

Complex Adaptive Systems, Volume 1  
Cihan H. Dagli, Editor in Chief  
Conference Organized by Missouri University of Science and Technology  
2011- Chicago, IL

## Multi-Pose Face Recognition And Tracking System

Binu Muraleedharan Nair<sup>a</sup>, Jacob Foytik<sup>a</sup>, Richard Tompkins<sup>a</sup>, Yakov Diskin<sup>a</sup>, Theus Aspiras<sup>a</sup>, Vijayan Asari<sup>a</sup>

<sup>a</sup>*Vision Lab, Kettering Laboratories, University of Dayton, 300 College Park, Dayton, OH 45469-0232, USA*

---

### Abstract

We propose a real time system for person detection, recognition and tracking using frontal and profile faces. The system integrates face detection, face recognition and tracking techniques. The face detection algorithm uses both frontal face and profile face detectors by extracting the 'Haar' features and uses them in a cascade of boosted classifiers. The pose is determined from the face detection algorithm which uses a combination of profile and frontal face cascades and, depending on the pose, the face is compared with a particular set of faces having the same range for classification. The detected faces are recognized by projecting them onto the Eigenspace obtained from the training phase using modular weighted PCA and then, are tracked using the Kalman filter multiple face tracker. In this proposed system, the pose range is divided into three bins onto which the faces are sorted and each bin is trained separately to have its own Eigenspace. This system has the advantage of recognizing and tracking an individual with minimum false positives due to pose variations.

**Keywords:** Principal Component Analysis, Face Recognition, Kalman Filter, Face Detection, Haar Features, Eigenfaces

---

### 1. Introduction

Real time systems for identifying humans in a scene has a lot of importance in security and surveillance applications where automatic detection, recognition and tracking of known individuals is required for scenarios such as restricted entry into the high profile locations, tracking of an individual in a sensitive areas etc. Human identification can be done by extracting and classifying the biometric features such as face, fingerprints, ear, iris, palm, gait or speech and all of these biometric features are either used separately or combined together depending on the security application [1]. From a video scene, biometrics such as face, ear and gait biometrics will be more suitable as these just require the images captured from a surveillance camera. Identification of humans using faces is a challenging task as the facial features of an individual are prone to changes due to illumination, facial expression, head orientation and head pose. In this paper, we are proposing a real time system to identify humans in a video scene by detecting, recognizing and tracking faces which vary in pose. The system can be divided into three major parts; Multi-view face detection using 'Haar' Cascades, Face recognition using weighted modular PCA and Multiple Face Kalman Tracker.

The system components are shown in Figure 1 and 2. In the next section, some of the techniques used in frontal face recognition are discussed along with the some of the modifications to incorporate view invariance in these algorithms.

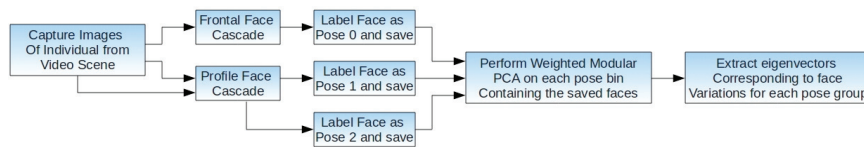


Figure 1: Training Phase

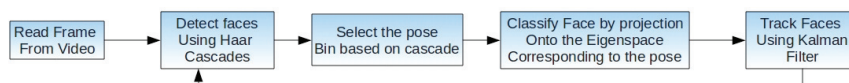


Figure 2: Testing Phase

## 2. Related Work

Turk et. al came up with the statistical approach of describing faces in terms of the variation occurring among the faces of different individuals in the dataset [2]. In other words, selected number of Eigenvectors or Eigenfaces are computed from the set of images of different individuals using principal component analysis where these Eigenfaces span the feature space which maximizes the variation between the training faces. Recognizing of a face requires only a projection of this test face onto to this reduced dimensionality Eigenspace and compares the weights obtained to those of the faces trained. Pentland et al. extended their approach of Eigenfaces to include pose variations in the database and they proposed two methods [3]. One is to get a parametric Eigenspace which will encode both the face and pose variation. The other is a set of Eigenspaces with each one representing the variation of a subset of faces with the same view. The appropriate Eigenspace for a test face can be determined by the distance-from-face-space metric [2] using each set of Eigenvectors. In addition to the Eigenfaces, other facial features such as eyes, noses and mouth were also coded to get Eigeneyes, Eigen noses and Eigenmouths. Their detection in a facial region is done by distance-from-feature-space metric.

An algorithm based on LDA was proposed by Etemad and Chellapa the problem was approached in a different than the one using PCA [4]. It focused on maximizing the separation of various face classes and minimizing the variance of faces images within a class rather than on finding a compact representation of face images like in PCA. Here, from the training images, the within class matrix and between class matrix are computed and combined to form the separation matrix. The face space or feature space is obtained by performing the Eigen analysis on this separation matrix. Performing PCA on the training set of images gives a basis set which separates pair-wise relationships between pixels, more specifically the first and second order statistics. The higher order statistical relationship between pixels, which is the phase spectrum of the face image, is not captured by the PCA. Bartlett et al illustrated that the phase spectrum captures the structural information of the image that is more useful for face recognition rather than the amplitude spectrum captured by PCA [5]. Independent Component Analysis, which is a generalized version of PCA, attempts to capture not only second order statistics but higher order statistics corresponding to the phase spectrum and thus creates a set of basis images which are independent of each other.

Liau et al. proposed an algorithm which is based on the "view-based" Eigen spaces where the view is incorporated by estimating the pose of the face in Y-Cr-Cb colorspace [6]. Depending on the pose, the faces in the database are grouped and the mean face per group is computed. Using a similarity measure using the Euclidean distance, the test face is compared with each of the mean faces for pose estimation and then, extracts the suitable features using PCA. Only a global set of eigenvectors is used and the Eigenspace spans the variations of the face due to both the pose and facial features. However, it remains unclear how the pose estimation is used to discriminate the variations due

to pose and the individual. The algorithm proposed in this paper uses a similar notion of "view-based" Eigenspaces mentioned in [3]. The database are grouped according to the pose as in [6] but each group has its own Eigenspace. Face recognition is done by projecting a detected face on to this selected Eigenspace and finding the closest match. In the next section, a theoretical overview and the implementation of the various system modules is explained.

### 3. Overview of the System Modules and Integration

#### 3.1. Face Detection

Detection of the face in an image region is done by first, computing the features known as the 'Haar' features and then applying them to a cascade of boosted classifiers [7, 8]. The image regions or regions of interest are set by scanning a window of size  $W \times H$ . The 'Haar' features which are computed from the region of interest can be classified as edge features, line features and center-surround features. Each 'Haar' feature is computed by aligning it within the region of interest and calculating the weighted sum of pixels in the shaded rectangles of the feature with opposite weights assigned to the different shades. It can be represented as  $feature_J = \sum_{i \in \{1,2\}} \omega_i \cdot RecSum(r_i)$  where  $r_i$  is the shaded rectangle of size  $(w < W, h < H)$  in the feature,  $w_i$  is the weight associated with that shade and  $J \in \{1, 2, 3, \dots, 14\}$  refers to a particular 'Haar' feature. So, from a region of interest, features as many as 16000 can be extracted and so, depending on the object to be recognized, a small portion of the features out of the big feature pool is used. The selection of the features is done by the classifier cascade where it is trained using the Discrete Adaboost algorithm [9]. For recognizing different objects, different cascades are available and in our system, both the frontal face and profile face cascades are used. The face cascades are already trained in such a way that when an image portion (windowed region) is passed through it, suitable features which can discriminate facial patterns from non-facial patterns are computed successively through each classifier stage.

The frontal face cascade detects faces with yaw variations ranging from  $-30^\circ$  to  $+30^\circ$  and this is considered as one pose bin. The profile face cascade detects right side of the face with yaw ranging from  $30^\circ$  to  $90^\circ$ . The same cascade can be used to detect left side of the face by getting the flipped version the face image along the  $x$  axis, thereby detecting faces with yaw variations of  $-30^\circ$  to  $-90^\circ$ . So, the left face detections is considered as the second pose bin and the right face detections are considered as the third pose bin. In the next section, we will see that the detected faces are separated into these bins where training and testing is done on each separately.

#### 3.2. Frontal Face Recognition and extension to Multi-Pose

The previous section explained briefly how a face is detected in an image irrespective of the identity of the individual. Once the faces are detected, the next step is to recognize the individuals whose faces are being detected and this requires the system to extract certain features from the face which will discriminate between individuals. For the face recognition system, the weighted modular Principal Component Analysis technique has been implemented [10]. In the first stage, suitable preprocessing should be done so that only the facial parts are present and non-facial parts such as hair, background scene etc. are removed. One of the ways to segment the facial parts is to use the skin segmentation algorithm and the other is to use a mask. However, for the profile faces, the mask used for frontal faces cannot be used as it will include the hair and ear sections of the face and so, different masks corresponding to each side of the faces are defined and used for profile face region segmentation. Since the PCA based face recognition, which considers global information of each face image, struggles with varying lighting conditions and head poses, a face divided into smaller regions each with its own set of representative weights is more suitable for recognition with varying conditions. Since illumination and pose invariance affect certain regions of the face, other regions may still be sufficient for a recognition decision. Weighted Modular PCA is a technique where the input face images are divided into different modules and PCA is applied to each module separately across the training images. In this modular PCA approach, each image in the training set is divided into  $N$  smaller images. These sub-images can be represented mathematically as

$$I_{ij}(m, n) = I_i\left(\frac{L}{\sqrt{N}}(j-1) + m, \frac{L}{\sqrt{N}}(j-1) + n\right), \forall i, j \quad (1)$$

where  $i$  varies from 1 to  $M$  and  $j$  varies from 1 to  $N$ ,  $M$  being the number of images in the training set and  $N$  being the number of sub-images. The average image of all the training sub-images is computed as

$$A = \frac{1}{M.N} \sum_{i=1}^M \sum_{j=1}^N I_{ij} \quad (2)$$

We then normalized each sub-image by computing the difference between the face sub-image and the average face sub-image. The covariance matrix  $C$  of the sub-images is calculated by

$$C = \frac{1}{M.N} \sum_{i=1}^M \sum_{j=1}^N Y_{ij} Y_{ij}^T \quad (3)$$

We compute the eigenvectors of the matrix  $C$  and keep the most significant number of eigenvectors,  $S$ . From these eigenvectors, the weights for each image in the training set can be computed by

$$W_{pnjK} = E_K^T (I_{pnj} - A), \forall p, n, j, K \quad (4)$$

where  $K$  takes the values 1, 2, ...,  $S$ ,  $n$  varies from 1 to  $C$ ,  $C$  being the number of images per individual, and  $p$  varies from 1 to  $P$ ,  $P$  being the number of individuals in the training set. We perform these computations on test sub-images for classification of test images into class bins. We have implemented a weighting factor that places higher weights for sub-images that contain higher variances within a face. These sections are known to have more distinguishable detail. Once the set of significant features for each module is determined, we then create a mean weight set per person per module to be a part of the trained database. The system after being trained, inputs the detected frontal faces (after masking) onto to recognition system where these faces are projected onto the modular Eigenspace. The appropriate weights are extracted and compared with those in the training set with the Euclidean distance measure. The minimum distance of the detected face from the training faces in the Eigenspace gives the class or identity of that face. The proposed system extends this face recognition algorithm to profile faces where the weighted modular PCA is used on each of the pose bins separately to get the corresponding modular Eigenspace. In short, for every bin of faces, an Eigenspace is extracted and during the testing phase, the detected face are projected onto a selected Eigenspace, the selection depending on whether the detected face is a frontal, left or right face.

### 3.3. Tracker using the Kalman filter

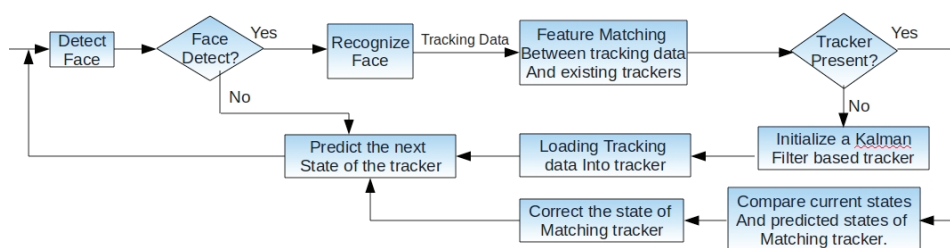


Figure 3: Tracking System

The Kalman filter is used as a tracking system and tracks the location of a detected face based on the position of the face and the identity of the individual. The states of the Kalman tracker are set to the coordinates of the bounding box of the detected face and the features used in the matching purposes for tracking is the recognition result determined by the system. A block diagram illustrating the flow of tracking for one detected face is shown in Figure 3. For each detected face in a single frame of the video sequence, a tracker based on the Kalman filter is assigned to it. The tracking data for a corresponding face are the coordinates of its bounding box and the identity of the face as determined by the system.



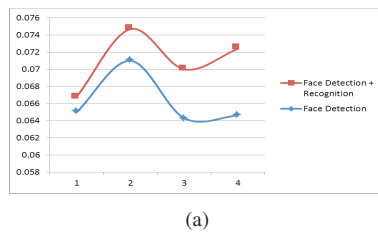
#### 4. Implementation and Performance Evaluation

The proposed system has been implemented using C++ with the OPENCV library. The system has been tested on a IP video feed of resolution  $704 \times 480$  from a Axis 214 PTZ Network Camera and set up on Z600 HP workstation with dual-quad-core 2Ghz processor on a Linux platform. The training of the individuals is done sequentially by saving 200 images of each individual at each pose, and then, using the saved faces at each pose bin to compute the Eigenspace separately. Figure 4 shows the face detection result, the face recognition (face detection and the recognition) and then the face tracking which includes both face detection and recognition. It is seen that the proposed system detects and recognizes faces with both the frontal and profile views and tracks them irrespective of the pose.



Figure 4: (a)-(d) Face Detection - green rectangle shows detection of right face, red rectangle for the frontal face and blue rectangle for the left face; (e)-(h) Multi-pose face detection and recognition ; (i)-(j) Face Tracking - shows the tracking of an individual based on the features.

The system operates at a frame rate of 10-12 frames per sec. A breakdown of the computation time of the system when 2–4 persons in the scene is given in the graph as shown in Figure 5. It can be seen that the most computation time is taken by the face detection module. This is because for detecting the face using 'Haar' cascades, a number of scans are required using a window which grows with each scan. We have optimized the face detection algorithm by setting the starting size of the window to be of size  $(1/6)^{th}$  of the frame width and height. The face detection operates faster but with the drawback that faces smaller than the starting window size will not be detected. Face recognition takes much less computation time as seen in the table as its just a matter of projecting the weights. However, depending on the number of faces, the computation time will increase with the number of faces being detected recognized. However, as seen in Figure.5, the computation time with face detection does not vary at all and remains almost a constant irrespective of the number of faces being detected. The tracker takes much less time as it just updates the state based on the position of the face and the computation time for this too depends on the number of faces being tracked. The accuracy of the system has been tested on a person rotating in front of the camera. About 250 face



Module/Num of Persons	1	2	3	4
Face Rec	0.0018	0.0033	0.0058	0.0077
Face Train	1.1326	2.2187	3.37749	4.38339

(b)

Figure 5: A graph and a table showing the computation time (secs) taken by the face detection alone and the combination of face recognition and face detection

detections in about 500 frames of the video sequence, around 22 were detected as "UnKnown" and the rest of the detections were accurately classified with either 4 – 5 giving false identities. The possible cause for this can be in the recognition of the profile faces when compared with other profile faces. Unlike the frontal face where we can use a fixed mask to get all or most of the face region while eliminating the background, the profile face cannot be masked as easily. In the system however, we have used a fixed mask so as to remove most of the background, and have trained the system in such a way that the pose variations vary from  $-60^{\circ}$  to  $+60^{\circ}$ .

## 5. Conclusions

In this paper, we have proposed a system which combined the multi-pose face detection, face recognition and face tracking where we extended the face recognition algorithm to recognize profiles faces of individuals. The system was tuned to work at real-time and have good accuracy in recognizing and tracking faces. But, some issues such as the lighting variations and small background inclusion into face recognition affects the performance. Moreover, the only three poses are included in the system, namely the frontal, right and left. Our future work will be to use a skin segmentation algorithm to create a face mask which varies according to the pose of the face and thus make it possible to eliminate the background all together from the face training system. Furthermore, we shall include a pose estimation module which will help us to determine the pose and then select a suitable Eigenspace rather than using the cascades.

## References

- [1] B.Bhanu, V.Govindaraju, *Multibiometrics for Human Identification*, Cambridge University Press, 2011.
- [2] M. Turk, A. Pentland, Eigenfaces for recognition, *Journal of Cognitive Neuroscience* 3 (1) (1991) 71–86.
- [3] A.Pentland, B.Moghaddam, T.Starner, View-based and modular eigenspaces for face recognition, in: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1994, pp. 84–91.
- [4] K.Etemad, R.Chellappa, Discriminant analysis for recognition of human face images, *Journal of the Optical Society of America* 14 (8) (1991) 1724–1733.
- [5] M.S.Bartlett, J.R.Movellan, T.J.Sejnowski, Face recognition by independent component analysis, *IEEE Transactions on Neural Networks* 13 (2002) 1450–1464.
- [6] H.F.Liau, L.M.Ang, K.P.Seng, A multiview face recognition system based on eigenface method, in: *International Symposium on Information Technology*, 2008, pp. 1–5.
- [7] P.Viola, M.Jones, Robust real-time face detection, in: *IEEE International Conference on Computer Vision*, Vol. 2, 2002, p. 747.
- [8] R.Lienhart, J.Maydt, An extended set of haar-like features for rapid object detection, in: *International Conference on Image Processing Proceedings*, Vol. 1, 2002.
- [9] Y.Freund, R.E.Schapire, Experiments with a new boosting algorithm, in: *In Machine Learning: Proceedings of the Thirteenth International Conference*, 1996, pp. 148–156.
- [10] R.Gottumukkal, V.K.Asari, An improved face recognition technique based on modular pca approach, *Pattern Recognition Letters* 25 (2004) 429–436.