k} \in \omega_k)$ is the probability that I_{x_k} is an k-th orientation facial image. We assume that ω_k forms a Gaussian distribution with mean f_k and covariance matrix $\sum_k$ in the $n\times n$ dimensional image space. Hence:

$$P(I_{x_k} \in \omega_k) = \exp(-(I_{x_k} - f_k)^t \textstyle\sum_k^{-1} (I_{x_k} - f_k)/2\,)/Z_k,$$

for $k=1,...,K,$ where Z_k is the normalization constant making $P(I_{x_k} \in \omega_k)$ a legal probability density function.

Because consecutive images of a moving face are inputs, $P(x_{k+1}|x_k)$ will be large when x_k is close to x_{k+1}. We also model $P(x_{k+1}|x_k)$ as a Gaussian distribution by:

$$P(x_{k+1}|x_k) = \begin{cases} \exp(-(\| c_k - c_{k+1} \|^2 + \| r_k - r_{k+1} \|^2)/2\sigma^2)/Z, & \text{if } l_{k+1} = l_k, \\ 0, & \text{otherwise,} \end{cases}$$

where $x_k=(c_k, r_k, l_{k,})$, $x_{k+1}=(c_{k+1}, r_{k+1}, l_{k+1})$, and Z is a constant for normalization. We set $P(x_{k+1}|x_k)$ to zero when the scale l_k

and l_{k+1} are different. It is because that, in our dynamic-cursor system, a user sits in front of the monitor with a generally fixed distance to the camera. The cropped faces of a user under various orientations thus have approximately the same scales.

An equal *a priori* assumption is applied to all initial positions/scales by setting $P(x_1) = 1/p$ for all $x_1 \in \Gamma = \{(c, r, l) \mid 0 \le c < (s_l W) - n,\ 0 \le r < (s_l H) - n,\ l = 1, \ldots, L\}$, where p is the cardinality of Γ.

Taking a natural log of (2), we derive the equation:

$$\arg\max_{x_1,\ldots,x_K} \ln\Big[P(x_1) \prod_{k=1,\ldots,K} P(I_{x_{k+1}} \in \omega_k). \prod_{k=1,\ldots,K-1} \mathrm{P}(x_{k+1}|x_k)\Big]$$

$$= \arg\min_{x_1,\ldots,x_K} J(x_1,\ldots,x_K; I_1,\ldots,I_K),$$

where $J(x_1,\ldots,x_K; I_1,\ldots,I_K) =$

$$\begin{cases} \infty, & \text{if } P(x_{k+1} \mid x_k) = 0 \\ \displaystyle\sum_{k=1}^{K} (I_{x_k} - f_k)^t \Sigma_k^{-1} (I_{x_k} - f_k)/2 + \sum_{k=1}^{K-1} (\| c_k - c_{k+1} \|^2 + \| r_k - r_{k+1} \|^2)/2\sigma^2 + \ln p, & \text{otherwise.} \end{cases}$$

From (3), finding the $x^*_1,\ldots, x^*_K$ that maximizes the *a posteriori* probability is equal to finding the $x^*_1,\ldots, x^*_K$ that minimizes the objective function $J(x_1,\ldots,x_K; I_1,\ldots,I_K)$.

This is computationally infeasible if a brute-force method is used to find the optimal solution. We solve this problem by transforming it into a shortest-path search problem in a *matching graph*, as shown below.

B. Matching Graph and Shortest Path

We construct a graph G with K levels. In each level, there are p nodes where p is the number of all facial region candidates in an input image. In the k-th level (k=1,...,K), nodes are denoted as $V_{k;1}$, $V_{k;2}$, …, $V_{k;p}$ which represent p facial region candidates in this input image. For adjacent levels k and k+1, (k=1,...,K−1), the edge $E_{k;j;i}$ connects $V_{k;j}$ and $V_{k+1;i}$ if $l_k = l_{k+1}$ (i.e., the two nodes are of the same scale). Fig. 3 shows an illustration of graph G.

We define the node cost and the edge cost as follows.

Node cost is used to encode the log likelihood of PPE in (3). Each node $V_{k:j}$ is assigned a cost

$$\mathrm{Cost}(V_{k:j}) = (I_{x_k} - f_k)^t \Sigma_k^{-1} (I_{x_k} - f_k). \qquad (4)$$

Edge cost is used to encode the log likelihood of PTC in (3). For each edge $E_{k;j;i}$, we define

$$\mathrm{Cost}(E_{k;j;i}) = (\| c_k - c_{k+1} \|^2 + \| r_k - r_{k+1} \|^2)/\sigma^2. \qquad (5)$$

The node and edge costs have very straightforward interpretations. Equation (4) is a Mahalanobis distance that measures the possibility that the candidate region I_{x_k} is an image of a face under the k-th orientation. Equation (5) is a penalty term that penalizes interrupted motions.

Each path, starting from a node in image 1 and ending at a node at input image K, represents a sequence of facial region candidates. The cost of the path is the sum of the node and edge costs along this path. Our goal is to find the minimal-cost (or the shortest) path in the matching graph. The matched nodes on the minimal-cost path represent the most probable facial region candidates that satisfy both prototype similarities and temporal consistencies. The Dijkstra algorithm [5] is used to find the shortest path. It solves the single-source shortest-path problem on a weighted, directed graph $G = (V, E)$ when all edge weights are nonnegative. It maintains a set Y of vertices with shortest-path weights from the root. The algorithm repeatedly adds another vertex to the set Y in which the vertex is the point with minimum shortest path to the root and belongs to the set of V−Y, and relaxes all edges leaving this vertex. A detailed description can be seen in [5]. After performing the Dijkstra algorithm, each node along the shortest path then represents the detected facial region in each frame.

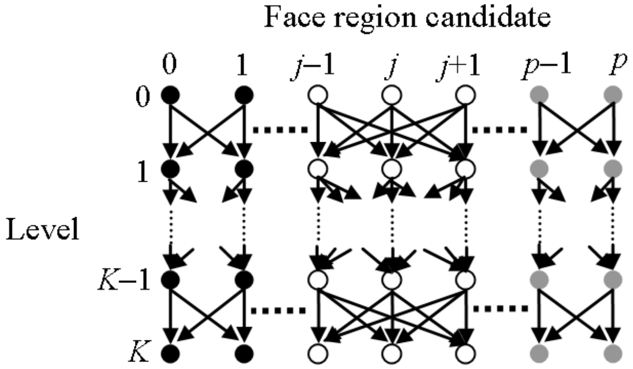


Fig. 3. An example of a matching graph. The graph has K levels and each level represents an input image. Nodes in the k-th level represent facial region candidates under the k-th orientation. Nodes of the same color are of the same scale. In this example, there are three scales marked with black, white, and gray. It is not allowed to go from one scale to another.

C. Pre-collected Training Data

To evaluate the node cost, we have to estimate: 1) f_k, the mean face under the k-th orientation, and 2) $\sum_k$, the covariance matrix. In our approach, videos of several users are taken for training (in the experiment, there were ten users) and a facial region in each frame is manually cropped. Each user has K orientations (in association with the pre-selected K cursor positions) to be cropped, and K is typically 25 in our work. For the k-th orientation, we calculate the mean face f_k and the sample covariance matrix $\sum_k$ of cropped faces (k=1, …, K). Fig. 4 shows the manually cropped training data of three users.

IV. Implementation Issues

A. Lighting Correction

To compensate for the problem caused by different illumination conditions, the light-correction process introduced in [9] is adopted to normalize the intensities of the facial region candidates in the input images and the

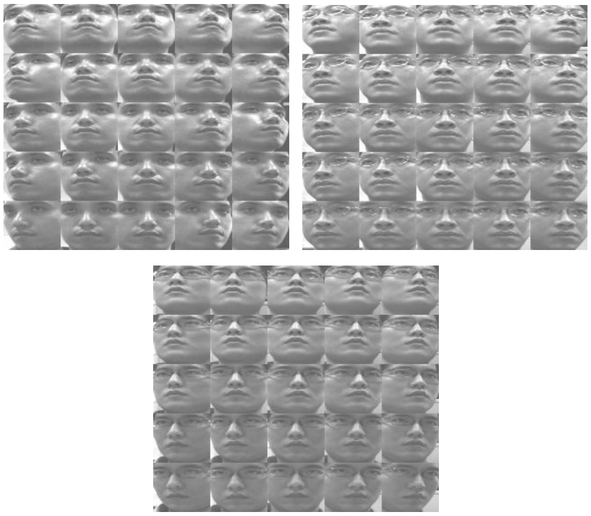


Fig. 4. The manually cropped training data of three of the ten subjects.

pre-collected facial templates. Here, we briefly review this method.

First, for each image in the video, we fit a plane to the intensities of it. Let the fitted plane be:

$$z = au+bv+c \tag{6}$$

where u, v is the pixel position in the image and z is the fitted intensity. We use the standard least-squared procedure to perform the fitting. To compensate for the illumination-variation influence, we subtract the fitted z value from the original intensity of each image. The subtracted image is further processed by histogram equalization to form a lighting-compensated image, which then serves as the input image for facial prototype collection.

B. *Coarse-to-fine Shortest-path*

To find the facial region in the k-th input image, we treat all possible positions as facial region candidates in the probabilistic framework introduced in Section III. However, the computational cost may be high. To speed up the process, we use a coarse-to-fine strategy in our implementation. The idea is to repeat the shortest-path search algorithm T times from a low-resolution image to a high-resolution image.

In our implementation, an image pyramid is built for each image. The image pyramid is constructed by recursively halving the resolutions in both row and column directions of an image. At the coarsest-level, all the positions are considered as centers of possible candidate regions. When the shortest path has been found at the coarsest level, its corresponding candidate positions and scales serve as an initial estimate of those for the next finer level. Furthermore, the positions and scales that are far from the shortest path will be ignored in the next finer level. Hence, not all positions and scales are addressed in the finer level, which speeds up the search process. Although this does not ensure finding the globally shortest path in association with the highest resolution images, in our experience, very good sub-optimal paths can be found and the obtained solutions are very convincing.

V. Experiment Results

Each input image is 102×60 and scaled by factors {0.8, 0.9, 1, 1.1, 1.2}. The facial candidate region is an n×n block, where n=32. Two experiments are performed to evaluate the proposed approach. Section V.A. describes the model-selection process that is performed to tune the parameters, while Section V.B. shows the results under different conditions.

A. *Cross-validation Performance and Model Parameter Selection*

The only parameter that needs to be tuned is $1/\sigma^2$, the weight of the edge cost in (5) (where the weight of the node cost is 1). To determine the parameter with a good performance, we perform a 5-fold cross-validation model selection on the pre-collected training data from the ten subjects mentioned in Section III.C. Typically, in m-fold cross-validation, m examinations are performed and each of them involves (m-1)/m data for training and the remaining 1/m data for test. Hence, in 5-fold cross validation, the training data is divided into five parts and each part contains two users, where four parts of them are treated as training data in turn and the other part is treated as validation data.

For a particular parameter value to be evaluated, the facial prototypes of eight users (i.e., four folds) serve as the training data for calculating the mean face f_k and the sample covariance matrix $\sum_k$ for setting the node cost in (4). We use the video containing the faces of the other two users in the remaining fold as the testing data and employ our algorithm to find the facial regions automatically. After cross validation among the five folds has been done, the error measurement is defined as the average Euclidian distance between the center positions of the facial regions found automatically and those manually cropped. We have investigated σ over a wide range and found that σ=1 is one of the parameters that have the best performance. In this case, the average distance error is 7 (pixels) in a 640×480 image, compared with the manually cropped region. This is sufficiently accurate for many face-related applications.

B. *Other Experiments*

Experiments were carried out to evaluate the performance of our approach in different environments with different users. Fig. 5 shows the prototype collection result for a novel user in an environment different from the one in which the training prototypes were collected, and Fig. 5 and Fig. 6 show the results of dimming and brightening the lighting for the same user. Fig. 7 shows the prototypes collected in an outdoor environment. In general, it takes about 30 seconds to crop 25 images with 640×480 resolution by using the proposed method. These results demonstrate that our system works well in various conditions. Assume that the width and height of a screen are w and h, respectively, and the distance between user and screen is d. The left-right and up-down viewing angles can be calculated by using the equations $\theta_{left\text{-}right}$ = 2×$acot$(2×d/w) and $\theta_{top\text{-}down}$ = 2×$acot$(2×d/h), respectively. In our case, *w*, *h* and *d* are 38, 33, and 62 Centimeters, respectively, and the left-right the up-down angular differences are about 35° and 30°, respectively.

VI. An Application Example

The multi-angle facial prototypes collection method developed in this paper can be employed in many face-related applications. For example, a sequence of well-segmented facial images is a suitable input for sequence-based face recognition and verification. Furthermore, the cropped facial images with known orientation can also be applied to face-pose estimation. In this section, we implement a face-pose estimation system by using the conjugate-gradient search framework in [8], and the support vector regression (SVR) [6] to estimate the orientation of a human head, and the result is shown in Fig. 8. In [8], the authors proposed a face pose estimation framework by using the conjugate-gradient search. In their work, face is tracked by using off-line collected human facial templates firstly and then its orientation is estimated in a Gabor wavelet projected image space. In addition, Li *et al.* [6] suggested a face detection and recognition framework by using support vector machine (SVM). To increase its accuracy, the SVR is used to estimate head pose, but facial templates are cropped manually under controlled conditions in [8]. Instead of cropping facial regions manually from various video sources as training data, a prototype collection module is built by using the method introduced in this paper to automatically collect multi-angle facial prototypes. This greatly increases the operating convenience of the system and reduces the workload when new prototypes have to be collected.

VII. Conclusions And Discussion

We have designed an interactive system for the on-line collection of multi-angle facial prototypes in this paper. By guiding the user's attention with a cursor moving along a particular trajectory, we pose a new type of face-detection problem – the orientation-enforced face detection problem. To deal with this issue, we formulate a probabilistic framework. We show that the problem can be reduced to a shortest-path-searching problem in a matching graph, using the Dijkstra algorithm to solve it.

Experimental results show that our method can collect multi-angle facial prototypes effectively. The collected prototypes can be used for applications such as face pose estimation (for which an example is shown in Section VI); face recognition and authentication [1][3][4][7][10][13][14]; and multi-view face detection [2][11][15]. By using our approach, multi-angle facial prototype collection can be made easy and user friendly.

Our dynamic-cursor interface is not only suitable for multi-angle facial prototype collection, but also has great potential to be integrated into a face recognition task, leading to a new type of face recognition framework – the orientation-enforced face recognition framework. Therefore, it's likely that face recognition will be made easier, since rough face angles can serve as extra informative knowledge. In future research, we will investigate the orientation-enforced face recognition framework.

References

[1]. Z. Biuk and S. Loncaric, "Face Recognition from Multi-Pose Image Sequence," *Proceedings of 2nd International Symposium on Image and Signal Processing and Analysis*, pp. 319-324, Pula, Coratia, 2001.

[2]. Y. S. Chen, C. H. Su, and J. H. Chen, et al., "Video-based Eye Tracking for Autostereoscopic Displays," *Optical Engineering*, Vol. 40, Issue 12, pp. 2726-2734, 2001.

[3]. J. H. Chen and C. S. Chen, "Using Inter-Feature-Line Consistencies for Sequence-Based Object Recognition," *Proceedings of European Conference on Computer Vision, ECCV 2004*, Vol. 1, pp. 108-120, Prague, Czech, May 2004.

[4]. J. H. Chen, S. L. Yeh, and C. S. Chen, "Inter-Subspace Distance: A New Method for Face Recognition with Multiple Samples," *Proceedings of 17th International Conference on Pattern Recognition, ICPR 2004*, Vol. 3, pp. 140-13, Cambridge, UK, 2004.

[5]. T. H. Cormen, C. E. Leiserson and R. L. Rivest, *Introduction to Algorithms*, MIT Press, Cambridge, MA, 1990.

[6]. Y. Li, S, Gong and H. Liddell, "Support Vector Regression and Classification Based Multi-view Face Detection and Recognition", *Proceedings of Fourth IEEE International Conference on Automatic Face and Gesture Recognition 2000, FG 2000*, pp. 300-305, Grenoble, France, 2000.

[7]. X. Liu, T. Chen, and H. Liddel, "Video-Based Online Face Recognition Using Identity Surface," *Pattern Recognition, Special Issue on Biometric*, Vol. 36, Issue 2, pp. 313-328, 2003.

[8]. S. McKenna and S. Gong, "Real Time Face Pose Estimation", *International Journal on Real Time Imaging*, Vol. 4, pp. 333-347, 1998.

[9]. H. Rowley, S. Baluja and T. Kanade, "Neural Network-Based Face Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 1, pp.23-38, 1998.

[10]. G. Shakhnarovich, J. W. Fisher, and T. Darrell, "Face Recognition from Long-Term Observations," *Proceedings of European Conference on Computer Vision, ECCV 2002*, Vol. 3, pp. 851-868, Copenhagen, Denmark, 2002.

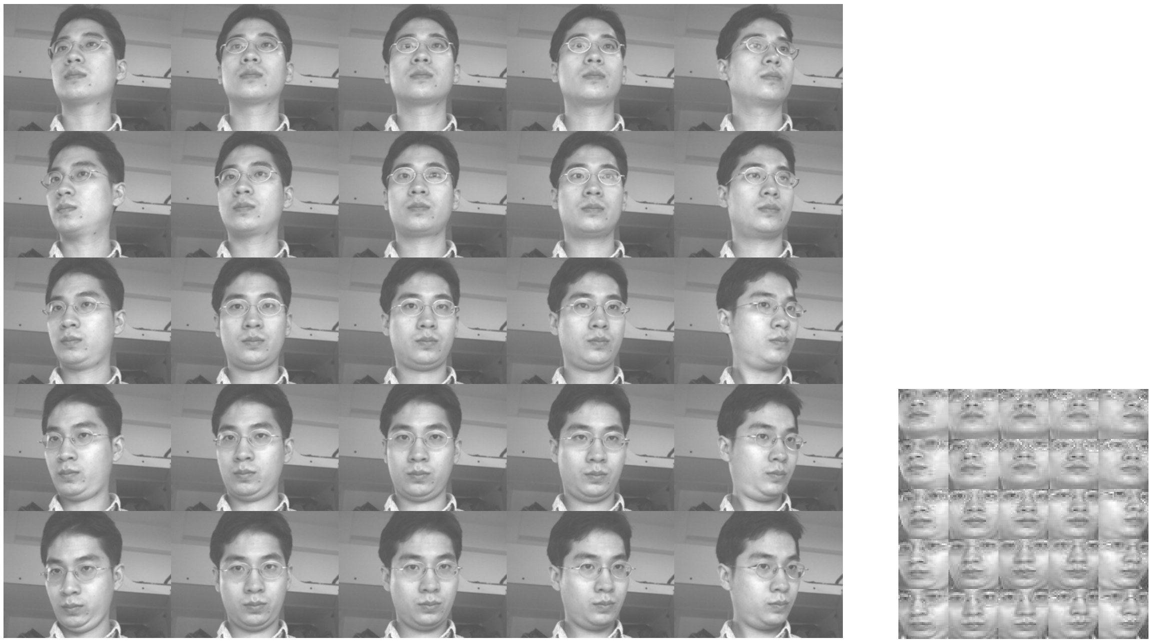
[11]. K. K. Sung, and T. Poggio, "Example-based Learning for View-Based Human Face Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 1, pp. 39-51, 1998.

[12]. M. Turk and A. Pentland, "Eigenfaces for Recognition," *Journal of Cognitive Neuroscience*, Vol. 3, No. 1, pp. 71-86, 1991.

[13]. L. Wolf and A. Shashua, "Kernel Principal Angles for Classification Machines with Applications to Image Sequence Interpretation," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2003*, Vol. 1, pp. 635-642, Madison, Wisconsin, USA, June 2003.

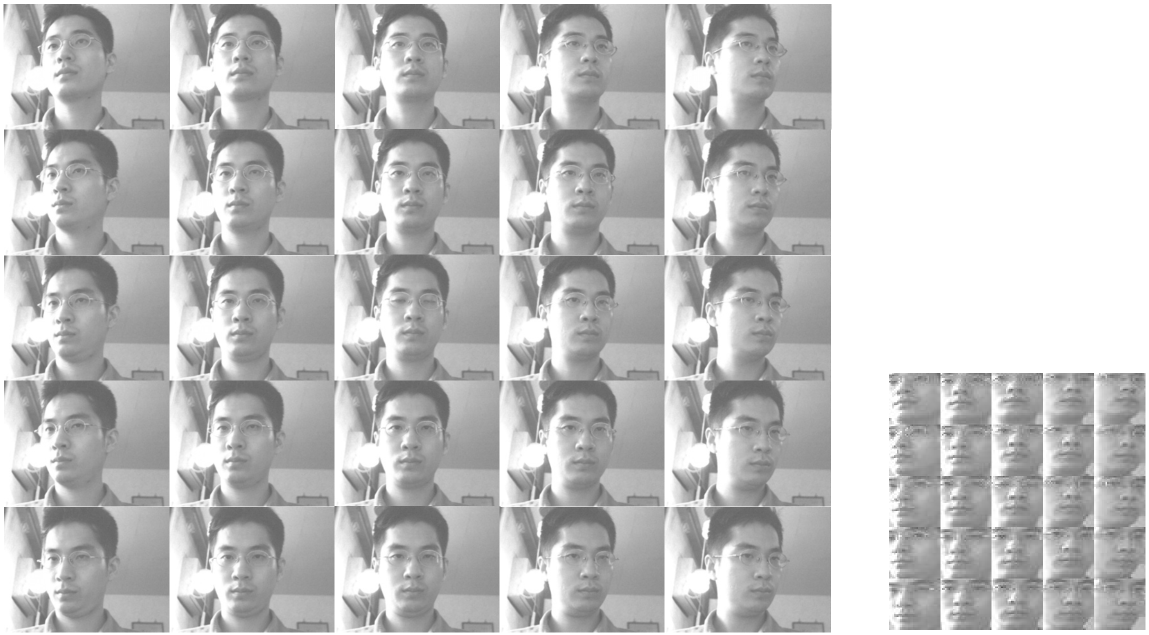
[14]. O. Yamaguchi, K. Fukui, and K. Maeda, "Face Recognition Using Temporal Image Sequence," *Proceedings of International Conference on Automatic Face and Gesture Recognition, FG98*, Nara, Japan, pp. 318-323, 1998.

[15]. Z. Q. Zhang, L. Zhu, S. Z. Li and H. Zhang. "Real-Time Multi-View Face Detection", Proceedings of the *5th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2002*, pp. 149-154, Washington D.C., USA, 2002.



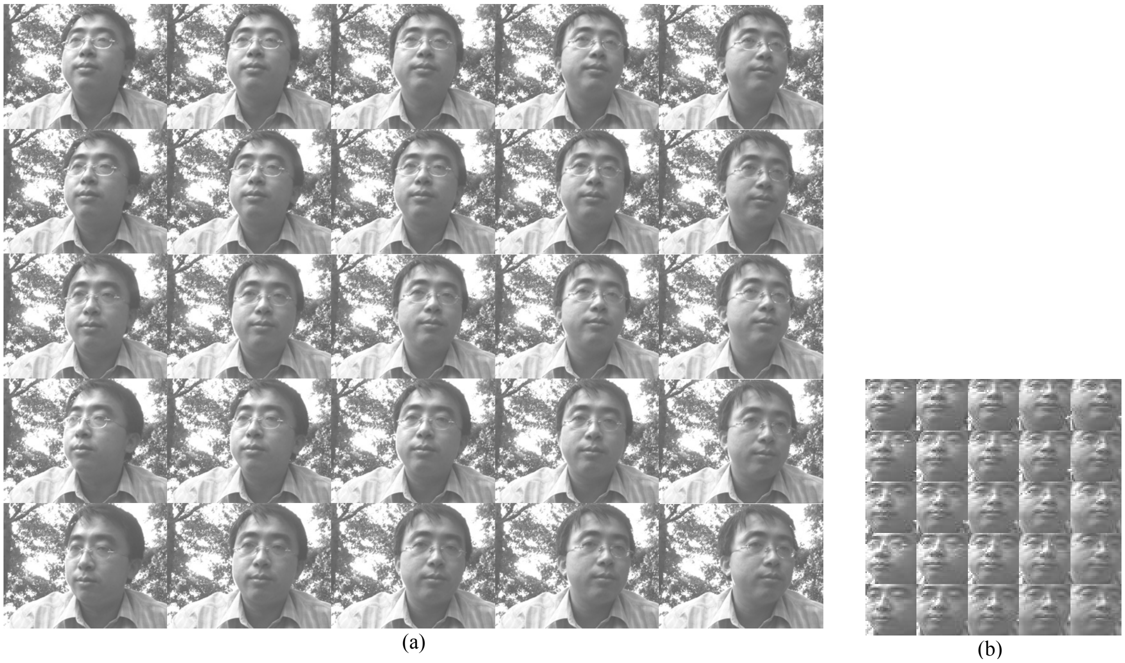
(a) (b)

Fig. 5. An example of a multi-angle facial prototype collected in an indoor environment. (a) Image frames of an acquired video. (b) Multi-angle facial prototypes cropped by using our method with image resolutions being 32×32.



(a) (b)

Fig. 6. Collection of multi-angle facial prototypes of the same person of Fig. 6 under different illumination conditions. (a) Image frames of an acquired video. (b) Multi-angle facial prototypes cropped by using our method with image resolutions being 32×32.



(a)

(b)

Fig. 7. Multi-angle facial prototypes collected in an outdoor environment. (a) Image frames of an acquired video. (b) Multi-angle facial prototypes cropped by using our method with image resolutions being 32×32.

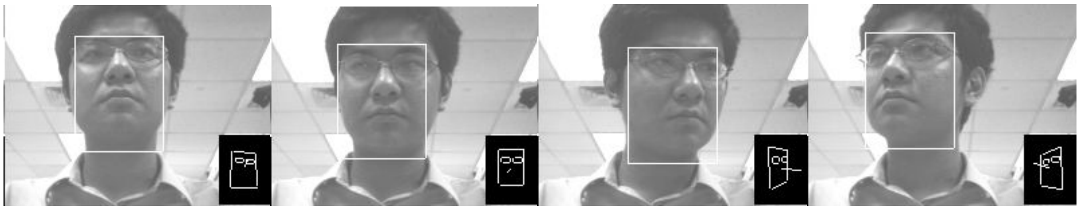


Fig. 8. Head orientation estimation combining conjugate-gradient search and support vector regression. The enrollment module of this system was built using the multi-angle facial prototype collection method proposed in this paper.