Indian Journal of Science and Technology, Vol 9(S1), DOI: 10.17485/ijst/2016/v9iS1/109672, December 2016
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

An Efficient Face Detection and Recognition for Video Surveillance

Dipti Mishra[1*], Mohamed Hashim Minver[2], Bhagwan Das[3], Nisha Pandey[4] and Vishal Jain[5]

[1]ECE Department, Pranveer Singh Institute of Technology, Kanpur - 209305, Uttar Pradesh, India; dipti.mishra28@gmail.com
[2]Addalaichenai National College, Srilanka; mhminver@gmail.com
[3]University Tun Hussein Onn Malaysia, Malaysi; Engr.bhagwandas@hotmail.com
[4]Gyancity Research Lab, India; nisha@gyancity.com
[5]Bharati Vidyapeeth's Institute of Computer Applications and Management (BVICAM) New Delhi - 110063, Delhi, India; vishaljain83@ymail.com

Abstract

In this paper, a comprehensive scheme is proposed for unconstrained joint face detection and recognition in video sequences for surveillance systems. Unlike conventional video based face recognition techniques, emphasis is laid on the acquisition of a pose constrained training video database followed by the extraction of well aligned face images from the training videos. We have proposed a new Indian Faces Video Database (IFVD) to demonstrate the performance of the proposed approach especially in the challenging environment of varying skin color and texture of faces from the Indian subcontinent. Our approach produces successful face tracking results on over 86% of all videos. The good tracking performance induces high recognition rates: 85.86 on Honda/UCSD and over 77.49 % on IFVD. The proposed technique is robust and aims to develop a unified framework to address the challenges of varying head orientation, pose and illumination level in a highly integrated fashion so as to benefit from the interdependence between the high fidelity face detection and the subsequent recognition phases.

Keywords: Adaboost, Classification, Face Detection, Face Recoginition, Kalman Tracking, Manifold Learning, SVM

1. Introduction

In recent years, face detection and recognition have attracted significant research interest which demands for robust face recognition techniques in surveillance and authentication applications. Video surveillance is used to catch the mischievous person which is unauthorized for access. So we need to have a biometric system which utilizes any biological information like our finger prints, iris, facial information etc. Spatio-temporal face recognition requires various aspects like detection, tracking, face preprocessing for illumination correction and recognition based on appearance manifolds. In literature, several schemes have been proposed in a variety of applications mentioned in[1]. However a key drawback of such work is that the framework adopted for developing the requisite mathematical models and algorithms typically considers only a single component at a time while strictly controlling other components, often manually. For instance in these applications[1] which considers face recognition, people used databases containing facial mug-shot images which are often manually aligned and cropped. However, practical detection and tracking systems seldom output well aligned images. Thereby such schemes which focus on the individual components, led to error propagation through the various other components. This is one of the principle reasons why such schemes fail to perform satisfactorily when employed in real-world scenarios. Thus

*Author for correspondence

it is important to design and study the performance of these components comprehensively as part of a real scenario, rather than optimizing the individual components, which tends to ignore the critical interdependencies. The beauty of the principle is that it is substantially easier for the user to provide training videos of the persons rather than manually cropped and registered images. So that the system should be able to successfully recognize suitably aligned face images from these video sequences. The quality of images has a direct impact on the recognition performance. Secondly the training videos should capture and represent a large variety of in-plane orientation and out of plane pose variations. This allows developing face models[2,3] to provide a richer description of each subject's face.

A novel algorithm is proposed for high fidelity detection of both the location and orientation of face using OpenCV detector, which facilitates the extraction of well registered face images from the training video set. In context of recognition, the appearance manifolds in[2] framework is subsequently employed to construct face models based on these appropriately registered images. Further the pose constraints are imposed during the training video acquisition. A new Indian Faces Video Database (IFVD) has also been constructed to capture the vast diversity in the faces of Indian subjects. Further the IFVD strives to represent maximal pose variations. The tracking and recognition procedure for the test videos is described by Lee[4] with a few modifications in order to achieve more robust and smooth detection performance. The performance of the proposed system is demonstrated using Honda/UCSD and IFVD video databases.

2. Literature Survey

For face detection, Viola and Jones proposed a scheme in[5] which continues to be a popular frontal face detector. The work was later extended in[6] to detected faces with varying poses like pan and tilt and orientations like in-plane and out-of-plane poses. Viola Jones detector based on Haar-like rectangular features were employed, learnt using Adaboost[7] for different combination of the pose and in-plane rotations in order to draw the final bounding box around the face. But the problem with that is the training time to train the videos and complexity is very large.

Moreover a drawback of that scheme is it employs face detection in each frame, rather than employing face tracking initialized by a face detector, and therefore fails to exploit prior information available in the form of spatio-temporal continuity of the video sequence. The visual tracker that was proposed earlier[8] is a popular tracking algorithm which employs subspace based adaptive appearance model that is learnt online, rather than tracking a fixed target. This was enhanced in[1] to deal with the problem of drift which occurs due to the adaptation of the appearance model to non-targets. While these schemes provide efficient solutions for tracking faces, they are computational complex. In contrast our algorithm offers a simpler solution which also addresses the issue of pose variations while simultaneously avoiding drift by using reinforcements from the face detector. Other commonly used appearance based techniques for face tracking in videos are based on Mean-Shift[9] and particle filters[10] both of which are pixel intensity based approaches. In contrast our algorithm employs an adaptive motion model in the form of a Kalman filter, which complements the pixel intensity based face detector more naturally.

Evident integration is also a important aspect for face recognition in real world videos. The work in[3] proposes a technique for face recognition from video sequence using high dimension probability density estimation followed by density matching using the Kullback Leiber Divergence[11]. However a major drawback of the scheme is that it is offline.

The probabilistic appearance manifold framework[3] offers an efficient on-line evidence integration technique for video based face recognition. This work was further extended in[4] to also incorporate face tracking using appearance manifolds. However appearance manifolds are not available during extraction of faces from training videos. This is the major thing which motivated us to the current work. The proposed scheme has increased robustness to pose variation together with a smoother tracking and recognition performance. Typically OpenCV detector is employed to detect and locate the frontal poses, then the bounding box are extracted from the video frame for face preprocessing for illumination correction to increase the facial features needed for recognition. The processed face is then resized and features are fed to classifiers for recognition.

3. Face Detection

OpenCV detector is able to detect the Spatio-temporal location of face in video frame. It is based on Viola and Jones detection which is using Haar Like rectangular

features. The limitation of this detection is it only able to detect the frontal faces and also detecting some non-faces in a crowded environment which is undesired. Most of the non-faces detected by the OpenCV detector are deprived of skin color pixels. To eliminate these non-faces we have incorporated Skin filter as shown in figure 1. Skin filter eliminates the detected objects if it doesn't contain certain number of skin pixels. We have implemented Color based and Gaussian Model based Skin filter to detect skin pixels. Color based skin filter uses particular parameter for the RGB color space defined in[12].

For skin pixels, skin ratio is calculated which is the ratio of total number of skin pixels and total number of pixels in the face. Skin ratio above the threshold is finally treated as face. The threshold is learnt by running the filter on various databases i.e. Choke Point Video database, Honda/UCSD Video database, Indian face video database. The limitation of this filter is luminance exists in color space which depends on position of light source. Secondly only certain skin colors can be detected due to which false positives become high. The luminance is removed by using Gaussian model based Skin filter which uses chromatic color space i.e., normalized RGB.

$$r = \frac{R}{R+G+B} \quad \text{and} \quad b = \frac{B}{R+G+B}$$

where R,G,B= Red, Green and Blue Channel pixel of RGB image. The Gaussian model for skin detection is used as discussed in[13]. The model consists of large skin dataset[14] with various age groups (young, middle and old) and racial groups. Total learning sample size is 245057 in which 50859 are the skin samples and rest is non-skin samples. Skin samples are passed through low pass filter to reduce noise in the dataset. According to[13], distribution of skin color is implemented using Gaussian Model $N(\mu,\sigma)$

$$\mu = \frac{1}{N}\sum_{i=0}^{N-1} x_i \quad \text{Where } x_i=[r,b]^T \tag{1}$$

$$\sigma = \frac{1}{N}\sum_{i=0}^{N-1} (x_i-\mu)(x_i-\mu)^T \tag{2}$$

Where N is Number of skin pixels and x_i is the RGB pixel matrix of i^{th} skin sample. Skin pixel by using Gaussian Model based Skin filter is calculated as shown in figure 2

By using this filter detection is relatively faster, false positives are low and detection can be taken over wide variety of skin color. Skin filter is then followed by PCA filter to check whether the detected object is face or non-face. This filter uses Pose faces for comparison between face and non-face. Each image of GTAV face database is cropped (as shown in Figure 3) across face and mean of all these images is calculated to get the full frontal face.

Each pose face image (N×N) is converted to column vector ($N^2\times1$) and all the five images are accumulated in matrix T. Then the size of this matrix is further reduced by using PCA to reduce computation and called as projection matrix ϕ_i .The object to be detected is also converted to column vector and resized by using PCA and is called test projection matrix ϕ_t .Then the euclidean distance between ϕ_i and ϕ_t is calculated. Among these five distance the smallest is compared with the threshold (learnt over various databases) to prove whether it is a face or non-face.

4. Face Detection

Facial poses can be broadly classified into two types i.e. frontal and non-frontal (as shown in Figure 4). The non-frontal poses are further classified into in-plane and out-of-plane poses. The OpenCV detector is not able to detect non-frontal faces, so to detect non-frontal faces there is need of tracking. Sometimes tracking is also needed when there is change of illumination over the face.

Efficient recognition takes place on faces with all facial features. Frontal and in-plane pose contains all facial features. So we have implemented a face tracker to detect these non-frontal poses. Mean shift tracking is implemented which is a non-parametric (kernel) based tracking and uses color histogram for tracking. It is an iterative mechanism which compares color histogram of the target object and the candidate region until the similarity coefficient i.e. Bhattacharya coefficient is maximized (as shown in Figure 5). Gaussian kernel of size equal to target window is used and then the kernel values are added to target window. Then the color histogram of target model and search window is calculated. Bhattacharya coefficient $\rho(y)$ is used to calculate the similarity score between the color histograms.

$$\rho(p(y),q) = \sum_{u=0}^{n-1} \sqrt{(p_u(y)q_u)} \tag{3}$$

Where p(y) is the n- bin histogram of search y location, q is the n-bin histogram of target and n is the size of histogram bins. Larger the value of $\rho(y)$, larger the

similarity score and the search window will move in the direction of increasing Bhattacharya coefficient, until it is maximized. Mean shift tracking is able to track non-frontal faces but the orientation problem (in-plane poses) is still not taken into consideration. Recognition can be further increased by resolving orientation problem. There is a need of a new tracker which tackle both i.e. tracking of non-frontal faces as well as orientation problem.

The proposed scheme (as shown in figure 6) employs faces which are localized in the video frame using bounding boxes. Each such bounding box is characterized by a quadruple [x,y,s,θ], where (x,y) denotes the coordinates of the center of the bounding box from the left corner of the frame, s denotes the mean and height of the bounding box and θ is the angle by which the bounding box is rotated with respect to horizontal. The algorithm leverages the spatio temporal characteristics of a typical surveillance video captured by a static camera. Since the algorithm is online, it can be assumed without loss of generality that the bounding box for i-1th frame is available while localizing the face in the ith frame. Typical modern surveillance cameras record high quality videos at 25-30 frames per second. Thus the ROI in the ith frame is substantially narrowed to within a few pixels around μ_{i-1}. Employing this prior information, N candidate bounding boxes are generated for the ith frame by random sampling from the multivariate Gaussian distribution centered at μ_{i-1} (Best N=20).

$$\rho_1.\rho_2.\rho_3.\rho_4..... \sim N(\mu,\sigma) \quad (4)$$

Where ρ_i is the bounding box for orientation estimation, μ is the previous bounding box i.e., $[x, y, s, \theta]$, $s = \frac{w+h}{2}$ Where w, h are the width and height of the detected face and $\sigma = diag[16,16,5,144]$.

Then to tackle tracking of non-frontal poses, we implemented Template matching which is a technique to find template image in source image by using different types of scoring. One of the most common measure used in template matching is to compare the similarity of different patches of input image with the template is SAD (Sum of Absolute Differences)[15]. Let I_i and I_t denotes intensities of the input and template images respectively. The matching score at the image coordinate (x,y) is given as,

$$SAD(x,y) = \sum_{i=1}^{T_r}\sum_{j=1}^{T_c}\left|I_i(x_i,y_i) - I_t(x_t,y_t)\right| \quad (5)$$

Where I_i and I_t are the source and template image. The coordinate with the largest score is the estimate of the location of the template within the input image. Figure 7 shows the results of template matching.

For a sufficiently large N, one of the candidate bounding boxes is oriented in the same direction as the actual face with a high probability. Note that if the face is oriented at an angle θ measured counter-clockwise from the horizontal, rotating the frame by 90^0 - θ makes the face vertical.The bounding boxes location are rectified by using OpenCV and template matching on rotated frame. But the template matching is not an efficient tracker as it does not work on rotated faces (The template can never be matched).So we implemented Kalman filter to correct bounding boxes of template matching for rotated frames using past observations. Kalman filtering is a recursive analytical technique to estimate time dependent physical parameters in the presence of noise. The Kalman filter uses the equations of kinematics to predict the movement of pixel in the next frame from the past statistical variations I the measurements. This comprises of 2 stages the prediction and the correction stage. The block diagram of working of Kalman filter is shown in Figure 8. The assumptions for the face tracking model are the velocity model and the measurement (w_i) and the process noise matrix(v_i) are taken as Gaussian distributed such that

$$E\left\{w_k w_l^T\right\} = \begin{Bmatrix} Q_k \; fork = l \\ 0 \quad otherwise \end{Bmatrix}$$

$$E\left\{v_k v_l^T\right\} = \begin{Bmatrix} R_k \; fork = l \\ 0 \quad otherwise \end{Bmatrix}$$

$$E\left\{w_k v_l^T\right\} = 0 \vee l, k$$

Where Q_k and R_k are symmetric positive semi-definite matrices. In velocity model, the system state matrix (b_i) will be:

b_i= [position, velocity]

The bounding box is given by $[x, y, s, \;]$,where (x,y)= position of face from top left corner of frame.s is the mean of width and height and theta is the face orientation angle. System state matrix for Kalman filter will be $b_i = [x, y, s, \theta, \dot{x}, \dot{y}, \dot{s}, \dot{\theta}]^T$, where $[\dot{x}, \dot{y}, \dot{s}, \dot{\theta}]$=[1,1,0.1,1]

The initial error covariance matrix (P_0) is given higher value to get a good estimate i.e. $P_0=I_8.10^4$ and error vari-

ance is taken as equal for all components of state vector i.e. R= 42.25.I_4. The estimated bounding box b_i^- using i^{th} bounding boxes generated from Gaussian distribution is $b_i^- = Ab_i + w_k$, where state transition matrix A is given as

$$A = \begin{Bmatrix} I_4 & I_4 \\ 0_4 & I_4 \end{Bmatrix}$$ And the process noise is

$$w_i = \left[\frac{a_x}{2}, \frac{a_y}{2}, \frac{a_s}{2}, \frac{a_\theta}{2}, a_x, a_y, a_s, a_\theta \right].$$

Updating the error covariance equation as $P_i^- = AP_{i-1}A^T + Q$, Where Q is the process error covariance matrix i.e. $E\{w_i w_i^T\}$

Now the Kalman filter gain is given by

$$K = P_i^- H^T (HP_i^- H^T + R)^{-1} \quad (5)$$

The measurement stage is updated by

$$b_i^+ = b_i^- + K(b_i^m - Hb_i^-) \quad (6)$$

Where H is called the output transition matrix and is given as $H = [I_4, 0_4]$, b_i^+ is the corrected value of system state matrix and b_i^m is measured system state matrix i.e. $b_i^m = [x, y, s, \theta]$.So the updated error covariance is given as,

$$P_i = (I - KH)P_i^- \quad (7)$$

Where P_i is estimated error covariance matrix. Therefore the proposed detailed algorithm for template matching and Kalman filtering is shown in Figure 9 and its results are shown in Figure 10 and Figure 11 respectively.The proposed QoR score is a measure of the extend of the match. between the alignment of the bounding box and that of the enclosed face. In order to characterize this, a linear classifier is trained on the Head Pose Database[16] which consists of 2790 labeled face images of 15 people with varying degrees of pan and tilt. The faces with either pan or tilt in the range [-15^0,+15^0] are chosen as the positive training samples while the remaining images are used as negative samples.

The classifier is trained on the HoG (Histogram of Oriented Histograms) features (z)[17] extracted from the images in the two classes. Since the pose variation (as shown in Figure 11) within these two classes is itself quite large, a single linear support vector classifier yields poor results. To overcome this problem, we employ multiple weak linear classifiers integrated together using the principle of Adaboost[18] to obtain a strong classifier. Each of the weak classifiers is obtained by learning one linear SVM on a subset of the training data obtained by sampling from a given discrete distribution over the training samples. The final classifier is a linear combination of the SVMs trained in each iteration, with the weighting coefficient set as a decreasing function of the classification error. The confidence score of the classifier is used as the metric for choosing the best candidate bounding box. Using the SVM classifier each of the N bounding boxes is scored. Final Score $H_k(z)$ with the maximum value is chosen as best frontal face estimate of the previous detected face.

$$H_k(z) = \sum_{i=1}^{T} \alpha_i h_i(z) \quad (8)$$

where $H_k(z)$ = Final score for k^{th} bounding box, α_i are the weights, $h_i(z)$ is the Classifier trained for HoG features of individual person= $a^T z + b$, z is HoG feature of person (1116×1), T is Number of weak classifiers (Thirty), k isindex of the bounding boxes [1, 2, .N].

Face preprocessing is also needed to increase the human facial features under following conditions to obtain good results in recognition. There may be illumination variation when the light source is changing or there may be illumination change caused due to pose variations. So there is highly need of normalizing the illumination of the face. Illumination correction algorithm is based on Weber law[19] which states that human perceives any stimuli relative to the background rather than perceiving it in absolute terms.

$$k = \frac{\Delta I}{I} \quad (9)$$

Where ΔI is increment threshold, I is initial stimulus intensity, k is weber fraction (remains constant with variable I)

Now according to Lambertian reflectance model[20], a face image is represented as:

$$F(x,y) = R(x,y)I(x,y) \quad (10)$$

Where F(x,y) = image pixel value, R(x,y)= reflectance value and I(x,y) denotes the illuminance at each pixel. Weber Law Descriptor (WLD) is given by

$$\varepsilon(x_c) = \arctan(\alpha \sum_{i=0}^{p-1} \frac{x_c - x_i}{x_c}) \quad (11)$$

Where α_i = adjusting parameter (magnifying/shrinking) the intensity difference between neighboring pixels, p is the number of neighboring pixels, x_c is the center pixel, x_i = neighboring pixel. WLD applied to the face images F(x,y) gives an illumination invariant representation of F known as "Weber-Face" (WF).

$$WF(x,y) = \arctan(\alpha \sum_{i\in A} \sum_{j\in A} \frac{F(x,y) - F(x - i\Delta x, y - j\Delta y)}{F(x,y)}) \quad (12)$$

Where A={-1,0,1}

The illumination component is commonly assumed to vary slowly, which gives us:

$$I(x - i\Delta x, y - i\Delta y) \approx I(x,y) \quad (13)$$

So the equation becomes

$$WF(x,y) = \arctan(\alpha \sum_{i\in A} \sum_{j\in A} \frac{R(x,y) - R(x - i\Delta x, y - j\Delta y)}{R(x,y)}) \quad (14)$$

After applying Weber's law, the reflectance gets constant on face image and the illumination corrected image is obtained as can be seen in Figure 12.

5. Face Recognition

Appearance manifolds[2] are low dimensional subspace based representations for face images of a person, suitable for drastic pose variations. Each person is represented by a collection of PCA subspaces (each of which corresponds to a particular pose) and the corresponding probabilities of transition from one pose to other. The manifolds are constructed from the images extracted from the training videos. We used the distance metric proposed in[2] to find the distance of a new image from the appearance manifolds of each individual in the training database. Manifolds represent space which resembles Euclidean space near each point and is non-linear in nature. The collection of linear subsets is called Pose Subspace. Each pose subspace is an affine plane computed through PCA. The connectivity between the pose subspaces is learnt over the training videos. Let M_k be the manifold for k^{th} person(shown in Figure 13). M_k will be the collection of pose subspace (C_k^i) where i=1….N for N pose subspace. To recognize the person we calculate the minimum Hausdorff distance between manifold (M_k) to I (test image).

$$k^* = \arg\min_k d_H(I, M_k) \quad (15)$$

Where d_H is the Hausdorff distance between the image I and M_k. Hausdorff distance measures the greatest of all distances from apoint in one set to the closest point in the other set.

$$d_H(x,y) = \max\{\sup_{x\in X} \inf_{y\in Y} d(x,y), \sup_{y\in Y} \inf_{x\in X} d(x,y)\}$$

Where X, Y are two different sets $d(X,Y)$ is the distance between them.

Distance between manifold and image is given by

$$d_H(I, M_k) = \int_{M_k} d(x,I) P_{M_k}(x|I) dx \quad (16)$$

Where $P_{M_k}(x|I)$ = conditional probability of x being optimal point on M_k given I

Using total probability theorem,

$$P_{M_k}(x|I) = \sum_{i=0}^{p} P_{C_k^i}(x|I) P(C^i{}_k|I) \quad (17)$$

Using both above equation we get,

$$d_H(I, M_k) = \sum_{i=0}^{p} P(C^i{}_k|I) \int_{M_k} d(x,I) P_{M_k}(x|I) dx \quad (18)$$

$P_{C_k^i}(x|I)$ is conditional probability of x belonging to pose subspace C_k^i in manifold M_k. $P(C_k^i|I)$ and $P(C_k^i|I)$ is the probability that C_k^i will contain x that has minimal distance from I

$$P(C_k^i | I_t, I_{0:t-1}) = \alpha P(I_t | C^i_{k_t}) \sum_{j=1}^{p} P(C^i_{k_t} | C^i_{k_{t-1}}) P(C^i_{k_{t-1}} | I_{t-1}, I_{0:t-2}) \quad (19)$$

where α is normalization factor which ensures that

$$\sum_{i=1}^{p} P(C_k^i | I_t, I_{0:t-1}) = 1,$$

$P(I_t|C_k^i)$ is the probability that face image $I_t \in C_k^i$ can be conveniently calculated as

$$P(I_t | C_k^i) = \gamma_k^t \exp(\frac{d_H^2(I_t, C_k^i)}{2\sigma^2}) \quad (20)$$

Where γ_k^t is the normalization factor ensures

$\sum_{i=1}^{P} P(I_t | C_k^i) = 1$, $P(C^i_{k_t} | C^i_{k_{t-1}})$ is the transition probability of x in the current frame lying in C_k^i such that the previous frame it belonged to C_k^i. Therefore the face recognition algorithm is given in Figure 10.

6. Results

We have experimented into two databases i.e., Honda/ UCSD video database[4] and Indian face Video database. Honda database is used to provide a standard video database for providing a wide range of different pose for evaluating face tracking/recognition algorithms. The resolution of each video sequence is 640×480. The recognition is performed by using 9 training video and 1-2 test videos per training video. IFVD characterize the effect of structured pose variations (glasses, beard, moustache and hair styles). The database consists of 30 training (one video per person) and 38 test videos and has significant pose variations with the resolution of 710×576.In face detection we have experimented on two different algorithms i.e., (i) Voila Jones (ii) Voila Jones and filters. Table I is showing the variation in results.

In face tracking we have performed on two different algorithms i.e., mean shift tracking and another is Template matching and Kalman Filter based tracking(TMKL) (shown in Table II)

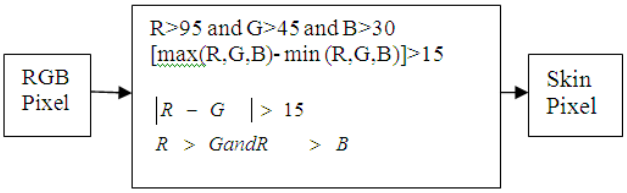


Figure 1. Skin filter.

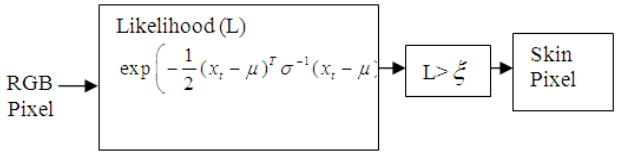


Figure 2. Gaussian based Skin filter.

Figure 3. Cropped Images at different angles.

In face recognition, appearance manifold is implemented on two types of tracking i.e. mean shift tracking and TMKL. The results are shown in Table III.

Most of the faces detected by mean shift but not by TMKL are non-frontal and the recognition of frontal faces is higher than non frontal.Face recognition is based on presence of facial features like size and location of nose, eyes, mouth etc.Non-Frontal faces lack these many of these features which results in low recognition. Table IV shows the difference in recognition rate of frontal and non-frontal faces.

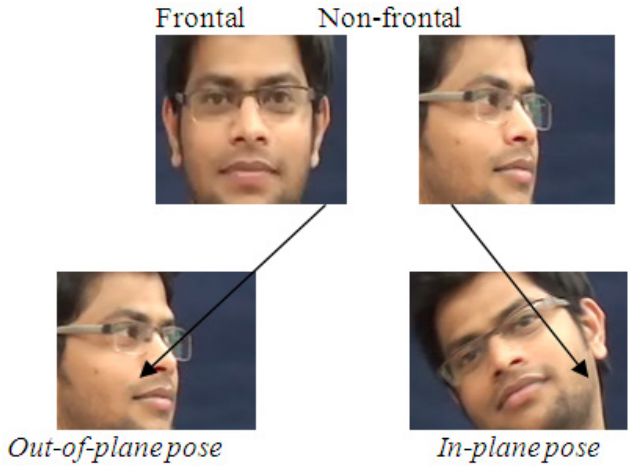


Figure 4. Different types of poses.

Table V contains the training and testing time for the system for varying image sizes. The reason for sluggish increase in training time is due to presence of many other functions like detection and preprocessing which are not heavily dependent on feature size. The reason for sudden increase in testing time is due to Nearest Neighbor Classifier is heavily dependent on size of feature vector as it takes norm of vectors.

The software can be made on many platforms such as Python, C++ and MATLAB. The video used for testing is of 12 sec with resolution of 720×576.The run time on Python, C++ and MATLAB is 25.57, 20.39.95.985603 seconds respectively++ and OpenCV have some compatibility problems with the GUI.

7. Conclusions

An end-to-end integrated face detection, tracking and recognition system has been proposed in this paper. The framework developed emphasizes the extraction of appropriately aligned faces from a pose constrained training video database. This has been shown to lead to a significant enhancement in the performance of the video based face detection and recognition system. A new IFVD has been developed to test the performance of the proposed scheme, along with existing video database. The

proposed system shows a robust tracking and recognition performance in the presence of orientation, pose and illumination variation with minimal system complexity.

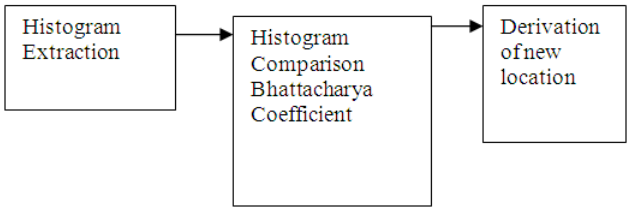


Figure 5. Histogram Comparison by Bhattacharya coefficient.

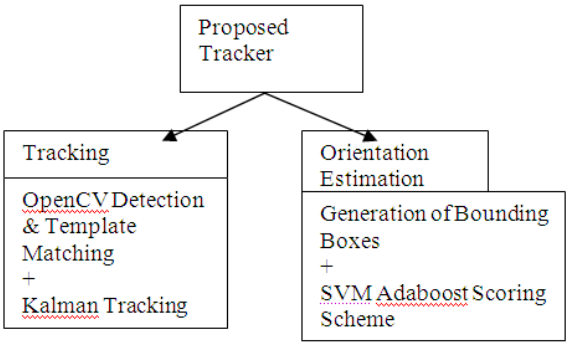


Figure 6. Proposed Algorithm.

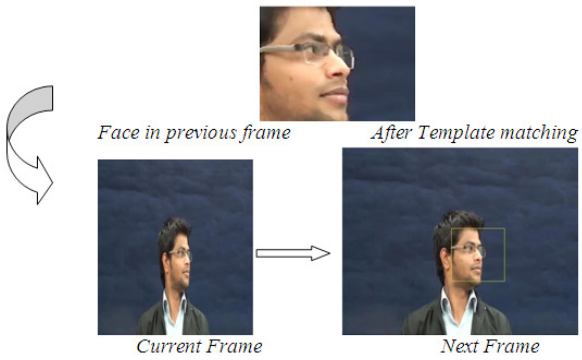


Figure 7. Results of template matching.

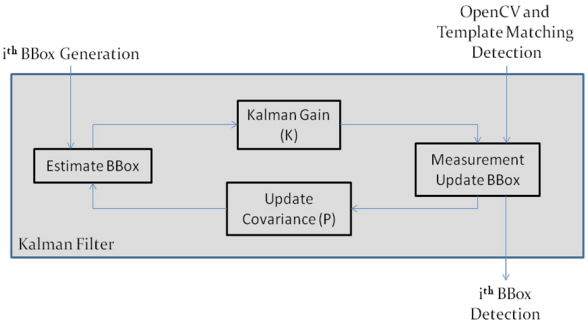


Figure 8. Block diagram operation of Kalman filtering.

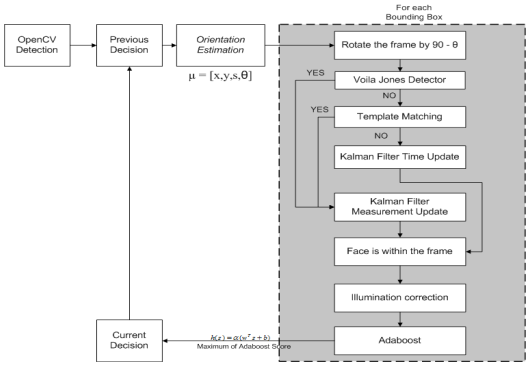
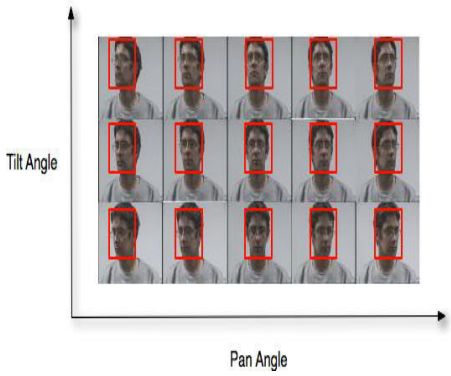


Figure 9. Face Tracking Algorithm.

Figure 10. Face Tracking results using Kalman Filter.



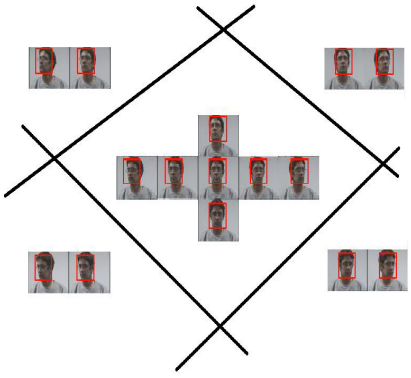
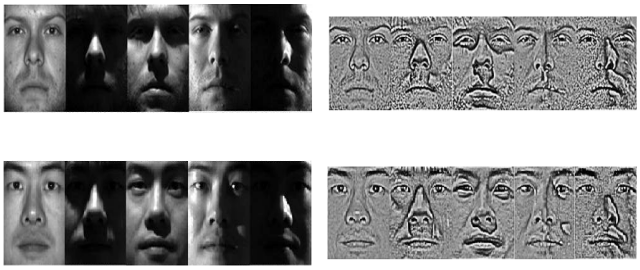


Figure 11. Different poses at different angles of pan and tilt.



Original images Weber Face images

Figure 12. Weber faces obtained after applying Weber's law.

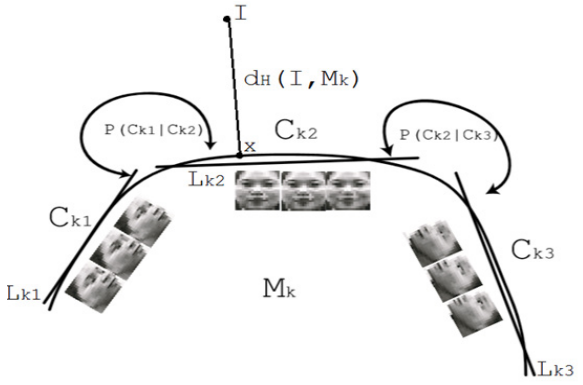


Figure 13. Appearance Manifold for k[th] person.

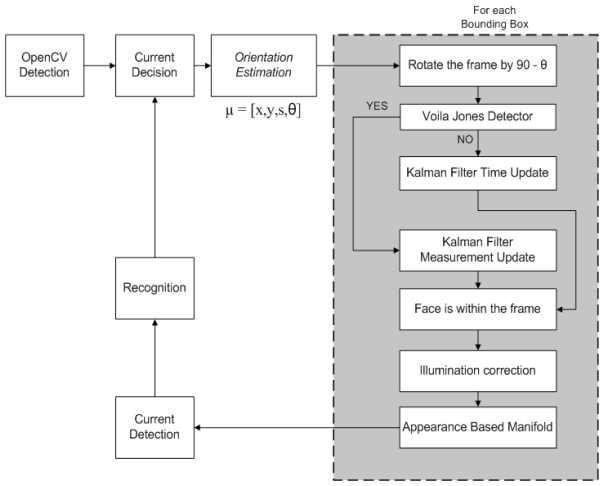


Figure 14. Face Recognition Algorithm at testing phase.

Table 1. Face Detection results of two Algorithm in (%)

Database	Honda/UCSD Database		Indian Video Database	
Algorithm	Voila Jones	Voila Jones and Filters	Voila Jones	Voila Jones and Filters
Accuracy	88.4	94.1	86.5	93.7

Table 2. Face Tracking results of two Algorithm in (%)

Database	Honda/UCSD Database		Indian Video Database	
Algorithm	MS tracking	TMKL tracking	MS tracking	TMKL tracking
Accuracy	84	82.4	87	86.4

Table 3. Face Recognition results of tracking Algorithm (%)

Database	Honda/UCSD Database		Indian Video Database	
Algorithm	MS tracking	TMKL tracking	MS tracking	TMKL tracking
Accuracy	78.67	85.86	70.67	77.49

Table 4. Recognition results of Frontal vs Non-Frontal poses in (%)

Feature Size	24×24	32×32	48×48
Frontal Face	76.27	77.39	75.96
Non-Frontal Poses	8.3	6.3	7.14

Table 5. Running time of various feature size

Feature Size	24×24	32×32	48×48
Training	5.88 hrs	5.96 hrs	6.03 hrs
Testing	1.21 s	2.44s	7.53 s

8. References

1. Kim M, Kumar S, Pavlovic V, Rowley H. Face tracking and recognition with visual constraints in real world videos. IEEE Conference Computer Vision and Pattern Recognition. 2008 Jun. p. 1–8.
2. Lee KC, Ho J, Yang MH, Kriegman D. Video-based face recognition using probabilistic appearance manifolds. Computer Vision and Pattern Recognition, 2003 Proceedings, 2003 IEEE Computer Society Conference. 2003; 1: I–313.
3. Shakhnarovich G, Fisher JW, Darell T. Face recognition from long-term observations. Computer vision ECCV, Springer. 2002; 851–65.
4. Lee KC, Ho J, Yang MH, Kriegman D. Visual tracking and recognition using probabilistic appearance manifolds. Computer Vision and Image Understanding. 2005; 99(3):303–31.
5. Viola P, Jones M. Robust real-time object detection. International Journal of Computer Vision. 2001; 4:34–47.

6. Jones M, Viola P. Fast Multiview face detection. Mitsubishi Electric Research Lab TR-20003-96. 2003; 3:14.
7. Freund Y, Schapire RE. A decision-theoritic generalization of online learning and an application to boosting. Computational learning theory, Spinger. 2008; 23–37.
8. Ross DA, Lim J, Lin RS, Yang MH. Incremental learning for robust visual tracking. International Journal of Computer Vision. 2008; 77(1-3):125–41.
9. Comaniciu D, Ramesh V, Meer P. Real-time tracking of non-rigid objects using mean shift. Computer Vision and Pattern Recognition, 2000 Proceedings. IEEE Conference. 2000; 2. p. 142–9.
10. Nummiaro K, Koller-Meier E, Van Gool L. An adaptive color-based particle filter. Image and Vision Computing. 2003; 21(1):99–110.
11. Kullback S. Information theory and statistics. 1968.
12. Peer P, Kovac J, Olina S. Human Skin color clustering for face detection. EUROCON 2003, International Conference on Computer as a Tool. 2003.
13. Yang J, Waibel A. A Real-Time Face Tracker. CMU CS Technical Report.
14. Bhatt R, Dhall A. Skin Segmentation Dataset. UCI Machine Learning Repository.
15. Dawoud NN. Fast template matching method based optimized sum of absolute difference.
16. Gourier N, Hall D, Crowley JL. Estimating face orientation from robust detection of salient facial structures. FG Net Workshop on Visual Observation of Deictic Gestures, FGnet (IST-2000-26434) Cambridge, UK. 2004; 1–9.
17. Dalal N, Triggs B. Histograms of oriented gradients for human detection. Computer Vision and Pattern Recognition. IEEE. 2005; 1:886–93.
18. Schapire RE, Singer Y. Improved boosting algorithms using confidence-rated predictions. Machine Learning. 1999; 37:297–336.
19. Chen J, Shan S, He C, Zhao G, Pietikainen M, Chen X, Gao W. WLD: A robust local image discriptor. IEEE Transactions Pattern Analysis and Machine Intelligence. 2010; 32(9):1705–20.
20. Lambert JH. Photometric sive de mensure gratibus luminis colorum umbre Eberhard Klett. 1760.