We are IntechOpen, the world's leading publisher of Open Access books
Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Real-Time Multi-Face Recognition and Tracking Techniques Used for the Interaction between Humans and Robots[1]

Chin-Shyurng Fahn and Chih-Hsin Wang[2]
National Taiwan University of Science and Technology, Taipei
Taiwan, R. O. C.

1. Introduction

The technology of biometric recognition systems for personal identification commonly manipulate the input data acquired from irises, voiceprints, fingerprints, signatures, human faces, and so on. The recognition of irises, voiceprints, fingerprints, and signatures belongs to the passive methods that require the camera with a high resolution or capture people's biometric information at a short range. These methods are not suitable for our person following robot to be developed, because they cannot provide convenience for users. Face recognition belongs to one of the active methods that users need keep away from a camera at a certain distance only. In this chapter, face recognition is regarded as a kind of human computer interfaces (HCIs) that are applied to the interaction between humans and robots. Thus, we attempt to develop an automatic real-time multiple faces recognition and tracking system equipped on the robot that can detect human faces and confirm a target in an image sequence captured from a PTZ camera, and keep tracking the target that has been identified as a master or stranger, then employ a laser range finder to measure a proper distance between target's owner and the robot.

Generally, three main procedures: face detection, face recognition, and face tracking are implemented on such a system. In literature, a large number of face localization techniques had been proposed. According to the literature (Hjelmås & Low, 2001), the methods for face detection can be basically grouped into feature-based and image-based approaches. The development of the feature-based approaches can be further divided into three areas: active shape models, feature analysis, and low-level analysis for edges, grey-levels, skin colours, motions, and so forth. On the other hand, the image-based approaches can be categorized into linear subspace methods, neural networks, and statistical approaches.

The development of face recognition is more and more advanced in the past twenty years (Zhao et al., 2003). Each system has its own solution. At present, all researches about face recognition take the features that are robust enough to represent different human faces. A

novel classification method, called the nearest feature line (NFL), for face recognition was proposed (Li et al., 1999). The derived FL can capture more variations of face images than the original feature points do, and it thus expands the capacity of an available face image database. However, if there are a lot of face images in the database, the recognition accuracy will reduce. Two simple but general strategies for a common face image database are compared and developed two new algorithms (Brunelli & Poggio, 1993); the first one is based on the computation of a set of geometrical features, such as nose width and length, mouth position, and chin shape, and the second one is based on almost-grey-level template matching. Nevertheless, under the different lighting conditions, the characteristic of geometry will change. In the literature (Er et al., 2002), the researchers combined the techniques of principal component analysis (PCA), Fisher's linear discriminant, and radial basis function to increase the correct rate of face recognition. However, their approach is unable to carry out in real time.

On the whole, the methods of tracking objects can be categorized into three ways: match tracking, predictive tracking, and energy functions (Fan et al., 2002). Match tracking has to detect moving objects in the entire image. Although the accuracy of match tracking is considerable, it is very time-consuming and we can not improve the performance effectively. Energy functions are often adopted in a snake model. They provide a good help for contour tracking. In general, there are two methods of predictive tracking: the Kalman filter and a particle filter. The Kalman filter has great effects when the objects move in linear paths, but it's not appropriate for the non-linear and non-Gaussian movements of objects. On the contrary, the particle filter performs well for non-linear and non-Gaussian problems. A research team completed a face tracking system which exploits a simple linear Kalman filter (An et al., 2005). They used a small number of critical rectangle features selected and trained by an AdaBoost learning algorithm, and then detected the initial position, size, and incline angle of a human face correctly. Once a human face is reliably detected, they extract the colour distributions of the face and the upper body from the detected facial regions and the upper body regions for creating respective robust colour modelling by virtue of k-means clustering and multiple Gaussian models. Then fast and efficient multi-view face tracking is executed using several critical features and a simple linear Kalman filter. However, two critical problems, lighting condition change and the number of clusters in the k-means clustering, are not solved well yet. In addition to the Kalman filter, some real-time face tracking systems based on particle filtering techniques were proposed (Fahn et al., 2009; Fahn & Lin, 2010; Fahn & Lo, 2010). The researchers utilized a particle filter to localize human faces in image sequences. Since they have considered the hair colour information of a human head, it will keep tracking even if the person is back to the sight of a camera.

In this chapter, an automatic real-time multiple faces recognition and tracking system installed on a person following robot is presented, which is inclusive of face detection, face recognition, and face tracking procedures. To identify human faces quickly and accurately, an AdaBoost algorithm is used for training a strong classifier for face detection (Viola & Jones, 2001). As to face recognition, we modify the discriminative common vectors (DCVs) algorithm (Cevikalp & Wilkes, 2004; Cevikalp et al., 2005; Gulmezoglu et al., 2001) and employ the minimum Euclidean distance to measure the similarity of a detected face image and a candidate one. To raise the confidence level, the most likely person is determined by the majority voting of ten successive recognition results from a face image sequence. In the sequel, the results of recognition will be assigned into two classes: "master" and "stranger."

In our system, the robot will track the master unceasingly except that the power is turned off. In the face tracking procedure, we propose an improved particle filter to dynamically locate multiple faces. According to the position of the target in an image, we issue a series of commands (moving forward, turning left or turning right) to drive the motors of wheels on the robot, and evaluate the distance between a master and the robot by means of a laser range finder to issue a set of commands (stop or turn backward) until the robot follows to a suitable distance in front of the master.

2. Hardware description

The following describes the hardware system of our experimental robot whose frame size is 40 cm long, 40 cm wide, and 130 cm high as Fig. 1 shows. The robot has three wheels; the motors of two front wheels driving the robot to move forward/backward and turn left/right, whereas one rear wheel without dynamic power is used for supporting the robot only. By controlling the turning directions of the two front wheels, we can change the moving direction of the robot. In order to help the robot to balance, we settle one ball caster in the rear.

PTZ camera

Touch screen

Lithium power

Industrial PC

12V DC battery

Laser range finder

Robot wheel

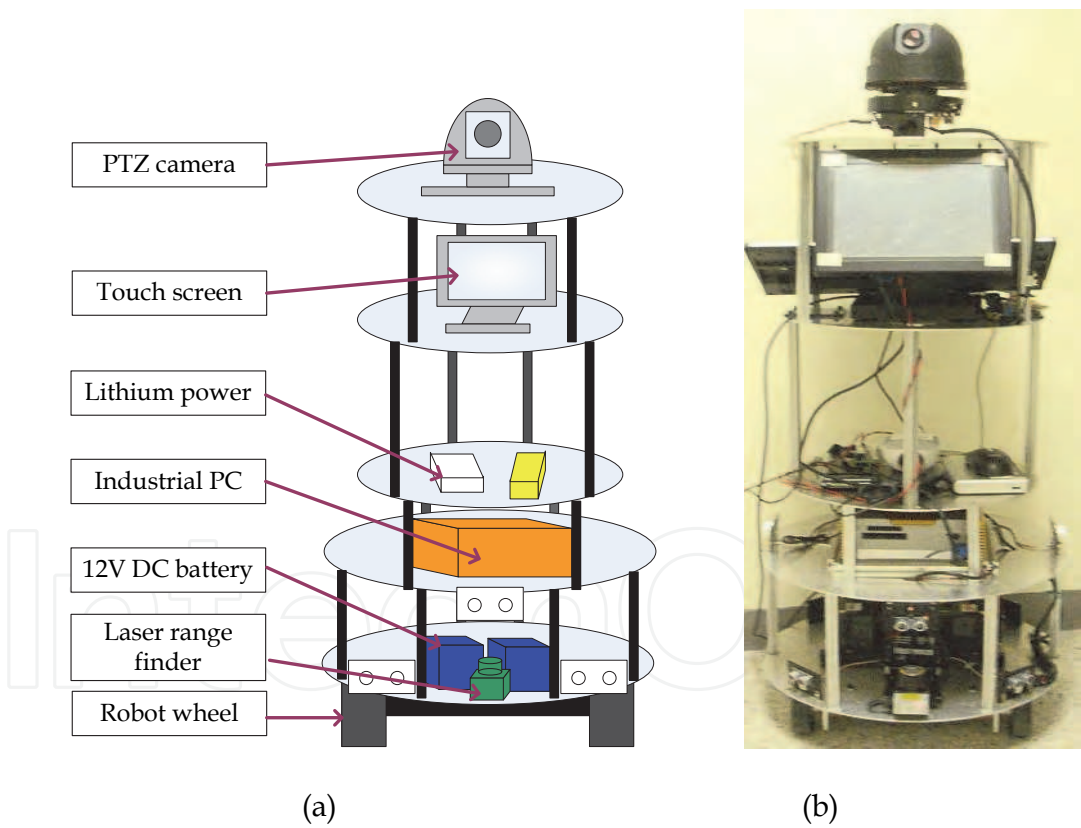(a)   (b)

Fig. 1. Illustration of our experimental robot: (a) the robot frame; (b) the concrete robot.

Fig. 2 illustrates the hardware architecture of our experimental robot. We can receive the distance data of the laser range finder from the Industrial PC via RS-232. In the camera system, the images are acquired through the USB 2.0 video grabber. After processing the captured images and receiving the distance data from the laser range finder, the program will send commands through RS-232 to the motor system. The H8/3694F microprocessor

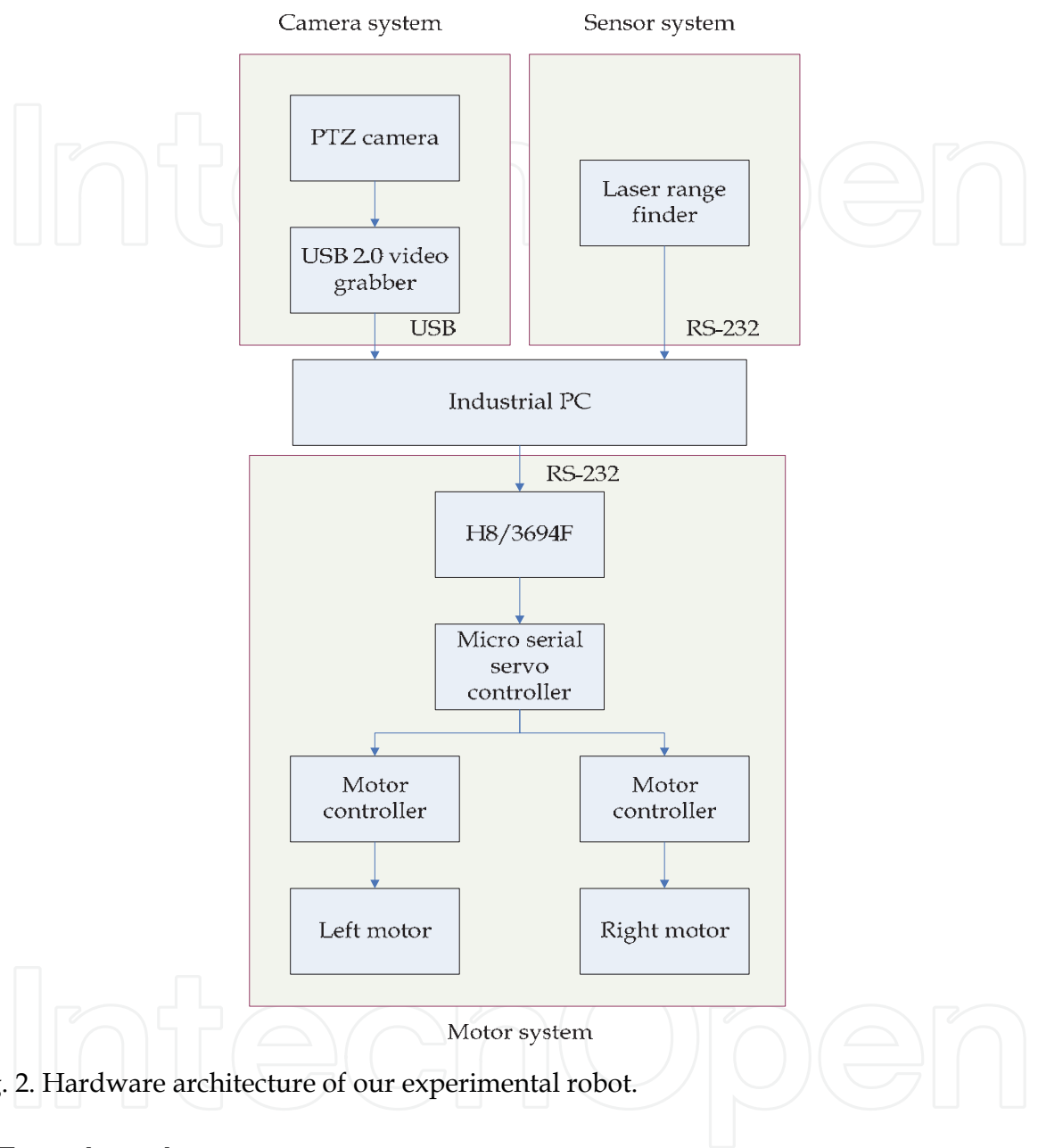can link the micro serial servo controller, and then the left and right motors of the front wheels are driven.

Fig. 2. Hardware architecture of our experimental robot.

3. Face detection

The face detection is a crucial step of the human face recognition and tracking system. For enabling the system to extract facial features more efficiently and perfectly, we must narrow the range of face detection first, and the performance of the face detection method can't be too low. In order to detect human faces quickly and accurately, we take advantage of an AdaBoost (Adaptive Boosting) algorithm to build a strong classifier with simple rectangle features involved in an integral image (Viola & Jones, 2001). Therefore, no matter what the size of a rectangle feature we use, the execution time is always constant. Some rectangle features are fed to the strong classifier that can distinguish positive and negative images.

Although the AdaBoost algorithm spends a lot of time on training the strong classifier, the face detection result attains high performance in the main. Fig. 3 depicts the boosting learning algorithm that we adopt for face detection.

Given: n training images (x_1, y_1), (x_2, y_2), …, and (x_n, y_n), where $y_i = 0, 1$ stand for negative and positive examples, respectively.

Initialize weights:

$$\begin{cases} w_{1,i} = \dfrac{1}{2m} & \text{if } m \text{ is the number of negative examples} \\ w_{1,i} = \dfrac{1}{2l} & \text{if } l \text{ is the number of positive examples.} \end{cases}$$

For $t = 1, 2, …, T$**:**

1. Normalize the weights:

$$w_{t,i} = \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}.$$

2. Compute the weighted error of each weak classifier:

$$\varepsilon_t = \min_{f,p,\theta} \sum_i w_i |h(x_i, f, p, \theta) - y_i|.$$

3. Choose the weak classifier that has the minimum error:

$$h_t(x) = h(x, f_t, p_t, \theta_t).$$

4. Update the weights:

$$\begin{cases} w_{t+1,i} = w_{t,i} \beta_t & \text{if } x \text{ is classified correctly} \\ w_{t+1,i} = w_{t,i} & \text{otherwise,} \end{cases}$$

where $\beta_t = \dfrac{\varepsilon_t}{1 - \varepsilon_t}.$

The final strong classifier is

$$C(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \dfrac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise,} \end{cases}$$

where $\alpha_t = \log \dfrac{1}{\beta_t}.$

Fig. 3. The boosting learning algorithm used in face detection.

The cascaded structure of weak classifiers using the AdaBoost algorithm is constructed one by one, and each weak classifier has its own order. A positive result will be processed by the next weak classifier, whereas a negative result at any currently processed point leads to the immediate rejection for the corresponding sub-window. First, we input the rectangle features via different sub-windows into many weak classifiers, and the detection error rate is the minimum in the first weak classifier that can delete a lot of negative examples, then the remaining images are more difficult to be removed, which are further processed by the successive weak classifiers. Such an operation is repeatedly executed until the last weak classifier is performed, and the remainders are face images.

In order to take advantage of the integral image, we adopt the concept like processing a pyramid of images. That is, a sub-window constituting the detector scans the input image on many scales. For example, the first detector is composed of 20×20 pixels, and an image of 320×240 pixels is scanned by the detector. After that, the current image is smaller than the previous one by 1.25 times. A fixed scale detector is then employed to scan each of these images. When the detector scans an image, subsequent locations are acquired from shifting the window by some number of pixels, $s \times \Delta$, where s is the current scale. Note that the choice of Δ affects both the speed of the detector and the accuracy of the detection. Detection results are clustered into a class to determine the representative position of the target via comparing the distance between the centres of a detection result and the class with a given threshold. The threshold changes with the current scale like shifting pixels mentioned above. If more than one detection result overlaps with each other, the final region is computed from the average of their positions of each overlapping part.

4. Face recognition

Through the face detection procedure, we can take face images that are then fed to the face recognition procedure. To begin with, we execute the image normalization to make the sizes of face images be the same. The intensity of the images will be also adjusted for reducing the lighting effect. After the size normalization and intensity adjustment, we subsequently perform the feature extraction process to obtain the feature vector of a face image. The idea of common vectors was originally introduced for isolated word recognition problems in the case that the number of samples in each class is less than or equal to the dimensionality of the sample space. The approaches to solving these problems extract the common properties of classes in the training set by eliminating the differences of the samples in each class. A common vector for each individual class is obtained from removing all the features that are in the directions of the eigenvectors corresponding to the nonzero eigenvalues of the scatter matrix of its own class. The common vectors are then used for pattern recognition. In our case, instead of employing a given class's own scatter matrix, we exploit the within-class scatter matrix of all classes to get the common vectors. We also present an alternative algorithm based on the subspace method and the Gram-Schmidt orthogonalization procedure to acquire the common vectors. Therefore, a new set of vectors called the discriminative common vectors (DCVs) will be used for classification, which results from the common vectors. What follows elaborates the algorithms for obtaining the common vectors and the discriminative common vectors (Gulmezoglu et al., 2001)

4.1 The DCV algorithm

Let the training set be composed of C classes, where each class contains M samples, and let x_m^c be a d-dimensional column vector which denotes the m-th sample from the c-th

class. There will be a total of $N = MC$ samples in the training set. Supposing that $d > M - C$, three scatter matrices S_W, S_B, and S_T are respectively defined below:

$$S_W = \sum_{c=1}^{C} \sum_{m=1}^{M} (x_m^c - \overline{x}^c)(x_m^c - \overline{x}^c)^T, \tag{1}$$

$$S_B = \sum_{c=1}^{C} (\overline{x}^c - \overline{x})(\overline{x}^c - \overline{x})^T, \tag{2}$$

and
$$S_T = \sum_{c=1}^{C} \sum_{m=1}^{M} (x_m^c - \overline{x})(x_m^c - \overline{x})^T = S_W + S_B, \tag{3}$$

where $\overline{x}$ is the mean of all samples and $\overline{x}^c$ is the mean of samples in the c-th class.

In the special case, $w^T S_W w = 0$ and $w^T S_B w \neq 0$ for all $w \in R^d - 0_d$, modified Fisher's linear discriminant criterion attains a maximum. However, a projection vector w, satisfying the above conditions, does not necessarily maximize the between-class scatter matrix. The following is a better criterion (Bing et al., 2002; Turk & Pentland, 1991):

$$F(W_{out}) = \underset{|W^T S_W W| = 0}{\arg\max} |W^T S_B W| = \underset{|W^T S_W W| = 0}{\arg\max} |W^T S_T W|. \tag{4}$$

To find the optimal projection vectors w in the null space of S_W, we project the face samples onto the null space of S_W and then obtain the projection vectors by performing principal component analysis (PCA). To accomplish this, the vectors that span the null space of S_W must first be computed. Nevertheless, this task is computationally intractable since the dimension of this null space is probably very large. A more efficient way of realizing this task is resorted to the orthogonal complement of the null space of S_W, which significantly becomes a lower-dimensional space.

Let R^d be the original sample space, V be the range space of S_W, and $V^\perp$ be the null space of S_W. Equivalently,

$$V = span\{\alpha_k | S_W \alpha_k \neq 0, \ k = 1, 2, \ldots, r\} \tag{5}$$

and
$$V^\perp = span\{\alpha_k | S_W \alpha_k = 0, \ k = r+1, r+2, \ldots, d\}, \tag{6}$$

where $r < d$ is the rank of S_W, $\{\alpha_1, \alpha_2, \ldots, \alpha_d\}$ is an orthonormal set, and $\{\alpha_1, \alpha_2, \ldots, \alpha_r\}$ is the set of orthonormal eigenvectors corresponding to the nonzero eigenvalues of S_W.

Consider the matrices $Q = \begin{bmatrix} \alpha_1 & \alpha_2 \ldots \alpha_r \end{bmatrix}$ and $\overline{Q} = \begin{bmatrix} \alpha_{r+1} & \alpha_{r+2} \ldots \alpha_d \end{bmatrix}$. Since $R^d = V \oplus V^\perp$, every face image $x_m^c \in R^d$ has a unique decomposition of the following form

$$x_m^c = y_m^c + z_m^c, \tag{7}$$

where $y_m^c = Px_m^c = QQ^Tx_m^c \in V, z_m^c = \bar{P}x_m^c = \bar{Q}\bar{Q}^Tx_m^c \in V^\perp$, and P and $\bar{P}$ are the orthogonal projection operators onto V and $V^\perp$, respectively. Our goal is to compute

$$z_m^c = x_m^c - y_m^c = x_m^c - Px_m^c. \tag{8}$$

To do this, we need to find a basis in V, which can be accomplished by an eigenanalysis in S_W. In particular, the normalized eigenvectors α_k corresponding to the nonzero eigenvalues of S_W will be an orthonormal basis in V. The eigenvectors can be obtained by calculating the eigenvectors of the smaller $N \times N$ matrix defined as $S_W = A^TA$ where A is a $d \times N$ matrix of the form depicted below:

$$A = \left[x_1^1 - \bar{x}^1 \cdots x_M^1 - \bar{x}^1 \quad x_1^2 - \bar{x}^2 \cdots x_M^C - \bar{x}^C \right]. \tag{9}$$

Let λ_k and v_k be the k-th nonzero eigenvalue and the corresponding eigenvector of A^TA, where $k < N - C$. Then $\alpha_k = Av_k$ will be the orthonormal eigenvector that corresponds to the k-th nonzero eigenvalue of S_W. The sought-for projection onto $V^\perp$ is achieved using Eq. (8). In this manner, it turns out that we obtain the same unique vector for all samples of a class as follows:

$$x_{com}^c = x_m^c - QQ^Tx_m^c = \bar{Q}\bar{Q}^Tx_m^c, \quad m = 1, 2, \ldots, M, \quad c = 1, 2, \ldots, C. \tag{10}$$

That is, the vector on the right-hand side of Eq. (10) is independent of the sample indexed m. We refer to the vectors x_{com}^c as the common vectors.

The theorem states that it is enough to project a single sample from each class. This will greatly reduce the computational load of the calculations. After acquiring the common vectors x_{com}^c, the optimal projection vectors will be those that maximize the total scatter of the common vectors:

$$F(W_{out}) = \underset{|W^TS_WW|=0}{\arg\max} |W^TS_BW| = \underset{|W^TS_WW|=0}{\arg\max} |W^TS_TW|$$
$$= \underset{W}{\arg\max} |W^TS_{com}W|, \tag{11}$$

where W is a matrix whose columns are the orthonormal optimal projection vectors w_k, and S_{com} is the scatter matrix of the common vectors:

$$S_{com} = \sum_{c=1}^{C} (x_{com}^c - \bar{x}^{com})(x_{com}^c - \bar{x}^{com})^T, \quad c = 1, 2, \ldots, C, \tag{12}$$

where $\bar{x}^{com}$ is the mean of all common vectors:

$$\overline{x}^{com} = \frac{1}{C}\sum_{c=1}^{C} x_{com}^{c}.$$ (13)

In this case, the optimal projection vectors w_k can be found by an eigenanalysis in S_{com}. Particularly, all eigenvectors corresponding to the nonzero eigenvalues of S_{com} will be the optimal projection vectors. S_{com} is typically a large $d \times d$ matrix. Thus, we can use the smaller $C \times C$ matrix $A_{com}^{T} A_{com}$ to find nonzero eigenvalues and the corresponding eigenvectors of $S_{com} = A_{com} A_{com}^{T}$, where A_{com} is the $d \times C$ matrix of the form expressed as

$$A_{com} = \left[x_{com}^{1} - \overline{x}_{com} \cdots x_{com}^{C} - \overline{x}_{com} \right].$$ (14)

There will be $C-1$ optimal projection vectors since the rank of S_{com} is $C-1$ if all common vectors are linearly independent. If two common vectors are identical, then the two classes, which are represented by this vector, cannot be distinguished from each other. Since the optimal projection vectors w_k belong to the null space of S_W, it follows that when the image samples x_m^c of the i-th class are projected onto the linear span of the projection vectors w_k, the feature vector $\Omega_c = \left[\left\langle x_m^c, w_1 \right\rangle \cdots \left\langle x_m^c, w_{C-1} \right\rangle \right]^T$ of the projection coefficients $\left\langle x_m^c, w_k \right\rangle$ will also be independent of the sample indexed m. Therefore, we have

$$\Omega_c = W^T x_m^c, \ m = 1, 2, \ldots, M, \ c = 1, 2, \ldots, C.$$ (15)

We call the feature vectors Ω_c discriminative common vectors, and they will be used for the classification of face images. To recognize a test image x_{test}, its feature vector is found by

$$\Omega_{test} = W^T x_{test},$$ (16)

which is then compared with the discriminative common vector Ω_c of each class using the Euclidean distance. The discriminative common vector found to be the closest to Ω_{test} is adopted to identify the test image.

Since Ω_{test} is only compared to a single vector for each class, the classification is very efficient for real-time face recognition tasks. In the Eigenface, Fisherface, and Direct-LDA methods, the test sample feature vector Ω_{test} is typically compared to all feature vectors of samples in the training set. It makes these methods be impractical for real-time applications for large training sets. To overcome this, they should solve the small sample size problem alternatively (Chen et al., 2000; Huang et al., 2002).

The above method can be summarized as follows:

Step 1. Compute nonzero eigenvalues and their corresponding eigenvectors of s_w using the matrix $A^T A$, where $S_W = A A^T$ and A is given by Eq. (9). Set matrix $Q = \left[\alpha_1 \ \alpha_2 \ldots \alpha_r \right]$, where $\alpha_k, k = 1, 2, \ldots, r$ are the orthonormal eigenvectors of S_W with rank r.

Step 2. Choose any sample from each class and project it onto the null space of S_W to obtain the common vectors:

$$x_{com}^c = x_m^c - QQ^T x_m^c, \quad m = 1, 2, \ldots, M, \quad c = 1, 2, \ldots, C. \tag{17}$$

Step 3. Compute the eigenvectors w_k of s_{com}, associated with the nonzero eigenvalues, by use of the matrix $A_{com}^T A_{com}$, where $s_{com} = A_{com} A_{com}^T$ and A_{com} is given in Eq. (14). There are at most $C - 1$ eigenvectors corresponding to the nonzero eigenvalues to constitute the projection matrix $W = \begin{bmatrix} w_1 & w_2 \ldots w_{C-1} \end{bmatrix}$, which will be employed to obtain feature vectors as shown in Eqs. (15) and (16).

4.2 Similarity measurement

There exist a lot of methods about similarity measurement. In order to implement face recognition on the robot in real time, we select the Euclidean distance to measure the similarity of a face image to those in the face database after feature extraction with the aid of the DCV transformations.

Let the training set be composed of C classes, where each class contains M samples, and let y_m^c be a d-dimensional column vector which denotes the m-th sample from the c-th class. If y is the feature vector of a test image, the similarity of the test image and a given sample is defined as

$$d(y, y_m^c) = \left\| y - y_m^c \right\|. \tag{18}$$

And the Euclidean distance between the feature vectors of the test image y and class c is

$$d(y, c) = \min \left\{ d(y, y_1^c), d(y, y_2^c), \ldots, d(y, y_m^c) \right\}. \tag{19}$$

After the test image compared with the feature vectors in all training models, we can find out the class whose Euclidean distance is minimum. In other words, the test image is identified the most possible person, namely Id_y. Note that a range of the threshold T_c is prescribed for every class to eliminate the person who does not belong to the face database as Eq. (20) shows:

$$Id_y = \begin{cases} \arg\min_c d(y, c) & \text{if } \min\left\{ d(y, c) \leq T_c, \ c = 1, 2, \ldots, C \right\} \\ \text{NULL} & \text{otherwise.} \end{cases} \tag{20}$$

5. Face tracking

Our detection method can effectively find face regions, and the recognition method can know the master for people in front of the robot. Then the robot will track the master using our face tracking system. On the other hand, we can exchange the roles of the master with a stranger, and the robot will resume its tracking to follow the stranger. Until now, many tracking algorithms have been proposed by researchers. Of them, the Kalman

filter and particle filter are applied extensively. The former is often used to predict the state sequence of a target for linear predictions, and it is not suitable for the non-linear and non-Gaussian movement of objects. However, the latter is based on the Monte Carlo integration method and suit to nonlinear predictions. Considering that the postures of a human body are nonlinear motions, our system chooses a particle filter to realize the tracker (Foresti et al., 2003; Montemerlo et al., 2002; Nummiaro et al., 2003). The key idea of this technique is to represent probability densities by sets of samples. Its structure is divided into four major parts: probability distribution, dynamic model, measurement, and factored sampling.

5.1 Particle filter

Each interested target will be defined before the face tracking procedure operates. The feature vector of the i-th interested face in an elliptical shape in the t-th frame is denoted as follows:

$$F_{t,i} = \left\{ x_{t,i}, y_{t,i}, w_{t,i}, h_{t,i}, Id_{t,i} \right\}, \tag{21}$$

where $(x_{t,i}, y_{t,i})$ is the centre of the i-th elliptical window in the t-th frame (i.e., at time step t), $w_{t,i}$ and $h_{t,i}$ symbolize the minor and major axes of the ellipse, respectively, and $Id_{t,i}$ is person's identity assigned from the face recognition result.

According to the above target parameters, the tracking system builds some face models and their associated candidate models, each of which is represented by a particle. Herein, we select the Bhattacharyya coefficient to measure the similarity of two discrete distributions resulting from the respective cues of a face model and its candidate models existing in two consecutive frames.

Three kinds of cues are considered in the observation model; that is, the colour cue, edge cue, and motion cue. The colour cue employed in the observation model is still generated from the practice presented in the above literatures, instead of modelling the target by a colour histogram. The raw image of each frame is first converted to a colour probability distribution image via the face colour model, and then the samples are measured on the colour probability distribution image. In this manner, the face colour model is adapted to find interested regions by the face detection procedure, next handle the possible changes of the face regions due to variable illumination frame-by-frame, and the Bhattacharyya coefficient is used to calculate the similarity defined as

$$M_{C_{t,i}} = BC\left(C_{face_{t-1,i}}(l,m), C_{face_{t,i}}(l,m) \right), \tag{22}$$

where $C_{face_{t-1,i}}(l,m)$ and $C_{face_{t,i}}(l,m)$ mean the discrete distributions of the colour cues of the i-th face model and its candidate model at time steps $t-1$ and t, respectively.

In comparison to our earlier proposed methods (Fahn et al., 2009; Fahn & Lin, 2010; Fahn & Lo, 2010), the edge cue is a newly used clue. Edges are a basic feature in images, and they are not more influenced than the colour model suffering from illumination. First, the Sobel filter is applied to obtain all the edges in each frame, and the Bhattacharyya coefficient is employed to represent the similarity measurement just like the colour model:

$$M_{E_{t,i}} = BC\left(E_{face_{t-1,i}}(l,m), E_{face_{t,i}}(l,m)\right), \tag{23}$$

where $E_{face_{t-1,i}}(l,m)$ and $E_{face_{t,i}}(l,m)$ stand for the discrete distributions of the edge cues of the i-th face model and its candidate model at time steps $t-1$ and t, respectively.

As for the motion feature, our tracking system records the centre positions of interested objects in the previous num frames continued, say twenty. Both the distance and direction are then computed in the face tracking procedure for each interested object acting as a particle. At time step t, the average moving distance $Adis_{t,i}$ of the i-th particle (i.e., a candidate model) referring to its average centre positions $Ax_{t,i}$ and $Ay_{t,i}$ in the previous num frames are expressed below:

$$Ax_{t,i} = \frac{\sum_{k=1}^{num} x_{t-k,i}}{num}, \tag{24}$$

$$Ay_{t,i} = \frac{\sum_{k=1}^{num} y_{t-k,i}}{num}, \tag{25}$$

and
$$Adis_{t,i} = \frac{\sum_{k=1}^{num} \sqrt{(x_{t,i} - x_{t-k,i})^2 + (y_{t,i} - y_{t-k,i})^2}}{num}. \tag{26}$$

There are two states of the motion, including the accelerated velocity and decelerated velocity. If the interested face moves with the accelerated velocity, the distance between the current centre position and the average centre position will be larger than the average moving distance. If the interested face moves with the decelerated velocity, conversely, the distance between the current centre position and the average centre position will be less than the average moving distance.

In addition to the distance of face motion, the direction of face motion is another important factor. The angle $\theta_{t,i}$ symbolizing the direction of the i-th particle at time step t is expressed as

$$\theta_{t,i} = \frac{\left(\tan^{-1} \frac{y_{t,i} - Ay_{t,i}}{x_{t,i} - Ax_{t,i}}\right) \times 180}{\pi}. \tag{27}$$

It also depends on the average centre positions of the i-th particle in the previous num frames. All the scope of angles ranged from $-90°$ to $+90°$ is partitioned into four orientations, each of which covers the angle of $45°$. In order to get rid of an irregular face motion trajectory, we consider the most possibility of the direction by virtue of the majority voting on the number of times of the corresponding orientation of $\theta_{t-k,i}$, $k = 1, 2, \ldots, num$. Thus, the likelihood of face motion at time step t is determined by the following equation:

$$M_{v_{t,i}} = \begin{cases} \dfrac{dis_{t,i}}{Adis_{t,i}} \times 0.25 + vote_{t,i} \times 0.2 + 0.6 & \text{if } dis_{t,i} \leq Adis_{t,i} \\[4mm] \dfrac{dis_{t,i}}{Adis_{t,i}} \times 0.35 + vote_{t,i} \times 0.2 + 0.2 & \text{otherwise,} \end{cases} \tag{28}$$

where $dis_{t,i}$ is the distance between the centre position in the current frame and the average centre position in the previous num frames, $Adis_{t,i}$ is referred to Eq. (26), and $vote_{t,i}$ is the maximum frequency of the four orientations appearing in the previous num frames. Therefore, the likelihood of the motion feature combines both the measurements of the moving distance and moving direction.

When the occlusion of human faces happens, the motion cue is obvious and becomes more reliable. On the contrary, when the occlusion does not happen, we can weigh particles mainly by the observation of the colour and edge cues. As a result, we take a linear combination of colour, edge, and motion cues as follows:

$$M_{t,i} = (1 - \alpha_t)\beta_t M_{c_{t,i}} + \alpha_t \beta_t M_{E_{t,i}} + (1 - \beta_t)M_{v_{t,i}}, \tag{29}$$

where $0 \leq \alpha_t, \beta_t \leq 1$, and $\alpha_t + \beta_t = 1$. Fig. 4 shows the iteration procedure for tracking the i-th interested face using our improved particle filter.

Given: a particle set $S_{t-1,i} = \left\{ \left(s_{t-1,i}^{(j)}, \pi_{t-1,i}^{(j)} \right) \middle| j = 1,2,...,N \right\}$, a target $(s_{t-1,i}^{\otimes}, \pi_{t-1,i}^{\otimes}) \in S_{t-1,i}$, and its centre positions $x_{t-k,i}$ and $y_{t-k,i}$, $k = 1, 2, ..., num$ at time step $t-1$, perform the following steps:

Propagation: fluctuate each particle $s_{t-1,i}^{(j)}$ with the weight $\pi_{t-1,i}^{(j)}$ by a dynamic model to obtain the particle set $S_{t,i}$.

Observation: weigh the particles using colour, edge, and motion cues as:
1. Calculate the distance between each particle and the target by
   $Dis_{t,i}^{(j)} = \sqrt{1 - M_{t,i}^{(j)}}$, where $M_{t,i}^{(j)}$ is a linear combination of the three cues.
2. Weigh each particle with
   $\pi_{t,i}^{(j)} = \dfrac{1}{\sqrt{2\pi}\sigma} e^{-\frac{Dis_{t,i}^{(j)2}}{2\sigma^2}}$, where σ is the standard deviation of a Gaussian distribution.

Selection: check the weights of particles. If the particle $s_{t,i}^{(j)}$ with $\pi_{t,i}^{(j)}$ exceeds or equals a threshold, then duplicate $K_{t,i}^{(j)}$ even-weighted particles from it; otherwise, ignore the particle.

Estimate: acquire the new target $s_{t,i}^{\otimes}$ with the largest weight as:
$\pi_{t,i}^{\otimes} = \pi_{t,i}^{*(k)}$ for $k = \arg \max\limits_{k} \pi_{t,i}^{*(k)}$ and its centre positions $x_{t,i}^{\otimes}$ and $y_{t,i}^{\otimes}$.

Fig. 4. The iteration procedure for tracking the i-th interested face by the improved particle filter.

After the current filtering process is finished, we execute the next filtering process again until a local search is achieved. Such a method will track faces steadily even if humans move very fast. In the first filtering process, we establish thirty particles to track faces. But, in the subsequent filtering processes, only ten particles are set up to facilitate the face tracking operation.

5.2 Occlusion handling

Occlusion handling is a major problem in visual surveillance (Foresti et al., 2003). The occlusion problem usually occurs in a multi-target system. When multiple moving objects occlude each other, both the colour and edge information are more unreliable than the velocities and directions of moving objects. In order to solve the occlusion problem, we have improved the strategies of the filtering process and modified the weights of the particles depicted in the previous subsection. Compared with the earlier proposed methods (Fahn et al., 2009; Fahn & Lin, 2010; Fahn & Lo, 2010), the major improvement is that the directions of moving objects in a number of the past frames are incorporated into the constituents of the motion cue. We will check whether the occlusion ceases to exist. If only one human face is detected in the face detection procedure, then the occlusion will not occur in this condition. Nevertheless, if two and more faces are detected, the occlusion will occur when moving faces are near to each other, and the system will go to the occlusion handling mode.

5.3 Robot control

We can predict the newest possible position of a human face by means of the particle filtering technique. According to the possible position, we issue a control command to make the robot track the i-th face continuously. Such robot control comprises two parts: PTZ camera control and wheel motor control. The following is the command set of robot control described at time step t, where the first two commands belong to the PTZ camera control and the remaining ones used for the wheel motor control:

$$S_{Command_{t,i}} = \left\{ up_{t,i}, down_{t,i}, forward_{t,i}, stop_{t,i}, backward_{t,i}, left_{t,i}, right_{t,i} \right\}. \tag{30}$$

5.3.1 PTZ camera control

The principal task of the PTZ camera control is to keep the i-th face appearing in the centre of the screen. It prevents the target from situating on the upper or lower side of the scope of the screen. If $y_{t,i}$ is lower than or equal to 50, then we assign the command $up_{t,i}$ to the PTZ camera at time step t; if $y_{t,i}$ is greater than or equal to 190, then we assign the command $down_{t,i}$ to the PTZ camera at time step t, as shown in Eq. (31):

$$Command_{t,i} = \begin{cases} up_{t,i} & \text{if } y_{t,i} \leq 50 \\ down_{t,i} & \text{if } y_{t,i} \geq 190 \\ \text{no-operation} & \text{otherwise,} \end{cases} \tag{31}$$

where $y_{t,i}$ is the centre of the i-th elliptic window in the y-direction at time step t.

5.3.2 Wheel motor control

The main task of the wheel motor control is to direct the robot to track the i-th face continuously. We utilize two kinds of information to determine which command is issued. The first information is the position of the face appearing in the screen. According to this, we assign the commands to activate the wheel motors that make the robot move appropriately. If $x_{t,i}$ is lower than or equal to 100, then we assign the command $left_{t,i}$ to the wheel motors at time step t; if $x_{t,i}$ is greater than or equal to 220, then we assign the command $right_{t,i}$ to the wheel motors at time step t as Eq. (32) shows:

$$Command_{t,i} = \begin{cases} left_{t,i} & \text{if } x_{t,i} \leq 100 \\ right_{t,i} & \text{if } x_{t,i} \geq 220 \\ \text{no-operation} & \text{otherwise,} \end{cases} \quad (32)$$

where $x_{t,i}$ is the centre of the i-th elliptic window in the x-direction at time step t.

The second information is the laser range data. We equip a laser range finder on the front of the robot to measure the distance $D_{t,i}$ between the robot and a target which is allowed to have the incline angle ranged from $-10°$ to $+10°$. In accordance with the distance $D_{t,i}$, we can assign the commands to the wheel motors to control the movement of the robot. If $D_{t,i}$ is lower than or equal to 60, then we assign the command $backward_{t,i}$ to the wheel motors at time step t; if $D_{t,i}$ is greater than or equal to 120, then we assign the command $forward_{t,i}$ to the wheel motors at time step t; otherwise, we assign the command $stop_{t,i}$ to the wheel motors at time step t, as stated in Eq. (33):

$$Command_{t,i} = \begin{cases} backward_{t,i} & \text{if } D_{t,i} \leq 60 \\ forward_{t,i} & \text{if } D_{t,i} \geq 120 \\ stop_{t,i} & \text{otherwise.} \end{cases} \quad (33)$$

Through controlling the robot, we can increase its functionality of interacting with people. If the robot can recognize humans' identities and follow him/her abidingly, then the relation between the robot and human being is definitely closer to each other, and it can be more extensive to use. So, we control the movement of the robot on the condition of state changes between the face tracking and face recognition modes. After face detection and target confirmation, the system will switch to the recognition mode and start the robot to execute a certain task.

6. Experimental results

Table 1 list both the hardware and software that are used for the development of the real-time face tracking and recognition system installed on our robot. All the experimental results presented in this section were obtained from taking an average of the outcome data for different 10 times to demonstrate the effectiveness of the proposed methods.

Hardware	Software
Host computer: Intel Core 2 Duo CPU T2300 2.0GHz with 1.0 GB RAM PTZ camera: Sony product FCB-EX480C Microprocessor: H8/3694F	Operating system: Microsoft Windows XP SP3 equipped with DirectX 9.0 Developing tool: Borland C++ Builder 6.0 H8/3694F Editor Software

Table 1. The Developing Environment of Our Face Tracking and Recognition System Installed on the Robot

6.1 Face detection

The training database for face detection consists of 861 face images segmented by hand and 4,320 non-face images labelled by a random process from a set of photographs taken in outdoors. The sizes of these face and non-face images are all 20×20 pixels. Fig. 5 illustrates some examples of positive and negative images used for training. Notice that each positive image contains a face region beneath the eyebrows and beyond the half-way between the mouth and the chin.
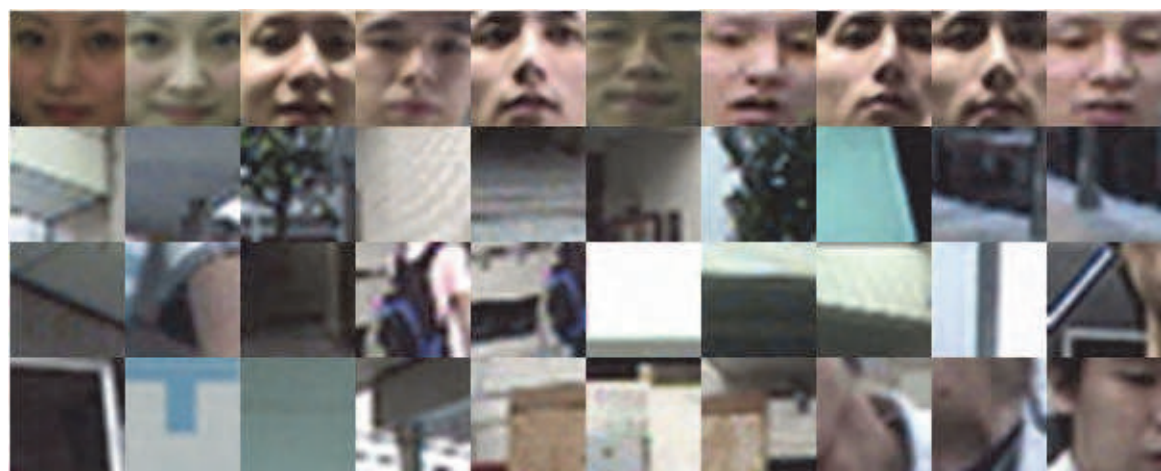
Fig. 5. Some examples of positive and negative images used for training. The first row is face images and the others are non-face images.

We perform the experiments on face detection in two different kinds of image sequences: ordinary and jumble. The image sequence is regarded as a jumble if the background in this sample video is cluttered; that is, the environment is possessed of many pixels like skin colours and/or the illumination is not appropriate. Then we define the "error rate" as a probability of detecting human faces incorrectly; for example, regarding an inhuman face as a human face. And the "miss rate" is a probability which a target appears in the frame but not detected by the system. Table 2 shows the face detection rates for the above experiments. From the experimental results, we can observe that the ordinary image sequences have better correct rates, because they encounter less interference, and the error rates are lower in this situation. The performance on the jumble image sequences is inferior to that on the ordinary ones. The main influence is due to the difference of luminance, but the effect of the colour factor is comparatively slight, particularly for many skin-like regions existing in the

background. It is noted that the error rate of the jumble image sequences is larger than that of the ordinary ones, because our database has a small number of negative examples. That will cause the error rate raise. Fig. 6 demonstrates some face detection results from using our real-time face detection system in different environments.

Measurement Experiment	The number of faces	The number of detected faces	Detection rate	Error rate	Miss rate
Ordinary image sequence	50	48	96.00%	2.00%	2.00%
	100	94	94.00%	2.00%	4.00%
	150	142	94.67%	0.67%	4.67%
	200	187	93.50%	1.00%	5.50%
Jumble image sequence	50	45	90.00%	6.00%	4.00%
	100	91	91.00%	5.00%	4.00%
	150	134	89.33%	6.00%	4.67%
	200	181	90.50%	6.50%	3.00%

Table 2. The Face Detection Rates of Two Different Kinds of Image Sequences

Fig. 6. Some face detection results from using our real-time face detection system.

6.2 Face recognition

Our face image database comprises 5 subjects to be recognized, and each of them has 20 samples. The nicknames are Craig, Eric, Keng-Li, Jane, and Eva (3 males and 2 females) respectively. The facial expressions (open or closed eyes and smiling or non-smiling) are also varied. The images of one subject included in the database are shown in Fig. 7.

Fig. 7. Images of one subject included in our database.

Subject / Person	Craig	Eric	Keng-Li	Jane	Eva
Craig	0~210	374~461	420~513	207~351	238~491
Eric	381~437	0~185	483~526	525~637	448~664
Keng-Li	372~450	348~493	0~95	366~513	418~622
Jane	215~429	524~618	342~572	0~226	259~373
Eva	225~411	459~586	409~597	276~356	0~100

Table 3. The Range of Thresholds for Each Class to Eliminate the Corresponding Person from the Possible Subjects

We have our robot execute face recognition in real time using the DCV method. After comparing the feature vector of the captured and then normalized face image with those of all training models, we can find out the class whose Euclidean distance is the minimum. In other words, it is identified as the most possible person. Table 3 presents the range of thresholds for each class to eliminate the corresponding person from the possible subjects included in the database. If the feature similarity of a captured and normalized face image is lower than or equal to the threshold value for every class, the corresponding person is assigned to one of classes. For every 10 frames per a period, we rank the frequency of assignments for each class to determine the recognition result. Fig. 8 graphically shows some face recognition results from using our real-time face recognition system installed on the robot.
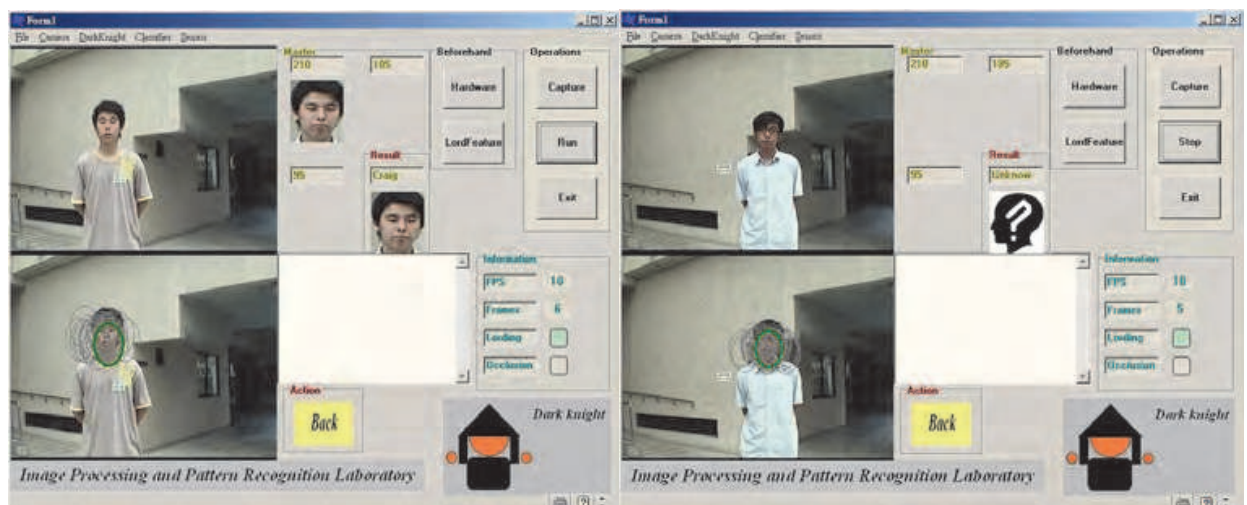

Fig. 8. Some face recognition results from using our real-time face recognition system installed on the robot.

6.3 Face tracking

Several different types of accuracy measures are required to evaluate the performance of tracking results. In this subsection, we utilize track cardinality measures for estimating the tracking rates. The reference data should be first determined in order to obtain any of the accuracy measures. Let A be the total number of actual tracks as the reference data, R be the total number of reported tracks, A' be the number of actual tracks that have corresponding reported tracks, and R' be the number of reported tracks that do not

correspond to true tracks. The measures are determined by m_c, the fraction of true tracks, f_c, the fraction of reported tracks that are false alarms where they do not correspond to true tracks, and the miss rate is m_d. The following defines these variables:

$$m_c = \frac{A'}{A}, \tag{34}$$

$$f_c = \frac{R'}{A}, \tag{35}$$

and

$$m_d = 1 - \frac{R}{A}. \tag{36}$$

The types of our testing image sequences are roughly classified into three kinds: a single face (Type A), multiple faces without occlusion (Type B), and multiple faces with occlusion (Type C). Each type has three different image sequences, and we will perform five tests on each of them. In these experiments, we use 30 particles to track human faces in each frame for the first filtering process, followed by 10 particles. Then we analyze the performance of face tracking on the three kinds of testing image sequences according to the aforementioned evaluation methods. Table 4 shows the face tracking performance for the above experiments.

Estimate / Experiment	A	R	A'	R'	f_c (%)	m_d (%)	m_c (%)
Type A	100	99	98	1	1.00	1.00	98.00
	150	147	146	1	0.67	2.00	97.33
	200	198	196	2	1.00	1.00	98.00
	250	248	246	2	0.80	0.80	98.40
Type B	100	99	98	1	1.00	1.00	98.00
	150	148	147	1	0.67	1.33	98.00
	200	198	195	3	1.50	1.00	97.50
	250	247	245	2	0.80	1.20	98.00
Type C	100	98	83	15	15.00	2.00	83.00
	150	146	124	23	15.33	2.67	82.00
	200	194	167	27	13.50	3.00	83.50
	250	241	205	36	14.40	3.60	82.00

Table 4. The Face Tracking Performance of Three Different Kinds of Image Sequences

From the experimental results, the tracking accuracy of Type C is lower than those of the other two types. This is because when the occlusion happens, the particles are predicted on the other faces that are also the colour regions. On the other hand, the correct rates of face tracking for Types A and B are similar to each other; that is, the tracking performance for a single face is almost the same to that for multiple faces without occlusion. When the robot and the target move simultaneously, some objects whose colours are close to the skin colour may exist in the background. In this situation, we must consider the speed of the target and assume that the target disappearing on the left side of the image does not appear on the right side immediately. Through simple distance limits, the robot can track the target

continuously. Moreover, at regular intervals we will either abandon the currently tracked targets then intend to detect the face regions once more, or keep tracking on the wrong targets until the head moves near the resampling range of the particle filter. Attempting to enlarge the resampling range can slightly conquer this problem. However, the wider the resampling range is, the sparser the particles are. It has good outcomes for the robot to track faces, but sometimes it will reduce the tracking correctness. Fig. 9 shows some face tracking results from using our real-time face tracking system equipped on the robot.

Fig. 9. Some face tracking results from using our real-time face tracking system for: (a) a single human; (b) multiple humans within the field of view of the PTZ camera equipped on the robot.

7. Conclusion and feature works

In this chapter, we present a completely automatic real-time multiple faces recognition and tracking system installed on a robot that can capture an image sequence from a PTZ camera, then use the face detection technique to locate face positions, and identify the detected faces as the master or strangers, subsequently track a target and guide the robot near to the target continuously. Such a system not only allows robots to interact with human being adequately, but also can make robots react more like mankind.

Some future works are worth investigating to attain better performance. In the face recognition procedure, if the background is too cluttered to capture a clear foreground, the recognition rate will decrease. Because most of our previous training samples were captured in a simple environment, sometimes static objects in the uncomplicated background are identified as the foreground. We can increase some special training samples in a cluttered background to lower the miss rate during the face detection. Of course, it will raise the face recognition accuracy, but need a lot of experiments to collect special and proper training samples.

8. References

K. H. An, D. H. Yoo, S. U. Jung, and M. J. Chung, "Robust multi-view face tracking," in Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Alberta, Canada, pp. 1905-1910, 2005.

Y. Bing, J. Lianfu, and C. Ping, "A new LDA-based method for face recognition," in Proceedings of the International Conference on Pattern Recognition, Quebec, Canada, vol. 1, pp. 168-171, 2002.

R. Brunelli and T. Poggio, "Face recognition: features versus templates," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 15, no. 10, pp. 1042-1052, 1993.

H. Cevikalp, M. Neamtu, M. Wilkes, and A. Barkana, "Discriminative common vectors for face recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 1, pp. 4-13, 2005.

H. Cevikalp and M. Wilkes, "Face recognition by using discriminative common vectors," in Proceedings of the International Conference on Pattern Recognition, Cambridge, United Kingdom, vol. 1, pp. 326-329, 2004.

L. F. Chen, H. Y. M. Liao, M. T. Ko, J. C. Lin, and G. J. Yu, "A new LDA-based face recognition system which can solve the small sample size problem," Pattern Recognition, vol. 33, no. 10, pp. 1713-1726, 2000.

M. J. Er, S. Wu, J. Lu, and H. L. Toh, "Face recognition with radial basis function (RBF) neural networks," IEEE Transactions on Neural Networks, vol. 13, no. 3, pp. 697-710, 2002.

C. S. Fahn, M. J. Kuo, and K. Y. Wang, "Real-time face tracking and recognition based on particle filtering and AdaBoosting techniques," in Proceedings of the 13th International Conference on Human-Computer Interaction, LNCS 5611, San Diego, California, pp.198-207, 2009.

C. S. Fahn and Y. T. Lin, "Real-time face detection and tracking techniques used for the interaction between humans and robots," in Proceedings of the 5th IEEE Conference on Industrial Electronics and Applications, Taichung, Taiwan, 2010.

C. S. Fahn and C. S. Lo, "A high-definition human face tracking system using the fusion of omnidirectional and PTZ cameras mounted on a mobile robot," in Proceedings of

the 5th IEEE Conference on Industrial Electronics and Applications , Taichung, Taiwan, 2010.

K. C. Fan, Y. K. Wang, and B. F. Chen, "Introduction of tracking algorithms," Image and Recognition, vol. 8, no. 4, pp. 17-30, 2002.

G. L. Foresti, C. Micheloni, L. Snidaro, and C. Marchiol, "Face detection for visual surveillance," in Proceedings of the 12th IEEE International Conference on Image Analysis and Processing, Mantova, Italy, pp. 115-120, 2003.

M. B. Gulmezoglu, V. Dzhafarov, and A. Barkana, "The common vector approach and its relation to principal component analysis," IEEE Transactions on Speech and Audio Processing, vol. 9, no. 6, pp. 655-662, 2001.

E. Hjelmås and B. K. Low, "Face detection: a survey," Computer Vision and Image Understanding, vol. 83, no. 3, pp. 236-274, 2001.

R. Huang, Q. Liu, H. Lu, and S. Ma, "Solving the small size problem of LDA," in Proceedings of the International Conference on Pattern Recognition, Quebec, Canada, vol. 3, pp. 29-32, 2002.

S. Z. Li and J. Lu, "Face recognition using the nearest feature line method," IEEE Transactions on Neural Networks, vol. 10, no. 2, pp. 439-443, 1999.

M. Montemerlo, S. Thrun, and W. Whittaker, "Conditional particle filters for simultaneous mobile robot localization and people-tracking," in Proceedings of the IEEE International Conference on Robotics and Automation, Washington, D.C., vol. 1, pp. 695-701, 2002.

K. Nummiaro, E. Koller-Meier, and L. Van Gool, "An adaptive color-based particle filter," Image and Vision Computing, vol. 21, no. 1, pp. 99-110, 2003.

M. Turk and A. Pentland, "Face recognition using eigenfaces," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Maui, Hawaii, pp. 586-591, 1991.

P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, vol. 1, pp. 511-518, 2001.

W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: a literature survey," ACM Computing Surveys, vol. 35, no. 4, pp. 399-458, 2003.
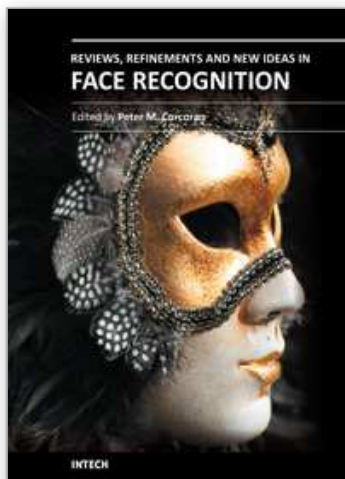
Reviews, Refinements and New Ideas in Face Recognition

Edited by Dr. Peter Corcoran

As a baby one of our earliest stimuli is that of human faces. We rapidly learn to identify, characterize and eventually distinguish those who are near and dear to us. We accept face recognition later as an everyday ability. We realize the complexity of the underlying problem only when we attempt to duplicate this skill in a computer vision system. This book is arranged around a number of clustered themes covering different aspects of face recognition. The first section on Statistical Face Models and Classifiers presents reviews and refinements of some well-known statistical models. The next section presents two articles exploring the use of Infrared imaging techniques and is followed by few articles devoted to refinements of classical methods. New approaches to improve the robustness of face analysis techniques are followed by two articles dealing with real-time challenges in video sequences. A final article explores human perceptual issues of face recognition.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Chin-Shyurng Fahn and Chih-Hsin Wang (2011). Real-time Multi-Face Recognition and Tracking Techniques Used for the Interaction between Humans and Robots, Reviews, Refinements and New Ideas in Face Recognition, Dr. Peter Corcoran (Ed.), ISBN: 978-953-307-368-2, InTech, Available from: http://www.intechopen.com/books/reviews-refinements-and-new-ideas-in-face-recognition/real-time-multi-face-recognition-and-tracking-techniques-used-for-the-interaction-between-humans-and

INTECH
open science | open minds