

# Intelligent CCTV via a Planetary Sensor Network

Ting Shan<sup>†</sup>, Brian C. Lovell<sup>†</sup> and Shaokang Chen  
*Intelligent Real-Time Imaging and Sensing Group*  
*EMI, School of ITEE, The University of Queensland*  
*Australia 4072*  
and

<sup>†</sup>*National Information and Communications Technology Australia (NICTA)*  
*{shanting, lovell, shaokang}@itee.uq.edu.au*

## ***Abstract***

*Intelligent Closed-Circuit Television (CCTV) has attracted worldwide attention and government interest since such systems were recently used to such great effect to track the movements of the four suicide bombers in the days before the attack on the London Underground in July 2005. A major reason for the rapid results obtained from the British CCTV installations is their recent conversion from a group of independent analog video systems to a network of integrated IP-connected cameras with central digital video storage. Here we explore some of the technical issues to be solved to build the natural descendant of today's integrated metropolitan CCTV systems to become a planetary-wide sensor network incorporating both fixed, mobile, and nomadic video sensors. In particular, we describe recent research by the authors to provide reliable person location and recognition services on a planetary sensor network.*

## **1. Introduction**

The Prime Minister of Australia, John Howard, returned from London after the July 2005 suicide bombings and said (Howard J 2005), “The biggest thing that I have learnt by a country mile out of my visit, particularly to Britain, is the extraordinary value of surveillance cameras.” The integrated security camera system deployed in Britain was primarily installed to reduce the incidence of assault and property damage. Yet it was able to help identify and discover the movements of the four suicide bombers within just 24 hours of the bombing. The effectiveness of this particular security system comes about because of the saturation coverage of the British people by CCTV. In Britain, CCTV systems cover cities, public transport and motorways, while in many other countries the coverage is quite haphazard. It was public demand for security in public places that led to this pervasiveness. Moreover, the adoption of centralised digital video databases, largely to reduce management and

monitoring costs, has also resulted in an extraordinary co-ordination of the CCTV resources.

It is therefore natural to consider the power and usefulness of a distributed CCTV system which could be extended not only to cover a city, but also to include virtually all video and still cameras on the planet. Such a system should not only include public CCTV systems in rail stations and city streets, but should also have the potential to include private CCTV systems in shopping malls and office buildings. With the advent of third generation (3G) wireless technology, there is no reason, in principle, that we could not include security cameras feeds from moving public spaces such as taxis, buses, and trains. There should also be the possibility of including the largest and cheapest potential source of image and video feeds which are those available from private mobile phone handsets with cameras. Many newer 3G handsets have both location service (GPS) and video capability, so the location of a phone could be determined and the video and image stream could be integrated into the views provided by the rest of the fixed sensor network.

Another reason to investigate the ad-hoc integration of video and images from the mobile phone network into a planetary sensor network comes from a current project of the authors to use mobile smart phones as a low-cost secure medical triage system in the event of natural disasters. In 2005, a phone-based medical triage system being developed jointly by a commercial partner and the University of Queensland was used by medical officers in major natural disaster areas (ABC News 2005) in the aftermath of 1) the tsunami in Banda Aceh, Indonesia, 2) Hurricane Katrina in the USA, and 3) the earthquake in Kashmir, Pakistan. During these trials the need for the delivery of person location services based on robust face recognition through the mobile phone network became apparent. For example such a service could have proved invaluable to quickly reunite families and help determine the identities of missing persons. In major natural disasters, millions of people may be displaced and housed in temporary shelters, as was indeed the case after hurricane Katrina devastated New Orleans. In such extreme disasters is extremely difficult to rapidly determine who has survived and where they are physically located.

## **2. Related Work**

In addition to our mobile phone based medical triage system, a possible testbed for intelligent CCTV is the emerging experimental planetary scale sensor web, IrisNet (Gibbons P B, Karp B, Ke Y, Nath S and Sehan S 2003). IrisNet uses internet connected desktop PCs and inexpensive, off-the-shelf sensors such as webcams, microphones, temperature, and motion sensors deployed globally to provide a wide-area sensor network. IrisNet is deployed as a service on PlanetLab ([www.planet-lab.org](http://www.planet-lab.org)), a worldwide collaborative network environment for prototyping next generation internet services initiated by Intel Research and Princeton University.

Gibbons *et al.* (Gibbons P B, Karp B, Ke Y, Nath S and Sehan S 2003) envisage a worldwide sensor web in which many users can query, as a single unit, vast quantities of data from thousands or even millions of planetary sensors. IrisNet stores its sensor derived data in a distributed XML schema which is well-suited to describing such hierarchical data as it employs self-describing tags. Indeed the robust distributed nature of the database can be most readily compared to the structure of the internet DNS naming service.

Wide area person recognition and location services are a valuable application that could be deployed on IrisNet. Apart from the obvious use of the technology for public security by law enforcement officers, a case can be made for access by the general public as well. For example, a mother who has lost her child in, say, a shopping mall could simply upload a photograph of her child from the image store in her mobile phone and the system would efficiently look for the child in an ever-widening geographic search space until contact was made. Clearly in the case of IrisNet, there is no possibility of humans being employed to identify all the faces captured by the planetary sensor web to support the search, so the task must be fully automated. Such a service raises inevitable privacy concerns which must be addressed, but the service also has the potential for public good as in this example of reuniting a worried mother with her lost child.

Now we will focus on some of the crucial technologies underpinning such intelligent CCTV services — automatically detecting and recognizing faces in image and video databases.

### **3. Robust Face Recognition**

In order to build a robust face recognition system suitable for deployment on multiple unmatched camera sensors with uncooperative subjects, we need to fulfill the four key requirements of accuracy, robustness, scalability, and speed. Our system has three major components comprising: 1) a Viola-Jones (Viola P and Jones M 2001) face detection module based on cascaded simple binary features to rapidly detect and locate multiple faces from the input still image or video sequences, 2) view-based Active Appearance Models (AAMs) (Cootes T F and Taylor C J 1996) to estimate facial pose and compensate for extreme pose angles, and 3) Adaptive Principal Component Analysis (Chen S K and Lovell B 2004) to recognize faces since the method is robust to poor lighting and extreme facial expressions (Fig. x.1).

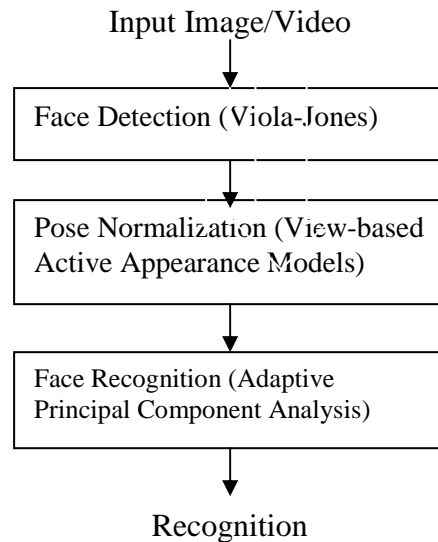


Fig. x.1 The framework of our robust face recognition system

### 3.1 Face Detection

Face detection is a challenging task that has attracted much attention in recent years. It is a necessary first-step in a face recognition system to locate a face or faces from cluttered backgrounds. It also can be used in diverse areas such as human-computer interaction, content-based image retrieval, and intelligent surveillance. Techniques for face detection can be divided into three categories: feature-based approaches, template matching, and image-based approaches (Hjelmas E and Low B K 2001).

Feature-based approaches such as using edges (Govindaraju V 1996 and Huang J, Gutta S and Wechsler H 1996), skin color (Lee C H, Kim J S, and Park K H 1996), motion (McKenna S, Gong S and Liddell H 1995) etc, are suitable for real-time systems due to their fast feature extraction, but they suffer from low detection rates. Edge detection is the first step in edge representation. After detection, these detected edges need to be labelled and matched to a face model. Facial features such as pupils, lips and eyebrows are normally darker than the regions around them. This property can be exploited to detect facial parts (Hoogenboom R and Lew M 1996, Wong C, Kortenkamp D and Speich M 1995) and also the face itself (Lv X G, Zhou J and Zhang C S 2000). A number of colour models have been used, including RGB (Sato S, Nakamura Y and Kanade T 1999), normalized RGB (Sun Q B, Huang W M and Wu J K 1998), HSI (Lee C H, Kim J S and Park K H 1996), YIQ (Dai Y and Nakano Y 1996), YES (Saber E and Tekalp A M 1998), YUV (Abdel-Mottaleb M and Elgammal A 1999). Naturally such colour models are ineffective if the light is not white or the camera is monochrome.

Two main template matching approaches are used. The first approach is “feature searching” (Jeng S H, Liao H Y M, Liu Y T and Chen M Y 1998) based on relative

positions of facial features. This technique first detects prominent facial features, and then uses knowledge of face geometry to verify the existence of a face by searching for the less prominent facial features. Eyes are most commonly used due to their unique appearance. The second major approach is using various deformable face models, such as snakes (Gunn S R and Nixon M S 1994, Nikolaidis A and Pitas I 2000, Yokoyama, Yagi Y and Yachida M 1998], deformable templates (Yuille A L, Hallinan P W and Cohen D S (1992): Feature extraction from faces using deformable templates. In: International Journal of Computer Vision, Volume 8, Issue 2, pp: 99-111 ), and point distributed models (Cootes T F and Taylor C J 1992, Cootes T F and Taylor C J 1996).

Image-based approaches treat face detection as a pattern recognition problem and avoid using face knowledge directly. The central idea is to use supervised learning to train a face/non-face classifier. Various statistical methods have been used including Eigenfaces (Turk M and Pentland A 1991, Sung K K and Poggio T 1998, Yang M H, Ahuja N and Kriegman D 2000), neural networks (Rowley H A, Baluja S and Kanade T 1998, Feraud R, Bernier O and Collobert D 1997, Roth D, Yang M H, and Ahuja N 2000), and support vector machines (SVMs) (Osuna E, Freund R and Girosi F 1997, Terrillon J, Shirazi M, Fukamachi H and Akamatsu S 2000). These techniques can generally achieve good performance, but most of them are computationally expensive and thus are not suitable for real-time applications.

In 2001, Viola and Jones (Viola P and Jones M 2001) proposed an image-based face detection system which can achieve remarkably good performance in real-time. The main idea of their method is to combine weak classifiers based on simple binary features which can be computed extremely quickly. Simple rectangular Haar-like features are extracted; face and non-face classification is done using a cascade of successively more complex classifiers which are trained by the AdaBoost learning algorithm. Our face detection module is based on the Viola-Jones approach using our own training sets.

### **3.1.1 Cascade Face Detector**

The cascade face detector uses a sequence of binary classifiers which discard non-face regions and only send likely face candidates to the next level of classifier. Thus it employs a “coarse-to-fine” strategy (Fleuret F and Geman D 2001). Simple classifiers can be constructed which reject the majority of non-face regions at the very early stage of detection, before the use of more complex classifiers with higher discriminative capability. In this way, the more discriminating but more complex classifiers concentrate their processing time on face-like regions as illustrated by Fig. x.2.

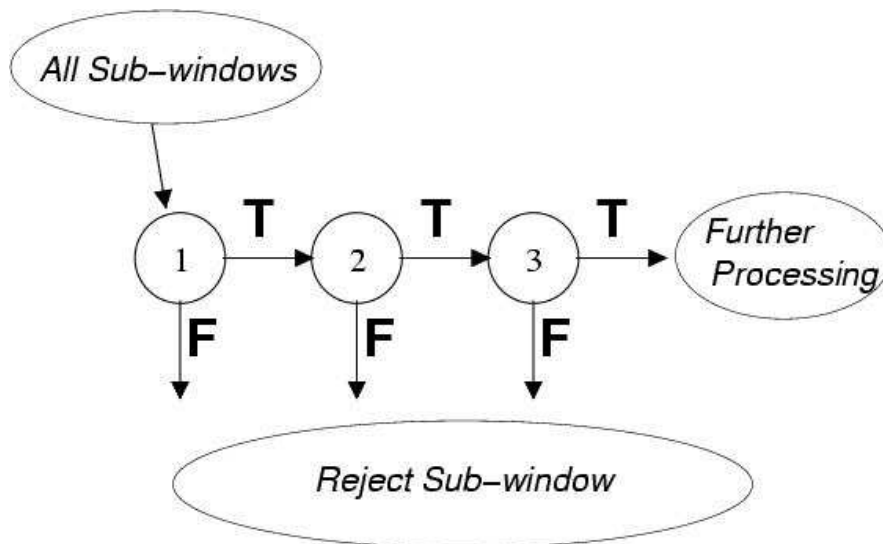


Fig. x.2: The cascade detection process

### 3.1.2 Adaboost Classifier

#### 3.1.2.1 Haar-like Wavelet Features

The Haar-like wavelet features (Mallat S G 1989) extracted from the image subwindows are an image representation method which characterises the texture similarities between different regions by computing the sum of pixel values in different regions.

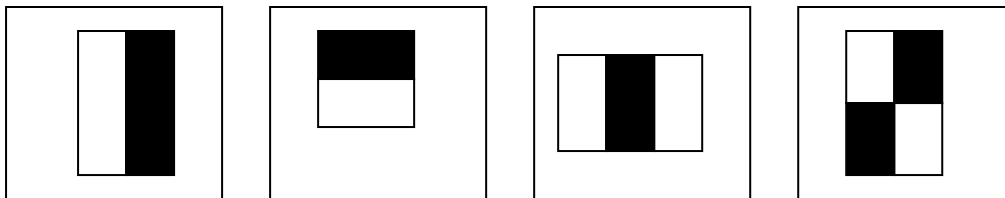


Fig. x.3: Four types of rectangle features defined in a sub-window. The value of the feature is the difference between the sums of pixels within the white and black rectangles.

The value of a two-rectangle feature is the difference between the sums of the pixel values within the two rectangular subregions. The value of a three-rectangle feature is the difference between the sums of pixel values within the two outside rectangles and the sum of pixels of the centre rectangle. A four-rectangle feature computes the difference between sums of pixel values in diagonal pairs of rectangles (Fig. x.3). Given a subwindow whose size is 24\*24 pixels, the exhaustive set of rectangular features is 116,300 (86,400 for the two-rectangle features, 27,6000 for the three-rectangle features, and 2,300 for the four-rectangle features), which is over-complete.

#### 3.1.2.2 Integral Image

The integral image, also known as the “summed-area table” (Crow F 1984) in the domain of computer graphics, can compute the Haar-like rectangular features very quickly. Placing the origin at the top left corner of the image, the value of the integral image at location  $(x, y)$ , denoted  $ii(x, y)$ , is calculated as the sum of the pixel values contained in the rectangular region bounded by the origin and  $(x, y)$ . The calculation of the integral image can be calculated efficiently by the recursion:

$$s(x, y) = s(x, y - 1) + i(x, y)$$

and

$$ii(x, y) = ii(x - 1, y) + s(x, y)$$

where  $s(x, y)$  is the sum of the column pixel values, with initialization values  $s(x, -1) = 0$ , and  $ii(-1, y) = 0$ . Thus we can calculate the integral image representation of the image in a single pass (Fig. x.4). The value of the sum of pixel values in any arbitrary rectangle region can be easily recovered from the integral image.

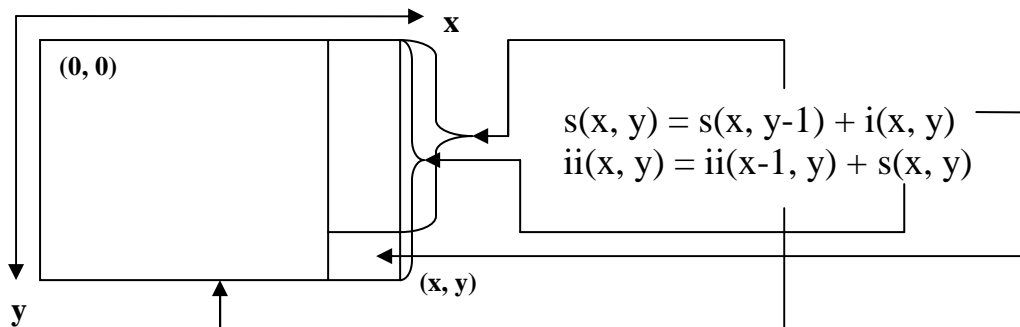


Fig. x.4: Representation of  $s(x, y)$  and  $ii(x, y)$

In Fig. x.5 the sum of pixel values in rectangle D can be computed efficiently by:

$$ii(4) + ii(1) - ii(2) - ii(3),$$

where  $ii(1)$  is the sum of pixel values in rectangle A,  $ii(2)$  is sum of pixel values in A and B,  $ii(3)$  is sum of pixel values in A and C, and  $ii(4)$  is the sum of pixel values in A, B, C and D.

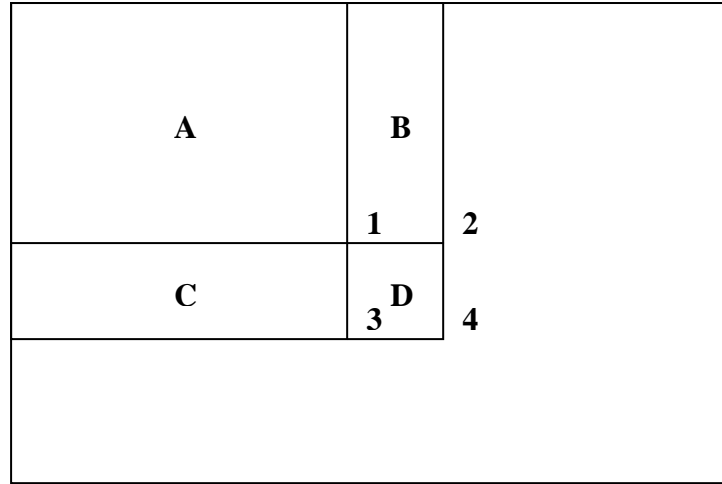


Fig. x.5: The sum of pixels in rectangle D can be computed by:  
 $ii(4) + ii(1) - ii(2) - ii(3)$

Similarly, to compute two, three and four rectangle features, we need only 6, 8 and 9 integral image values respectively.

### 3.1.2.3 Adaboost Learning Algorithm

The Adaboost learning algorithm is used to select the best rectangle features and linearly combine these features into a classifier. Adaboost is a boosting learning algorithm, which can fuse many weak classifiers into a single more precise classifier. The main idea of Adaboost is as follows. At the beginning of training, all training examples are assigned equal weight. During the process of boosting, the weak classifier with the lowest classification error is selected and the weights of the samples which are wrongly classified by the weak classifier increase. The final classifier is a linear combination of the weak classifiers of all rounds, where classifiers with lower classification error have a higher weight. Details of the learning algorithm can be seen in Table 1.

A weak classifier  $h_j$  contains a feature  $f_i$ , a threshold  $\theta_i$  and a direction  $\rho_i$

$$h_j = \begin{cases} 1 & \text{if } \rho_i f_i(x) < \rho_i \theta_i \\ 0 & \text{otherwise} \end{cases}$$

Here  $x$  is a 24\*24 pixel sub-window of an image.

- Given example images  $(x_1, y_1), \dots, (x_n, y_n)$  where the labels  $y_i = 0, 1$  for negative and positive examples respectively.
- Initialize weights to  $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$  for training example  $y_i = 0, 1$  respectively, where  $m$  and  $l$  are the number of negatives and positives respectively.
- For  $t = 1 \dots T$ 
  - 1) Normalize weights, so that  $w_t$  can be treated as a probability distribution
$$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^n w_{t,j}}$$
  - 2) For each feature  $j$  train a classifier  $h_j$  which is restricted to using a single feature. The error is evaluated with respect to  $w_t, \epsilon_j = \sum_i w_t |h_j(x_i) - y_i|$ .
  - 3) Chose the classifier  $h_j$  with lowest error  $\epsilon_t$ .
  - 4) Update weights according to:
$$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$$

where  $e_i = 0$  if example  $x_i$  is classified correctly, 1 otherwise, and  $\beta_t = \frac{\epsilon_t}{1 - \epsilon_t}$
- The final strong classifier is:
$$h(x) = \begin{cases} 1 & \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0 & \text{otherwise} \end{cases}$$

where  $\alpha_t = \log \frac{1}{\beta_t}$

Table1: The Adaboost learning algorithm. T hypotheses are constructed each using a single feature. The final hypothesis is a weighted linear combination of the T hypotheses where the weights are inversely proportional to the training errors.

### 3.1.3 Training and Detection Results

#### 3.1.4.1 Training Database

The face training database includes 4916 hand labelled faces downloaded from Peter Carbonetto's website (Carbonetto P 2005). The negative training data were randomly collected from the internet and do not contain human faces. Some example face images are shown on Fig. x.6.



Fig. x.6: Example of some face images used for training (Carbonetto P 2005)

#### 3.1.4.2 Structure of the Detector Cascade

After training, our final detector has 24 layers with a total of 2913 features. Some detection results by our implementation are shown in Fig. x.7.



Fig. x.7: Detection results on several photographs.

## 3.2 Pose Normalization

Pose normalization refers to compensating for the pitch, roll, and yaw motions of the head to allow for non-frontal viewing conditions. It acts as a bridge between the face detection and face recognition modules. Our face recognition system is exceptionally insensitive to illumination and facial expression changes and can attain very high recognition rate with frontal view face images.

### 3.2.1 In-Plane Rotation

Many facial features can be used to detect in-plane face rotation, including the eyes, mouth, and pupils. In our system, we train an eye localizer and rotate the face image to the eyes horizontal position by computing the angle between two eyes and horizontal baseline.

#### 3.2.1.1 Eye Localizer

Our eye localizer is trained using the same method with face detection module but different feature set due to the property of eye region. Feature set we used as shown in Fig. x.8.

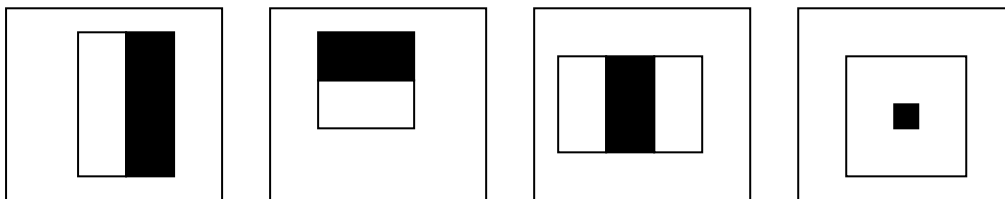


Fig. x.8: Features set used by eye localizer

We used the “BioID” (BioID 2005) face database which contains 1,521 face images (some of them wearing glasses). We manually cut two eye regions from every face image and rescaled them to 32\*16 to get 3,042 positive samples. Because in our system, the eye localization is always carried out after the face detection module, only on the face-like regions, we use the remaining part of the 1,521 face images as the negative training samples. Some eye training samples can be seen in Fig. x.9



Fig. x.9: Eye samples for training eye detector

The final eye localizer contains 36 stages and can process images very fast. It doesn't consume significant computation time as it only operates on the face-like regions whose size is relatively small compared to the whole image and faces are also a rare event in the video stream. Also, if the eye localizer can't locate eyes in the face candidate region, the candidate would be discarded as it is likely to have been wrongly detected. In other words, eye localizer acts as a verifier for the face detection module.

### 3.2.1.2 Rotation

After we locate the eyes in the face regions, we use the coordinates  $(x_{left}, y_{left})$ ,  $(x_{right}, y_{right})$  of the eyes to calculate the rotation angle  $\theta$  (Fig. x.10) by:

$$\theta = \arctan\left(\frac{y_{right} - y_{left}}{x_{right} - x_{left}}\right)$$

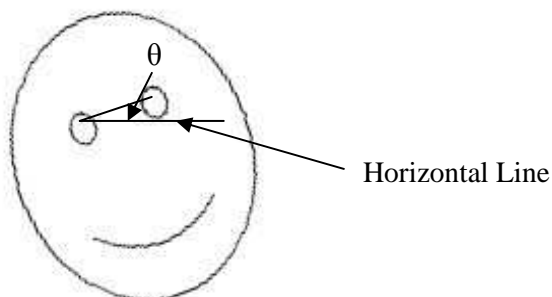


Fig. x.10: Rotation angle  $\theta$

Then we rotate the image to become a vertical frontal face image. A face-like region detected by the face detector and the rotated by the in-plane normalizer can be seen in Fig. x.11.



Fig. x.11: A face sample before and after in-plane image rotation

### 3.2.2 Out-of-plane Rotation

The out-of-plane rotation problem is much more complicated than in-plane rotation as it requires the construction of a 3D face model. Recently

Cootes *et al.* have shown that human faces from different view angles (from left profile to right profile) can be modeled by 3 distinct Active Appearance Models (AAMs) (Cootes T F, Walker L and Taylor C J 2000), these models can be used to estimate head pose and track faces by switching between the different models and synthesizing any new view of a face from a single view.

Motivated by Cootes, we propose a similar approach to solve our out-of-plane rotation problem. In our approach, we apply our face detector to a given input image to locate the initialization position before the AAM search is carried out. We only use one face model which can represent face pose variation from -45 degree to 45 degree horizontally and -30 degree to 30 degree vertically to compensate for the pose change problem. We argue that it is meaningless to attempt to recognize a person from a face image with too large a pose angle when we only know the person's frontal image — this is even a difficult task for human beings.

### 3.2.2.1 Building the Active Appearance Models

The Active Appearance Model is a powerful tool to describe deformable object images, it was originally introduced by Cootes and Taylor (Cootes T F and Taylor C J 2001). They demonstrate that a small number of 2D statistical models are sufficient to capture the shape and appearance of a face from any viewpoint (Cootes T F and Taylor C J 2001). The Active Appearance Model uses principal component analysis (PCA) on the linear subspaces to model both the shape and texture changes of a certain object class.

Given a collection of training images for a certain object class where the feature points have been manually marked, a shape and texture can be represented by applying PCA on the sample shape and texture distributions as:

$$x = \bar{x} + P_s c$$

and

$$g = \bar{g} + P_g c$$

where  $\bar{x}$  is the mean shape,  $\bar{g}$  is the mean texture and  $P_s$ ,  $P_g$  are matrices describing the respective shape and texture variations learned from the training sets, and the parameters,  $c$  are used to control the shape and texture change.

Cootes *et al* demonstrate that an Active Appearance Model trained on near frontal face images can handle pose change of up to 45 degree each side (Cootes T F and Taylor C J 2001). In our trials, we collected 20 frontal face images from Feret face database (Feret 2005). We first applied our face detector on these images, and then labeled each of them with 58 points around the main features including eyes, mouth, nose, eyebrows and chin. Some sample training face images can be seen in Fig. x.12.

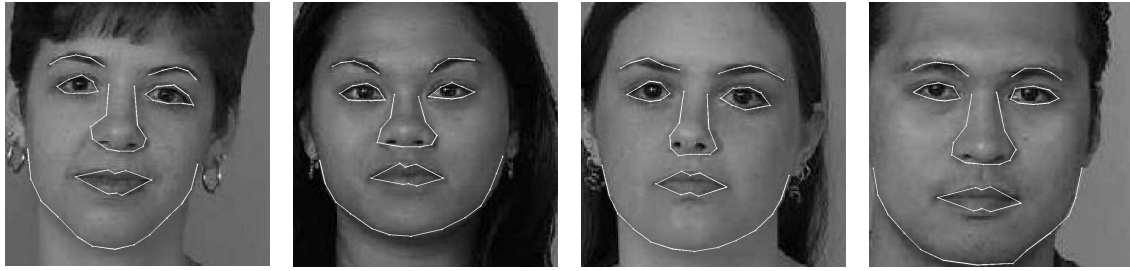


Fig. x.12: Sample training face images with labeled 58 points on main facial features

### 3.2.2.2 Combination of Face Detector with Active Appearance Model Search

The initialization of the Active Appearance Model Search is a problem since the original AAM search is a local optimization. Some failed AAM searches due to the poor initialization can be seen on Fig. x.13.

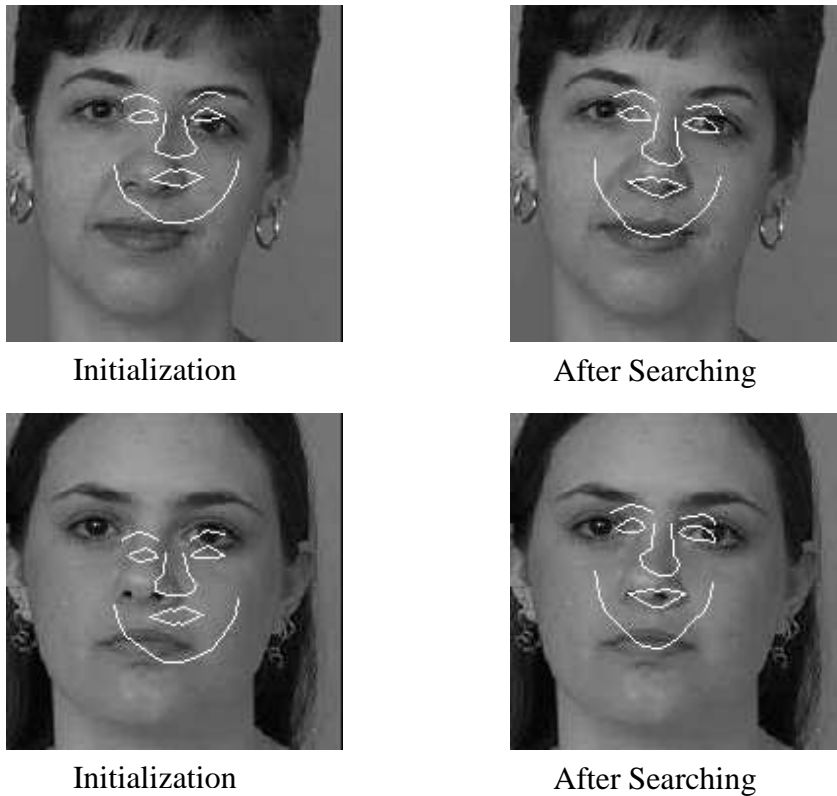


Fig. x.13: Failed AAM searches due to poor initialization.

We solve the initialization position problem by using our face detector to provide initialisation. The face detector finds the location of a human face in an input image and provides a good starting point for the subsequent AAM search which precisely marks the major facial features, (mouth, eyes, nose etc.). Some results from our combined face, eye detector and AAM search can be seen in Fig. x.14.

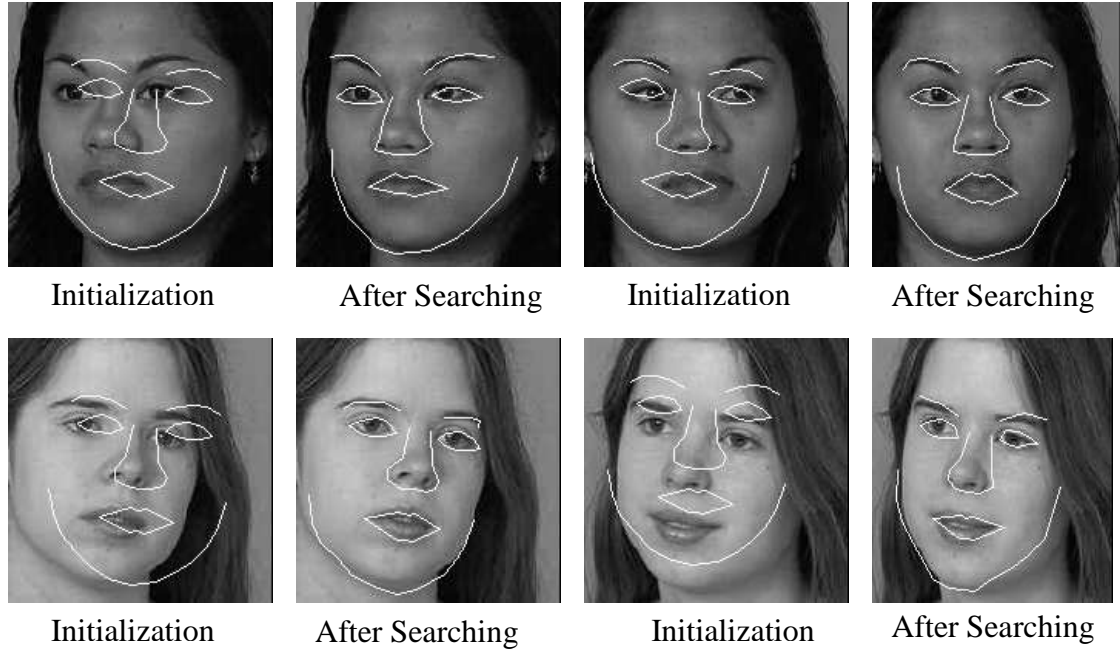


Fig. x.14: Some AAM search results on Feret Face Database

### 3.2.2.3 Predicting Pose

Here we follow the method of Cootes *et al* (Cootes T F, Walker L and Taylor C J 2000). They assume that the model parameters are related to the viewing angle,  $\theta$ , approximately by:

$$c = c_0 + c_c \cos(\theta) + c_s \sin(\theta)$$

where  $c_0$ ,  $c_c$  and  $c_s$  are vectors which are learned from the training data.

(Here we consider only head turning. Head nodding can be dealt with in a similar way).

Given a new face image with parameters  $\mathbf{c}$ , we can estimate orientation as follows. Let  $R_c^{-1}$  be the left pseudo-inverse of the matrix  $(c_c | c_s)$ , let  $(x_\alpha, y_\alpha)' = R_c^{-1}(c - c_0)$ , then the best estimate of the orientation is  $\tan^{-1}(y_\alpha / x_\alpha)$

### 3.2.2.4 Predicting Frontal View

After we estimate the angle  $\theta$ , we can use the model to synthesize new views, here we will synthesize a frontal view face image, which will be used for face recognition.

Let  $c_{res}$  be the residual vector not explained by the rotation model,

$$c_{res} = c - (c_0 + c_c \cos(\theta) + c_s \sin(\theta))$$

To reconstruct at a new angle,  $\alpha$ , we simply use the parameters:

$$c(\alpha) = c_0 + c_c \cos(\alpha) + c_s \sin(\alpha) + c_{res}$$

Here  $\alpha$  is 0, so the equation will be:

$$c(0) = c_0 + c_c + c_{res}$$

By changing parameter  $c$ , we can reconstruct the new frontal face image. Some synthesized results can be seen in Fig. x.15.

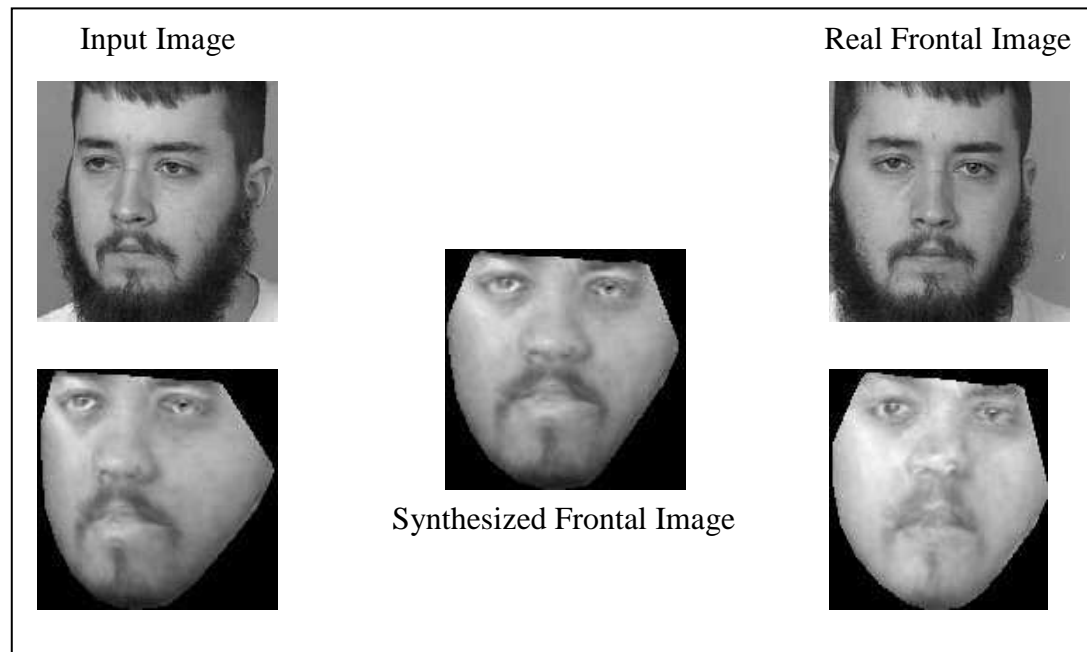


Fig. x.15: Some synthesized frontal face images from the Feret Face Database.

### 3.3 Face Recognition

Robust face recognition is a challenging goal because of the gross similarity of all human faces compared to large differences between face images of the same person due to variations in lighting conditions, view point, pose, age, health, and facial expression. Most systems work well only with images taken under constrained or laboratory conditions where lighting, pose, and camera parameters are strictly controlled

Recent research has been focused on diminishing the impact of nuisance factors on face recognition. Many approaches have been proposed for illumination invariant recognition (Yilmaz A and Gokmen M 2000, Gao Y S and Leung M K H 2002) and expression invariant recognition (Beymer D and Poggio T 1995, Black M J, Fleet D J and Yacoob Y 2000). But these methods suffer from the need to have large numbers of example images for training, which is often impossible in many situations when only few sample images are available such as in recognizing people from surveillance videos from a planetary sensor web or searching historic film archives.

In the last several years, research on face recognition has been focused on diminishing the impact of changes in lighting conditions, facial expression

and pose. Chen and Lovell (Chen S K and Lovell B 2004) developed Adaptive Principal Component Analysis (APCA) and Rotated APCA to compensate for illumination and facial expression variations.

### 3.3.1 Adaptive Principal Component Analysis

We first apply Principal Component Analysis (PCA) (Turk M and Pentland A 1991) for feature abstraction because of its good generalization capacity. Every face image can be projected into a subspace with reduced dimensionality to form an  $m$ -dimensional feature vector  $s_{j,k}$  with  $k = 1, 2, \dots, K_j$  denoting the  $k^{\text{th}}$  sample of the class  $S_j$ .

#### 3.3.1.1 Bayes Decision Rule

After constructing the face subspace for image representation, we need to warp this face space to enhance class separability. The Bayes classifier is the best classifier which achieves minimum error rate for pattern recognition if prior probabilities are known. The conditional density function is:

$$p(s | S_j) = \frac{\exp[-\frac{1}{2}(s - u_j)^T \text{cov}_j^{-1}(s - u_j)]}{(2\pi)^{\frac{m}{2}} |\text{cov}_j|^{-\frac{1}{2}}}$$

where  $u_j$  is the mean of class  $S_j$  and  $\text{cov}_j$  is the covariance matrix of  $S_j$ .

#### 5.3.1.2 Whitening and Eigenface Filtering

In order to compensate for the influence of between-class covariance on the estimation of pdf, we introduce a whitening power  $p$  to control the distribution, that is

$$\text{cov} = \text{diag}\{\lambda_1^{-2p}, \lambda_2^{-2p}, \dots, \lambda_m^{-2p}\},$$

where  $\lambda_i (i = [1 \dots m])$  are the eigenvalues extracted by PCA. Consequently, the whitening matrix  $Z$  is:

$$Z = \text{diag}\{\lambda_1^p, \lambda_2^p, \dots, \lambda_m^p\},$$

where the exponent  $p$  is determined empirically.

The aim of filtering is to enhance features that capture the main differences between classes (faces) while diminishing the contribution of those that are largely due to nuisance variations (within class differences) such as lighting. We thus define a filtering parameter  $\gamma$  which is related to identity-to-variation (ITV) ratio. The ITV is a ratio measuring the correlation of a change in person versus a change in variation for each of the eigenfaces. For an  $M$  class problem, assume that for each of the  $M$  classes (persons) we have examples under  $K$  standardized different lighting conditions. Let us denote the  $i^{\text{th}}$  element of the face vector of the  $k^{\text{th}}$  lighting sample for class (person)  $S_j$  by  $s_{i,j,k}$ . Then

$$ITV_i = \frac{\text{BetweenClassCo variance}}{\text{WithinClassCo variance}} = \frac{\frac{1}{M} \sum_{j=1}^M \frac{1}{K} \sum_{k=1}^K |s_{i,j,k} - \bar{\omega}_{i,k}|}{\frac{1}{M} \sum_{j=1}^M \frac{1}{K} \sum_{k=1}^K |s_{i,j,k} - \mu_{i,j}|}$$

$$\bar{\omega}_{i,k} = \frac{1}{M} \sum_{j=1}^M s_{i,j,k}$$

$$\mu_{i,j} = \frac{1}{K} \sum_{k=1}^K s_{i,j,k}$$

Here  $\bar{\omega}_{i,k}$  represents the  $i^{\text{th}}$  element of the mean face vector for lighting condition  $k$  for all persons and  $\mu_{i,j}$  represents the  $i^{\text{th}}$  element of the mean face vector for person  $j$  under all lighting conditions. We then define the filtering matrix  $\gamma$  by:

$$\gamma = \text{diag}\{ITV_1^q, ITV_2^q, \dots, ITV_m^q\},$$

where  $q$  is an exponential scaling factor determined empirically. After transformation, the conditional pdf is given by:

$$p(s | S_j) = \frac{\exp[-\frac{1}{2} \sum_{i=1}^m \frac{(s_i - \mu_{i,j})^2}{\lambda_i^{-2p} ITV_i^{-2q}}]}{(2\pi)^{\frac{m}{2}} \prod_{i=1}^m \lambda_i^{-p} ITV_i^{-q}}$$

and the distance  $d$  between two face vectors  $s_{j,k}$  and  $s_{j',k'}$  is defined by the Euclidean distance of their transformed vectors:

$$d_{jj',kk'} = \|Z\gamma(s_{j,k} - s_{j',k'})\|_2.$$

Therefore, our final transformation matrix is:

$$U' = Z\gamma V$$

where  $V$  is the set of eigenvectors extracted by PCA.

### 3.3.1.3 Cost function

The whitening matrix  $Z$  controls the overall scatter of all samples and tends to make the subspace isotropic, while the filtering parameter  $\gamma$  is designed to enhance the separability of classes and may stretch the space. There should be a trade off between these two effects. We use the following cost function which is a combination of error rate and the ratio of within-class distance to between-class distance and optimize empirically using an objective function defined by:

$$OPT = \sum_{j=1}^M \sum_{k=1}^K \sum_m \frac{d_{jj,k_0}}{d_{jm,k_0}}, \forall m \in d_{jm,k_0} < d_{jj,k_0}, m \in [1 \dots m],$$

where  $d_{jj,k_0}$  is the distance between the sample  $s_{j,k}$  and  $s_{j,0}$  which is the standard image reference for class  $S_j$ .

Fig. x.16 shows the large improvement in robustness to lighting angle. The proposed APCA method allows us to recognize faces with high confidence even if they are half in shadow.

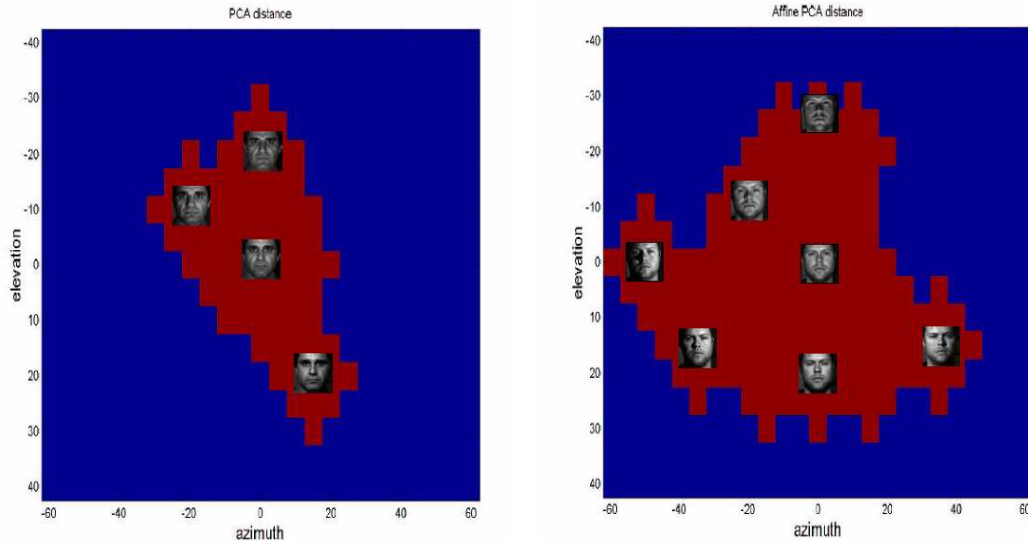


Fig. x.16: Contours of 95% recognition performance for the original PCA and the proposed APCA method against lighting elevation and azimuth.

### 3.3.2 Rotated APCA

We applied similar techniques to face images with variations in expression, but could not attain the levels of performance comparable to those obtained on illumination variant faces. This is because Eigenfeatures extracted by PCA from face images with illumination variation naturally cluster into two groups: 1) features strongly related to within-class covariance, and 2) features strongly related to between-class covariance. Usually the first three eigenfaces are strongly related to illumination (within-class) variation. Therefore, it is easy to find the eigenfeatures that represent within-class variation and suppress these with eigenfiltering. However, for expression change, since different people display the same expression in different ways, PCA does not successfully separate between-class and within-class features.

We therefore rotate the feature space according to within-class covariance to enhance representativeness of the features and to improve estimation of the conditional pdfs. After rotation, some features represent predominantly within-class variation and by selecting these via eigenfiltering the influence of between-class variation on estimation is diminished. Moreover, after rotation, features are highly distinguished in terms of their ITV and compression of within-class features will affect within-class covariance more than between-class covariance and hence improves separability. The rotation matrix  $R$  is a set of eigenvectors obtained by applying singular value decomposition to the within-class covariance matrix. Every face vector  $s$  is transformed into the new space by  $R$ :

$$r = R^T s.$$

Fig. x.17 shows significant recognition performance gains over standard PCA when both changes in lighting and expression are present.

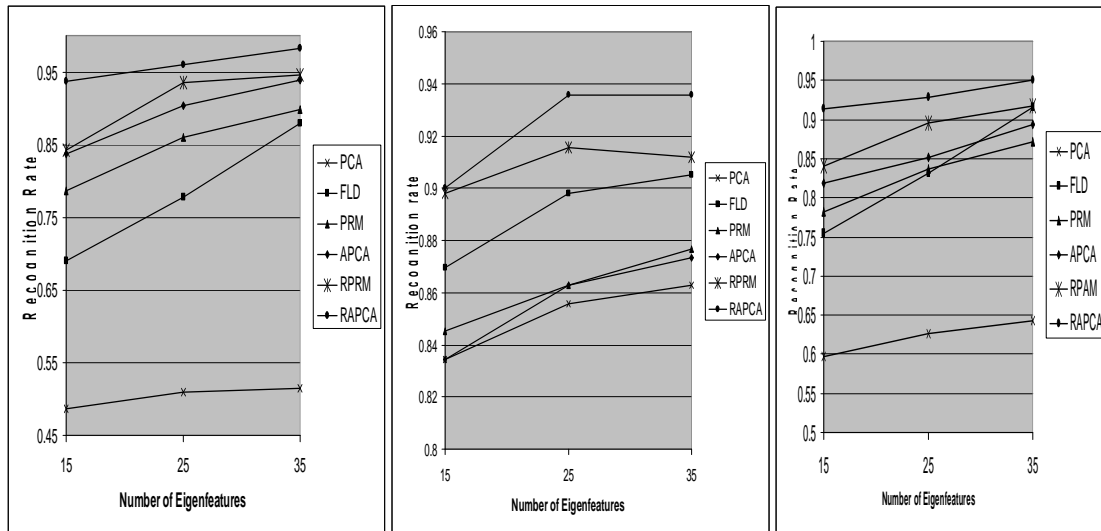


Fig. x.17: Recognition rates for RAPCA, APCA and PCA versus number of eigenfaces with variations in lighting and expression from Chen and L

### 3.3.3 Tracking

Firstly, we define a recognition confidence output, which indicates the confidence of the recognition result from the face recognizer. The confidence output is calculated from the angle  $\theta$  which measures how closely the input face image subspace matches the recognized face subspace. Two thresholds are selected to provide three levels of recognition confidence (low, medium and high). Depending on the level of confidence there are three possible outputs. If recognition confidence is low, the face will be surrounded with a red rectangle with the text "Unknown" appearing below. If medium, the rectangle will be yellow with a likely identity appearing below followed by a question mark. If high, the rectangle is green and the identity appears below without the question mark. Some selected frames are shown in Fig. x.18.

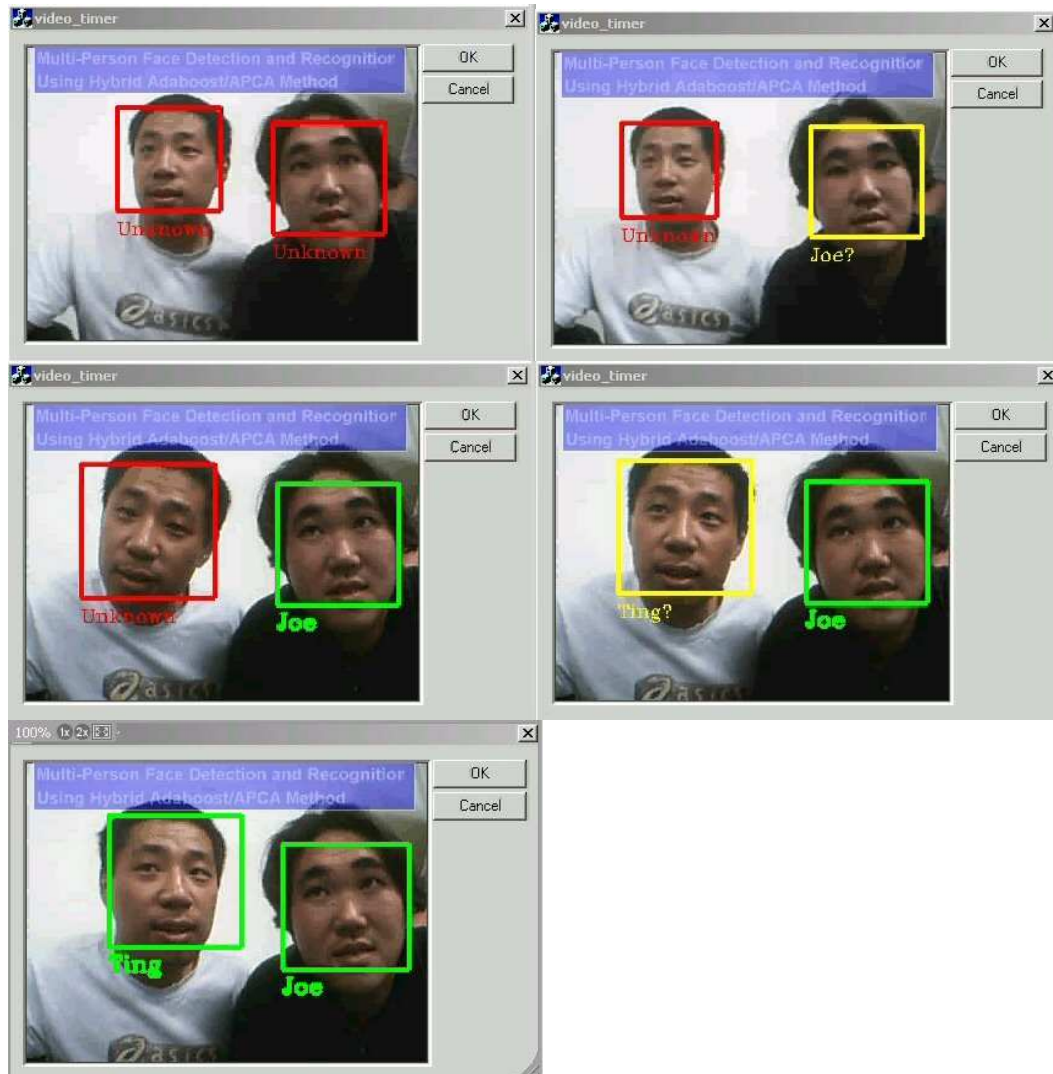


Fig. x.18: Some selected frames (from left to right, up to down) showing different recognition confidence levels using different colour rectangles.

Only the face sub-window with high confidence will be registered for tracking. Here we adopt a simple strategy for face tracking that if the position of the corresponding face sub-windows from successive frames is smaller than a certain threshold, we will accept that these face images are from the same person. The confidence output decreases over time if the recognition result falls to the low or medium level.

## 4 Summary

In this chapter we describe some technologies underpinning the pattern recognition engine of a system for locating persons on a planetary sensor network. A fully automated system must be highly robust to nuisance problems such as lighting, expression change, pose, and camera variation. Although there is a rapidly emerging need for reliable pattern recognition technology due to the explosion of digital multimedia data and storage

capacity, the technologies are mostly unreliable and slow. Yet, the emergence of handheld computers with built-in speech and handwriting recognition ability, however primitive, is a sign of the changing times. The challenge for researchers is to produce pattern recognition algorithms, such as face detection and recognition, reliable and fast enough for deployment on data gathering networks of a planetary scale.

## 5 References

- ABC News (2005): <http://www.abc.net.au/pm/content/2005/s1283572.htm> [last visited 24-Nov-2005]
- Abdel-Mottaleb M and Elgammal A (1999): Face detection in complex environments from colour images. Proc: International Conference on Image Processing, Volume 3, pp: 622-626, 24-28 Oct
- Beymer D and Poggio T (1995): Face Recognition from One Example View. Proc. Int'l Conf. of Comp. Vision, 500-507
- BioID (2005): <http://www.humanscan.de/company/index.php> [last visited 14-Dec-2005]
- Black M J, Fleet D J and Yacoob Y (2000): Robustly estimating Changes in Image Appearance. In: Computer Vision and Image Understanding, 78(1), 8-31.
- Carbonetto P (2005): <http://www.cs.ubc.ca/~pcarbo/> [last visited 14-Dec-2005]
- Chen S K and Lovell B (2004): Illumination and Expression Invariant Face Recognition with One Sample Image per Class. In: 17<sup>th</sup> International Conference on Pattern Recognition (ICPR' 04) – Volume 1 pp. 300-303
- Cootes T F and Taylor C J (1992): Active shape models—‘smart snakes’. In: Proc. British Machine Vision Conference. Springer-Verlag, pp 266-275
- Cootes T F and Taylor C J (1996): Locating faces using statistical feature detectors. Proc of the 2<sup>nd</sup> International Conference on Automatic Face and Gesture Recognition, pp: 204
- Cootes T F, Walker L and Taylor C J (2000): View-Based Active Appearance Models. 4<sup>th</sup> International Conference on Automatic Face and Gesture Recognition, pp: 227-232, March.
- Cootes T F and Taylor C J (2001): Active Appearance Models. In: IEEE PAMI, Vol.23, No.6, pp.681-685.
- Crow F (1984): Summed-area tables for texture mapping. Proc of SIGGRAPH, Volume 18(3), pages 207-212.
- Dai Y and Nakano Y (1996): Face-texture model based on sgld and its application. In: Pattern Recognition 29, pp. 1007-1017, June
- Feraud R, Bernier O and Collobert D (1997): A constrained generative model applied to face detection. In: Neural Processing Letters 5(2): 11-19
- Feret (2005): <http://www.itl.nist.gov/iad/humanid/feret/> [last visited 23-Nov-2005]
- Fleuret F and Geman D (2001): Coarse-to-Fine Face Detection. In: International Journal of Computer Vision, 41:85-107.
- Gao Y S and Leung M K H (2002): Face Recognition Using Line Edge Map. In IEEE PAMI. 24(6), June, 764-779
- Gibbons P B, Karp B, Ke Y, Nath S and Sehan S (2003), IrisNet: An Architecture for a Worldwide Sensor Web. In: Pervasive Computing, 2(4), 22-23, Oct – Dec
- Govindaraju V (1996) Locating human faces in photographs. In: International Journal of Computer Vision, Volume 19, Issue 2, August, pp: 129-146.

- Gunn S R and Nixon M S (1994): A dual active contour for head boundary extraction. In: IEE Colloquium on Image Processing for Biometric Measurement, pp: 6/1 – 6/4, 20 Apr.
- Hjelmas E and Low B K (2001) Face Detection: A Survey. In: Computer Vision and Image Understanding, Volume 83, Number 3, September, pp.236-274 (39)
- Hoogenboom R and Lew M (1996): Face detection using local maxima. In: 2<sup>nd</sup> International Conference on Automatic Face and Gesture Recognition, Oct 14-16, Killington, Vermont, USA
- Horward J (2005): <http://smh.com.au/news/national/howard-backs-more-security-cameras/2005/07/24/1122143730105.html> [last visited 23-Nov-2005]
- Huang J, Gutta S and Wechsler H (1996) Detection of human faces using decision trees. In IEEE Proc. of 2<sup>nd</sup> Int.Conf. on Automatic Face and Gesture Recognition, Vermont
- Jeng S H, Liao H Y M, Liu Y T and Chen M Y (1998): An efficient approach for facial feature detection using geometrical face model. Proc: 13<sup>th</sup> International Conference on Pattern Recognition, Volume 3, pp: 426-430, 25-29 Aug
- Lee C H, Kim J S and Park K H (1996): Automatic human face location in a complex background. 2nd International Conference on Automatic Face and Gesture Recognition, Oct 14-16, Killington, Vermont, USA
- Lv X G, Zhou J and Zhang C S (2000): A novel algorithm for rotated human face detection. In: Computer Vision and Pattern Recognition, Volume:1, pp: 760 – 765
- Mallat S G (1989): A theory for multi-resolution signal decomposition: The wavelet representation. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, 11(7): 674-693.
- McKenna S, Gong S and Liddell H (1995) Real-time tracking for an integrated face recognition system. In 2<sup>nd</sup> Workshop on Parallel Modelling of Neural Operators, Faro, Portugal, Nov
- Nikolaidis A and Pitas I (2000): Facial feature extraction and pose determination. In Pattern Recognition 33, 1783-1791.
- Osuna E, Freund R and Girosi F (1997): Training support vector machines: An application to face detection. Proc: Computer Vision and Pattern Recognition, pp: 130-136, June
- Roth D, Yang M H, and Ahuja N (2000): A SNoW-based face detector. In: Advances in Neural Information Processing Systems 12, pp: 855-861, MIT Press.
- Rowley H A, Baluja S and Kanade T (1998): Neural network-based face detection. Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp: 203-208, 18-20 June
- Saber E and Tekalp A M (1998): Frontal-view face detection and facial feature extraction using colour, shape and symmetry based cost functions. In: Pattern Recognition Letters, Volume 19, Issue 8, pp: 669 - 680
- Satoh S, Nakamura Y and Kanade T (1999): Name-it: Naming and Detecting Faces in News Videos. IEEE MultiMedia, Vol. 06, No. 1, pp. 22-35, Jan-Mar
- Sun Q B, Huang W M and Wu J K (1998): Face detection based on colour and local symmetry information. Proc of 3<sup>rd</sup> International Conference on Face & Gesture Recognition, pp: 130.
- Sung K K and Poggio T (1998): Example-based learning for view-based human face detection. ITTEE Transaction on Pattern Analysis and Machine Intelligence.

- Terrillon J, Shirazi M, Fukamachi H and Akamatsu S (2000): Invariant face detection with support vector machines. In: 15<sup>th</sup> International Conference on Pattern Recognition, Volume 4, pp: 210-217, 3-7 Sep.
- Turk M and Pentland A (1991): Face recognition using eigenfaces. Proc. Computer Vision and Pattern Recognition, pp: 586-591.
- Wong C, Kortenkamp D and Speich M (1995): A mobile robot that recognises people. Proc: 7<sup>th</sup> International Conference on Tools with Artificial Intelligence, pp: 346-353
- Yang M H, Ahuja N and Kriegman D (2000): Face detection using mixtures of linear subspaces. Proc. 4<sup>th</sup> IEEE International Conference on Automatic Face and Gesture Recognition, pp: 70.
- Yilmaz A and Gokmen M (2000): Eigenhill vs. eigenface and eigenedge. In Procs of International Conference Pattern Recognition, Barcelona, Spain, 827-830
- Yokoyama, Yagi Y and Yachida M (1998): Facial contour extraction model. Proc of 3<sup>rd</sup> International Conference on Face & Gesture Recognition, pp: 254
- Yuille A L, Hallinan P W and Cohen D S (1992): Feature extraction from faces using deformable templates. In: International Journal of Computer Vision, Volume 8, Issue 2, pp: 99-111.