



INDIAN INSTITUTE OF TECHNOLOGY, KANPUR
DEPARTMENT OF ELECTRICAL ENGINEERING

B.TECH. PROJECT REPORT

**Face Tracking and Recognition with Orientation,
Pose and Illumination Variations**

Submitted by:
Shubham Gupta (10699)
Tanmay Gupta (10758)

Under the guidance of:
Prof. Aditya K. Jagannatham

April 17, 2014

Abstract

In this paper we propose a scheme for unconstrained face detection and recognition in videos. Unlike other works where face detection, tracking and recognition are treated as separate entities in a pipeline, our algorithm benefits from the interdependence and interaction between them. Further our algorithm is designed such that the recognition stage only needs to tackle the out-of-plane pose variations as the in-plane orientation variations are handled during the detection and tracking stages. Illumination variations are explicitly taken care of using a normalization technique based on a reflectance model. We believe this funnel like feature where later, more sensitive stages need to process lesser variations would improve the accuracy of the system. We also introduce a new video database for evaluating video based detection and recognition algorithms. The training videos in the database also demonstrate the kind of image acquisition needed for a good performance of the system.

Contents

Abstract	i
1 Introduction	1
2 Literature Survey	2
3 Training : Face Detection and Tracking	3
3.1 Orientation Estimation	3
3.1.1 Randomization for Orientation Estimation	4
3.1.2 Adaboost SVM based scoring of bounding boxes	4
3.2 Kalman Filter	5
3.2.1 Mathematical Background	6
3.2.2 Formulation of Face Tracking as a Kalman Filtering Problem	7
3.3 Template Matching	8
3.4 Illumination Correction	9
3.5 Algorithm	11
4 Online Tracking and Recognition System	14
4.1 Appearance Manifolds	14
4.1.1 Approximating the conditional probabilities	15
4.2 Algorithm	16
5 Experiments and Results	19
5.1 Databases	19
5.1.1 Honda-UCSD database	19
5.1.2 Indian Database	19
5.2 Results	19
5.2.1 Training Stage	19
5.2.2 Recognition Stage	21
6 Conclusions	24
7 Possible Extensions	25
Bibliography	26

Chapter 1

Introduction

In recent years a lot of work has been done to enhance the performances of face detection, tracking and recognition individually. However, optimal design of a complete system for face recognition in videos, right from training data acquisition to automatic recognition of faces from test videos, has received negligible attention.

This relatively new paradigm of both learning and recognizing faces from videos poses new challenges but also results in some simplifications. The biggest challenge in automating this entire process is that each stage adds to the uncertainties already inherent in the system due to illumination and pose variations. Further, one has to choose between numerous options already available for performing detection, tracking and recognition separately, given that not all possible combinations are feasible either conceptually or computationally.

Having said that, the simplifications resulting from such automation could improve its performance while at the same time make such systems reach a wider consumer community. First of all using video modality for learning face models as in [1] naturally allows the capture of pose variations that the system is likely to encounter during testing. At the same time recognizing from videos makes it possible to integrate or fuse recognition results across multiple frames as demonstrated in [2] and [1]. Also the temporal correlation between the consecutive frames allows us to use face tracking, which is initialized and complemented by simple face detectors.

In this work we propose an end-to-end integrated face recognition system for video based applications. The integration starts right from the acquisition of training videos for learning the face models for recognition. In our proposed video database, the head movements in the training videos are not random, unlike the Honda/UCSD Video Database, but are chosen so as to capture a large subset of possible pose variations while at the same time simplifying the process of image acquisition and automated extraction of faces from these video. The recognition part closely follows the work of Lee et.al in [3] with an inclusion of a closely integrated frontal face detector and Kalman filter into the framework. Our algorithm also uses an illumination correction stage to counter the problem of illumination variations in the video yielding a significant boost in performance. In totality our approach aims at keeping the system complexity minimal while achieving near state-of-the-art results.

Our work can be broadly classified into 2 tasks:

- (1) Extraction of face images from training video dataset using a novel face detection and tracking algorithm and generating appearance manifolds [add ref] based on this face dataset
- (2) Based on the generated manifolds build a robust face tracking and recognition system for real world videos.

Chapter 2

Literature Survey

Face recognition in videos is challenging at multiple levels. The first challenge is to accurately localize faces in each frame in the presence of in-plane and out-of-plane head pose variations. Misalignments in face localization significantly deteriorate recognition performance. In [4] a popular frontal face detector has been proposed, which was later extended in [5] to detect faces with different poses and orientations. This work uses a decision tree to first estimate the out-of-plane pose in each window. It then uses perceptrons based on simple haar like rectangular features learned using adaboost for different combinations of that pose and in-plane rotations to draw the final bounding box around the face. While fairly robust, the training time and effort is quite large.

Another popular technique is the Incremental Visual Tracker [6] where instead of tracking a fixed target, a subspace based adaptive appearance model is learned for the target in an incremental online fashion. This was improved in [7] to deal with the problem of drift which occurs due to adaption of the appearance model to non-targets. While these may be elegant solutions for tracking of faces alone, we want to see if we could trade off tracking accuracy for a little simplicity, without bearing the cost of reduction in recognition performance. This choice would of course depend on the recognition algorithm.

In our algorithm we have considerably simplified the solution using just the frontal face detector proposed in [4], a motion model in the form of Kalman filter [8] and a simple randomization scheme involving HOG features and Adaboosted SVM. Other commonly used appearance based techniques for face tracking in videos are those based on Mean-Shift [9] and Particle filters [10], both of which are pixel intensity based approaches. Kalman Filter on the other hand is based on an adaptive motion model and hence would complement the pixel intensity based face detection better than either of them.

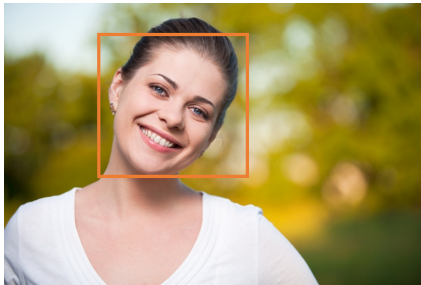
Our final face detection, tracking and recognition framework for test videos is similar to [3] with an extra addition of a frontal face detector and Kalman filter.

Chapter 3

Training : Face Detection and Tracking

Face detection and tracking in videos provide more flexibility as compared to face detection in still images. In our proposed algorithm we use a prediction-correction paradigm for robust face acquisition. The face localization in the current frame depends both on the current detection and the prediction based on the localization of the face in the previous frame using an adaptive motion model. Our algorithm comprises of 4 major components - (i) orientation estimation, (ii) Kalman filter based smoothing, (ii) template matching, and (iv) illumination correction. In the following sections first each of the components are individually explained. Then we sketch an outline of the algorithm for combining these components together for face detection and tracking.

3.1 Orientation Estimation



(a) Illustration of Viola-Jones Face Detection



(b) Orientation Tracking in Face Detection

Figure 3.1: A figure with two subfigures

Figure 3.1a Illustrates the inability of the commercially available face detectors to track the in-plane orientation of the face. However in order to reduce the number of poses that the recognition stage has to deal with, the desired tracking result is 3.1b so that the face image could be cropped and transformed to a view in which the face is vertical. The following subsections describe our approach to orientation estimation.

3.1.1 Randomization for Orientation Estimation

Utilizing the temporal continuity in a video, we use the detection in the previous frame to generate N candidate bounding boxes in the current frame. These candidate bounding boxes are sampled from a multivariate Gaussian Distribution with mean, $\mu = [x \ y \ s \ \theta]$ as the detection in the previous frame and an appropriately chosen covariance matrix Σ . Thus,

$$\rho_1, \rho_2, \dots, \rho_N \sim \mathcal{N}(\mu, \Sigma) \quad (3.1)$$

Note that the i^{th} random bbox is characterized by an angle with respect to the horizontal θ_i . If the frame is rotated by an angle $90 - \theta_i \ \forall i = 1 \dots N$, then for sufficiently large N , one of the bboxes would enclose the most vertical view of the face. This is illustrated in Figure 3.2.



Figure 3.2: Orientation Tracking in Face Detection

In order to choose the best candidate out of these N random bboxes, a scoring technique based on a learned adaboost SVM classifier is proposed. The bbox with the maximum score is chosen as the final estimate of the orientation of the face. The following section discusses this approach in detail.

3.1.2 Adaboost SVM based scoring of bounding boxes

A classifier needs to be learned in a supervised fashion to discriminate between vertical and non-vertical faces. The Head Pose Database (see Figure 3.3) consists of 2790 face images of 15 people with varying degrees of pan and tilt. The faces with either pan or tilt in the range $[-15, 15]$ are chosen as the positive training samples while the remaining images are used as negative examples.

The classifier is trained on the HoG (Histogram of Oriented Gradients) based features extracted from the images in the two classes. Since the pose variations within these 2 classes itself are quite large, a single linear support vector classifier is not sufficient and yields poor results. To overcome this problem, we use multiple weak linear classifiers combined together using the principle of Adaboost to obtain a strong classifier (see Figure 3.4). The main idea of Adaboost is that the sample which are misclassified in a given classifier are assigned higher

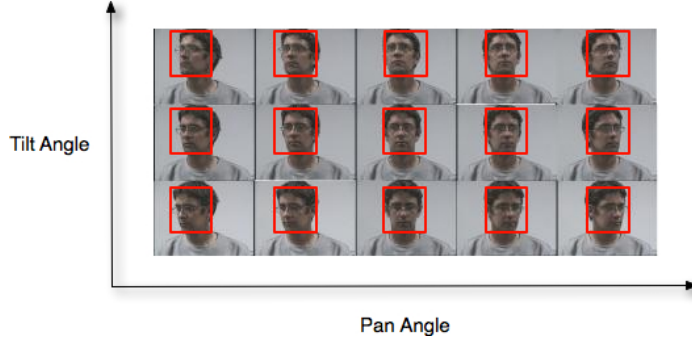


Figure 3.3: Head Pose Database

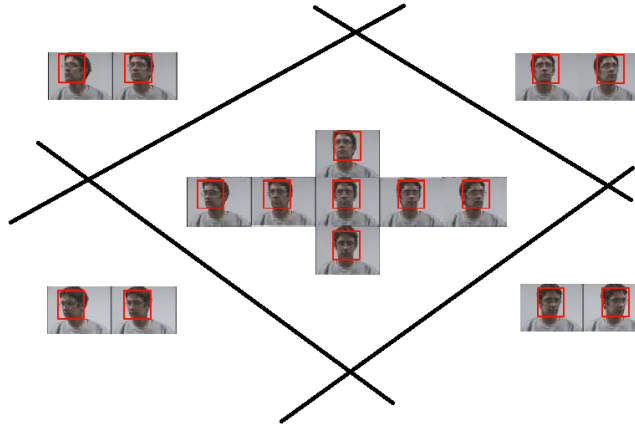


Figure 3.4: Illustration of combination of multiple weak classifiers

weights when training the successive classifiers. See Figure 3.5 for illustration. The score corresponding to an image with HoG feature z is given by

$$h(z) = \sum_{i=1}^T \alpha_i h_i(z) \quad (3.2)$$

The image with the highest score best corresponds to a vertical face image. Algorithm 1 describes the details of Adaboost algorithm

3.2 Kalman Filter

Kalman filter uses the equations of kinematics while accounting for statistical variations in the measurements and the model itself. Without delving into the derivation of the equations of Kalman filter we simply state the results in the following section.

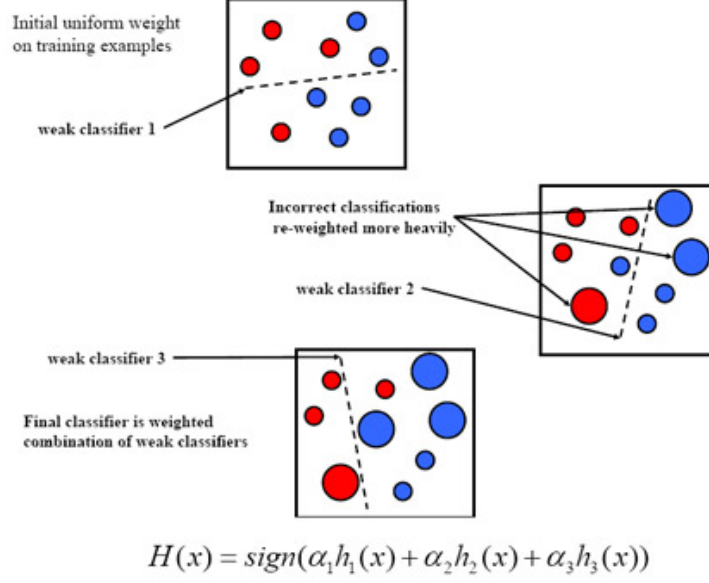


Figure 3.5: Illustration of Adaboost

Algorithm 1 Adaboost for combining multiple weak classifiers

- 1: **Input:** $\{x_1, \dots, x_n\} \sim D_1$ $\{y_1, \dots, y_m\} \sim D_2$
 - 2: Initialize $p_i = D_1(x_i) = \frac{1}{n}$ and $q_i = D_2(y_i) = \frac{1}{m}$
 - 3: Each sample denotes the HoG feature vector of the corresponding face image
 - 4: **Output:** Final score function $h(z) = \sum_{i=1}^T \alpha_i h_i(z)$
 - 5: **for** $t=1$ to T **do**
 - 6: Choose k samples from D_1 and D_2 each
 - 7: Train weak classifier using these samples: $h_t(z) = \text{sign}(a_t^T z + b_t)$
 - 8: Calculate error: $\epsilon_t = \sum_{i=1}^n p_i \mathbf{1}_{\{h_t(x_i) \neq 1\}} + \sum_{i=1}^m q_i \mathbf{1}_{\{h_t(y_i) \neq -1\}}$
 - 9: **if** $\epsilon_t < 0.5$ **then**
 - 10: $\alpha_t = \log\{\frac{1-\epsilon_t}{\epsilon_t}\}$
 - 11: $p_i = p_i e^{(-\alpha_t h_t(x_i))}$ and $q_i = q_i e^{(\alpha_t h_t(y_i))}$
 - 12: **else**
 - 13: Discard the current classifier h_t and repeat the iteration with different samples
 - 14: **end if**
 - 15: **end for**
-

3.2.1 Mathematical Background

It consists of two stages. The Predict stage uses the learned motion model and the past state to estimate the next state while the Update stage uses the measurements corresponding to the next state and uses it to correct the motion model as well as to give a better estimate of the state vector given the prediction and measurements. The mathematical representation of the two stages is as follows:

a) Predict Stage

A priori predicted state, $x_k^- = Ax_{k-1} + w_k$, where w_k is the AWGN process noise

A priori predicted estimate covariance, $P_k^- = AP_{k-1}A^T + Q$, where $Q = \mathbb{E}[w_k w_k^T]$

b) Update Stage

Measurement vector $z_k = Hx_k + v_k$, where v_k is the measurement noise

Optimal kalman gain, $K_k = P_k^- H^T (H P_k^- H^T + R)^{-1}$, where $R = \mathbb{E}[v_k v_k^T]$

A posteriori state estimate, $x_k = x_k^- + K_k(z_k - Hx_k^-)$

A posteriori estimate covariance, $P_k = (1 - K_k H) P_k^-$

In the context of face tracking the measurement z and state vectors x are defined as

$$z = [x \ y \ s \ \theta] \quad (3.3)$$

$$x = [x \ y \ s \ \theta \ \dot{x} \ \dot{y} \ \dot{s} \ \dot{\theta}] \quad (3.4)$$

Where x and y represent the top left coordinates of the bounding box and s is the size. The dotted counter parts are their time derivatives representing velocities. The following sections describe the meaning and initialization of all the parameter in the above equations.

3.2.2 Formulation of Face Tracking as a Kalman Filtering Problem

A is used to translate the equations of kinematics under uniform velocity into matrix form

$$x(t) = x(t - \Delta t) + \dot{x}\Delta t \text{ and } v(t) = v(t - \Delta t) \quad (3.5)$$

$$\implies x_k^- = \begin{bmatrix} \mathbb{I}_4 & \mathbb{I}_4 \\ 0 & \mathbb{I}_4 \end{bmatrix} x_{k-1} + w_k \quad (3.6)$$

$$\implies x_k^- = Ax_{k-1} + w_k \quad (3.7)$$

The acceleration term is accounted for by w_k the process noise which is assumed to be AWGN.

P , the estimate covariance matrix is a measure of the accuracy of the estimated state. It is updated by the filter over time, hence we need to only provide a reasonable initial value. Since we do not know the state at startup, we must set it to a large diagonal matrix with large values along the diagonal. This makes the filter prefer the measurements over the motion model which is inaccurate in the beginning. Hence

$$P_0 = I \cdot 10^4$$

Q is the process error covariance matrix i.e. $E[w_k \cdot w_k^T]$. w_k as mentioned is nothing but the acceleration term in the equations of kinematics

$$x(t) = x(t - \Delta t) + \dot{x} \cdot \Delta t + \frac{1}{2} \cdot a \cdot (\Delta t)^2 \quad (3.8)$$

$$\dot{x}(t) = \dot{x}(t - 1) + a \cdot \Delta t \quad (3.9)$$

$$\implies w_k = [\frac{\ddot{x}}{2} \ \frac{\ddot{y}}{2} \ \frac{\ddot{s}}{2} \ \frac{\ddot{\theta}}{2} \ \ddot{x} \ \ddot{y} \ \ddot{s} \ \ddot{\theta}]^T \quad (3.10)$$

Assuming that acceleration is same, a_x for x, y and s parameters but different, a_θ for θ parameter, we get

$$Q = \begin{bmatrix} \frac{a_x^2}{4} & 0 & 0 & 0 & \frac{a_x^2}{2} & 0 & 0 & 0 \\ 0 & \frac{a_x^2}{4} & 0 & 0 & 0 & \frac{a_x^2}{2} & 0 & 0 \\ 0 & 0 & \frac{a_x^2}{4} & 0 & 0 & 0 & \frac{a_x^2}{2} & 0 \\ 0 & 0 & 0 & \frac{a_\theta^2}{4} & 0 & 0 & 0 & \frac{a_\theta^2}{2} \\ \frac{a_x^2}{2} & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & \frac{a_x^2}{2} & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{a_x^2}{2} & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \frac{a_\theta^2}{2} & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.11)$$

Greater a_x and a_θ mean filter has to be more adaptive due to acceleration.

The measurement error covariance matrix R depends on the accuracy of the face detector. Let us assume that our face detector detects faces to within ± 10 pixels of the actual face location 95% of the time and that the error is Gaussian-distributed (which is a requirement of the Kalman filter). This implies that

$$2\sigma = 5 \implies \sigma^2 = 5 \quad (3.12)$$

Hence, assuming that the error variance is same for all components of the state vector the measurement error covariance matrix is given by

$$R = \mathbb{E}[v_k v_k^T] = \mathbb{I}_4 \times 5^2 \quad (3.13)$$

H defines the map between the measurement and the state vector as

$$z_k = Hx_k + v_k \quad (3.14)$$

$$\implies H = [\mathbb{I}_4 \ 0_4] \quad (3.15)$$

3.3 Template Matching

Template matching is a technique for finding small areas of an image(search image) that match to a template image. A basic method for template matching is to use a convolution mask, template. The convolution score is then used as the measure to identify the location of best possible match for the template in the given image.

One of the most common measure used in template matching to compare the similarity of different patches of input image with the template is **SAD**(Sum of Absolute Differences) [?]

The general setup of the template matching problem consists of an Input image(I), a template image(T) and a template matching box. Our goal is to locate the highest matching area.

To detect the best matching area, the template is compared against the source image by translating the origin of the template at each pixel of the input image (figure 3.6) and **SAD score** is calculated at each point. A pixel with coordinates (x_i, y_i) in the input image has intensity $I_i(x_i, y_i)$ and a pixel with the coordinates (x_t, y_t) in the template image has intensity $I_t(x_t, y_t)$. The absolute difference in the intensities is given by -

$$Diff(x_i, y_i, x_t, y_t) = |I_i(x_i, y_i) - I_t(x_t, y_t)| \quad (3.16)$$

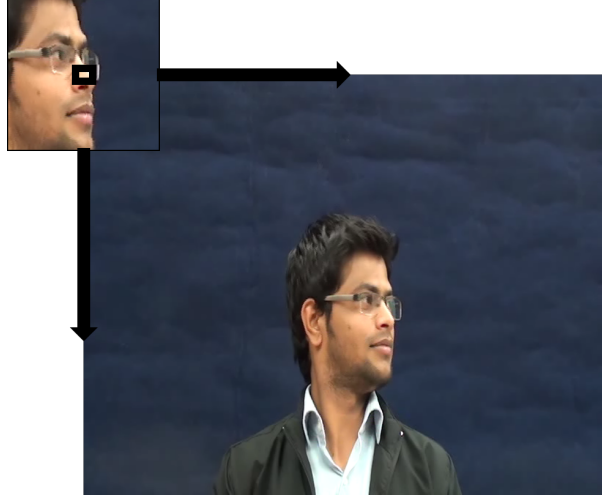


Figure 3.6: Transition of template over the input image

For a given position, (x,y) , of the origin of template the score is defined as -

$$SAD(x, y) = \sum_{i=0}^{T_r} \sum_{j=0}^{T_c} Diff(x + i, y + i, i, j) \quad (3.17)$$

where T_r, T_c denotes the number of rows and columns of template image.

The location with the lowest SAD score determines the best match of the template within the input image.

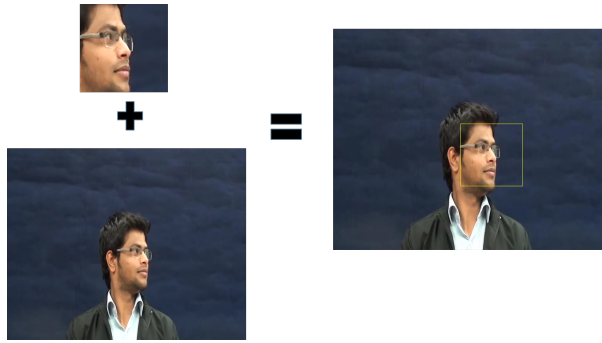


Figure 3.7: Sample Result for Template Matching

3.4 Illumination Correction

Human perception perceives any stimuli relative to the background rather than perceiving it in absolute terms. This idea forms the basis for the technique implemented here for nullifying the effect of variant illuminations on a face image. In [11] Chen *et al.* proposed a local descriptor called Weber Local Descriptor (WLD). It consists of 2 components: Differential Excitation and Orientation. Differential Excitation is calculated as a ratio of the intensity differences of current

pixels with its neighbors and intensity of the current pixel. Orientation captures the direction of the intensity variation in the neighborhood of the current pixel. We have made use of the differential excitation component of WLD to compute the ratio image from a given face image. The response for current pixel in output image is given by:

$$\varepsilon(x_c) = \arctan \left(\alpha \sum_{i=0}^{p-1} \frac{x_c - x_i}{x_c} \right) \quad (3.18)$$

where x_c is the centre pixel, x_i are the neighboring pixels, p is the number of neighboring pixels and α is used to adjust the intensity differences between neighboring pixels.

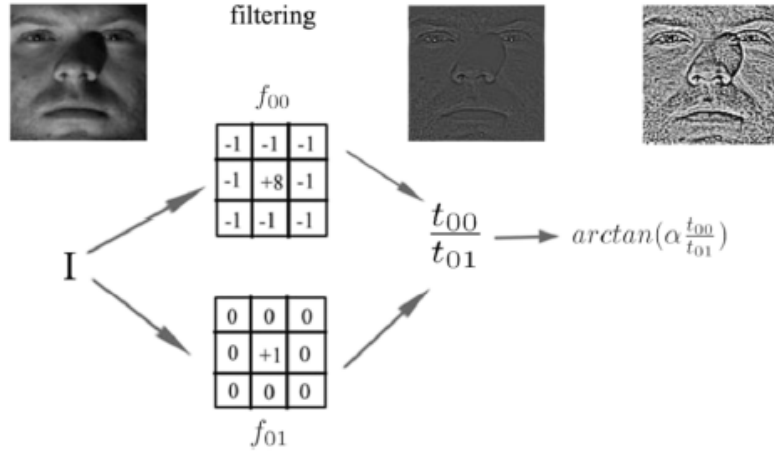


Figure 3.8: Illustration of computation of WLD [12]

According to the Lambertian reflectance model, a face image could be represented as:

$$F(x, y) = R(x, y)I(x, y) \quad (3.19)$$

where $F(x, y)$ is the image pixel value, $R(x, y)$ is a measure of reflectance and $I(x, y)$ denotes the illuminance at each pixel. Reflectance is largely determined by the facial surface texture and shape, hence, could be considered as the illumination insensitive part. Whereas illuminance depends only on the lighting source present.

As proved in [12], applying WLD to a face images $F(x, y)$ gives an illumination invariant representation of F known as “Weber-Face (WF)”:

$$WF(x, y) = \arctan \left(\alpha \sum_{i \in A} \sum_{j \in A} \frac{F(x, y) - F(x - i\Delta x, y - j\Delta y)}{F(x, y)} \right) \quad (3.20)$$

in which $A = \{-1, 0, 1\}$. From equation 3.19, we have

$$F(x - i\Delta x, y - j\Delta y) = R(x - i\Delta x, y - j\Delta y) \times I(x - i\Delta x, y - j\Delta y) \quad (3.21)$$

Illumination component is commonly assumed to vary very slowly, which gives us:

$$I((x - i\Delta x, y - j\Delta y) \approx I(x, y) \quad (3.22)$$

Substituting equations 3.19, 3.21, and 3.22 in equation 3.20, we get:

$$WF(x, y) = \arctan \left(\alpha \sum_{i \in A} \sum_{j \in A} \frac{R(x, y) - R(x - i\Delta x, y - j\Delta y)}{R(x, y)} \right) \quad (3.23)$$

It is evident from equation 3.23 that $\{WF(x, y)\}$ could be treated as an illumination insensitive representation of a face image F , as it comes out to be only dependent on the facial characteristics. This method thus eliminates the illumination part without actually estimating it, as in [13] or [14], which involves assumptions not applicable to real world scenarios.

As evident from the figure 3.9, the original images for a single individual having drastic variations in the lighting conditions, are hard to recognize even using the human perception/senses. Whereas the normalized representation of the images are fairly similar to each other for a given individual. Fig.3.9 shows the output of weber-faces normalization technique and Fig.?? shows the output for NL means normalization technique.

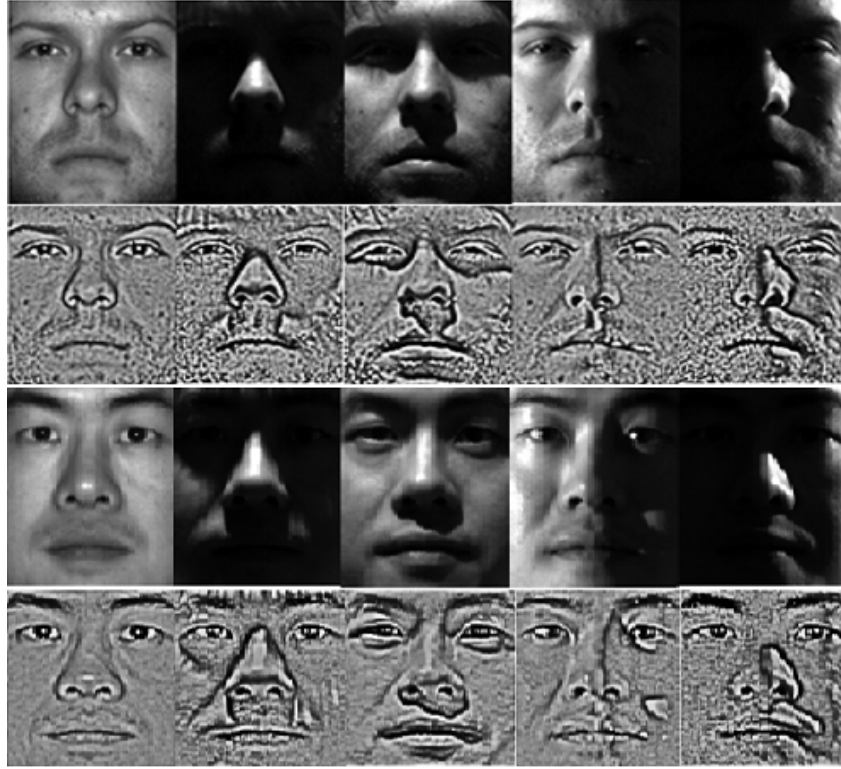


Figure 3.9: Samples from Yale B database for 2 different individuals and their corresponding Weber-faces

3.5 Algorithm

All the components defined in the previous sections are combined to model a complete face detection and tracking algorithm. The model could be visualized as in figure 3.10

The basic steps of the algorithm could be summarized as-

Algorithm 2 Face Detection and Tracking Algorithm

```
1: for each frame in the video do
2:   Input: The position parameters,  $[x, y, size, \theta]$ , of the bounding box around the face de-
      tected in the previous frame, say at time step  $(t - 1)$ 
3:   Position parameters,  $[x_i, y_i, size_i, \theta_i]$  of the 20 random bounding boxes are generated
      from a normal distribution with mean as the input and an appropriate sigma
4:   A priori prediction of the state of face in current frame is done using the Kalman Filter's
      prediction stage.
5:   for each of the 20 bounding boxes generated in line 3 do
6:     Rotate the frame by an angle of  $90 - \theta_i$ 
7:     Apply a Viola Jones face detector on the rotated frame
8:     if face is detected then
9:       it would act as a priori detection in the current frame
10:    else
11:      Template matching is applied with the detection from previous frame as the tem-
        plate
12:      Best match is used as a priori detection in the current frame
13:    end if
14:    Using this priori detection and the priori prediction from line 4, final prediction is
      done for the bounding box in consideration
15:    Illumination Correction is applied on this predicted face
16:    Adaboost score is calculated for this final prediction
17:  end for
18:  Maximum of the 20 adaboost scores is found and the corresponding bounding box is the
      detected face in the  $t^{th}$  frame
19:  This detection is then used as the input for frame at  $(t + 1)^{th}$  time step
20: end for
```

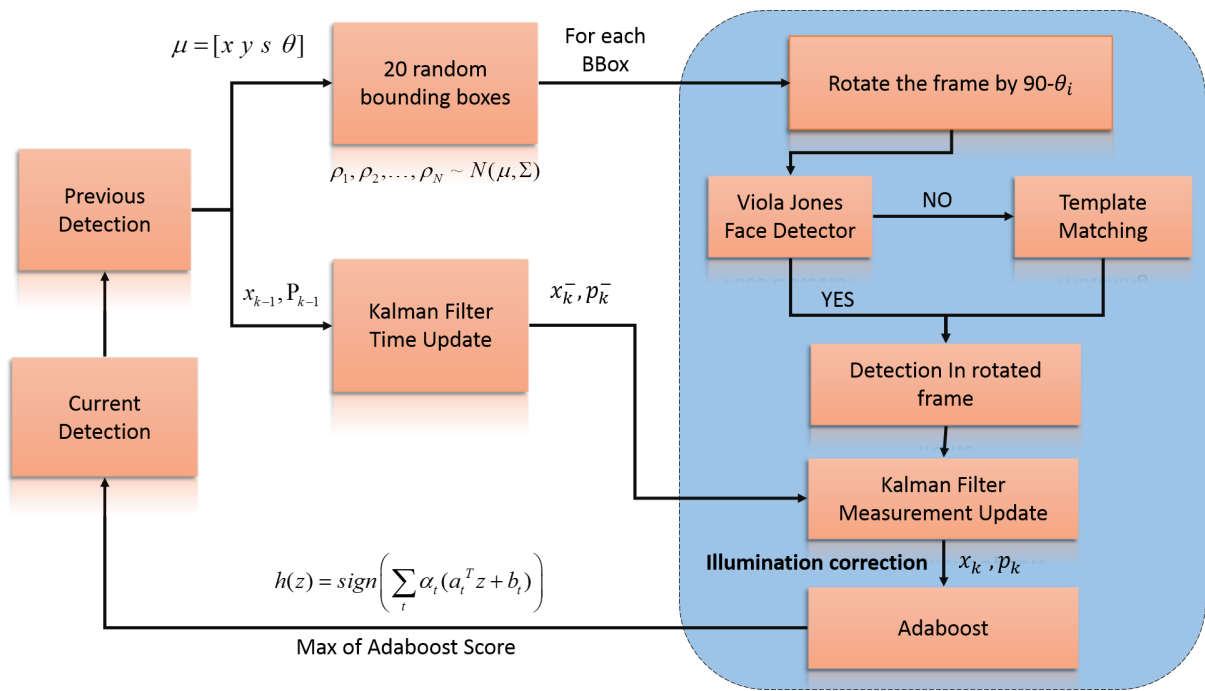


Figure 3.10: Detection and Tracking Algorithm

Chapter 4

Online Tracking and Recognition System

The previous chapter dealt with a generic method that could be applied for extraction of well tracked face images from a video. This chapter begins with the description of construction of Appearance Manifolds, a model for representation of face images of a person. Then it goes ahead to describe an algorithm that uses the appearance manifold for obtaining a score about the goodness of a crop of face image and also for recognition of person identity in the video. The face detection and tracking algorithm discussed before can be easily extended to this scenario.

4.1 Appearance Manifolds

In [1], Lee *et.al* presented a technique built around PCA to better incorporate the pose variations into subspace representation of face images of an individual. It also makes use of the fact that pose variations occur in a continuous fashion and uses the predictions in all the previous frames to recognize a person in the current frame.

To begin with, a given training sequence S_k of faces images of $person_k$ is partitioned into P parts by using K-means clustering with $K = P$. This step serves to segregate the different poses of the person in an unsupervised way. Pose cluster (C_k^i) for i^{th} pose is represented by a linear subspace L_k^i using PCA. The dimension of the subspace is maintained constant for all pose subspaces of all target individuals. This piecewise linear model of distribution of face images of a person in a low dimension vector space is called an Appearance Manifold M_k .

The task of recognizing a new face image is straight-forward at a conceptual level provided M_k is known accurately

$$k^* = \arg \min_k d_H(I, M_k) \quad (4.1)$$

$$\text{where } d_H(I, M_k) = \min_{x \in M_k} I - \hat{I}_x \quad (4.2)$$

Here \hat{I}_x is the image reconstructed from the low dimension feature vector. Since we are approximating the actual manifold using samples, we need to fall back on a probabilistic approach to estimate an optimal x^* in M_k such that \hat{I}_{x^*} is closest to I . To this end, the distance from manifold is defined as

$$d_H(I, M_k) = \int_{M_k} d(I_x, I) p_{M_k}(x|I) dx \quad (4.3)$$

where $p_{M_k}(x|I)$ is the conditional probability of x being the optimal point in M_k given face image I . Further, total probability theorem gives

$$p_{M_k}(x|I) = \sum_{i=1}^P p_{C_k^i}(x|I)P(C_k^i|I) \quad (4.4)$$

Substituting equation 4.4 in 4.3

$$d_H(I, M_k) = \sum_{i=1}^P P(C_k^i|I) \int_{C_k^i} d(I, x) p_{C_k^i}(x|I) dx \quad (4.5)$$

$$d_H(I, M_k) = \sum_{i=1}^P P(C_k^i|I) d_H(I, C_k^i) \quad (4.6)$$

where $P(C_k^i|I)$ is the conditional probability of x^* belonging to pose subspace C_k^i in manifold M_k . Equation 4.6 forms the fundamental equation to be used instead of equation 4.2 for recognizing the identity of the person in the face detected in the current frame at time t .

Since C_k^i has been approximated by a linear subspace L_k^i , therefore $d_H(I, C_k^i)$ is nothing but the $L2$ distance between I and the image reconstructed from its projection on L_k^i . Section 4.1.1 discusses how the term $P(C_k^i|I)$ can be calculated.

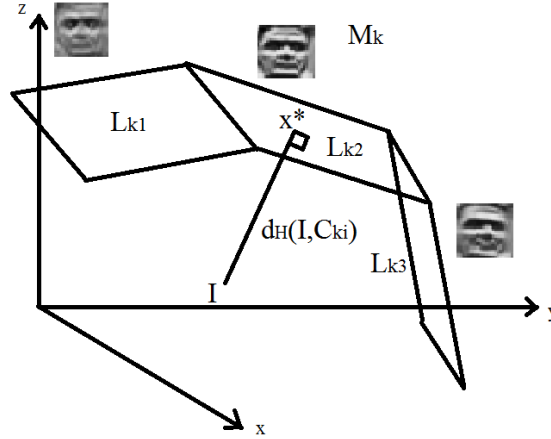


Figure 4.1: Representation of Manifold in 3 dimensional space

4.1.1 Approximating the conditional probabilities

At time t we have all the face images from $I_{0:t}$. So using Bayes' Theorem we get a recursive formula. iteratively $P(C_k^i|I)$ at time t can be expanded as

$$P(C_{k_t}^i|I_t, I_{0:t-1}) = \alpha P(I_t|C_{k_t}^i) \sum_{j=1}^P P(C_{k_t}^i|C_{k_{t-1}}^j) P(C_{k_{t-1}}^j|I_{t-1}, I_{0:t-2}) \quad (4.7)$$

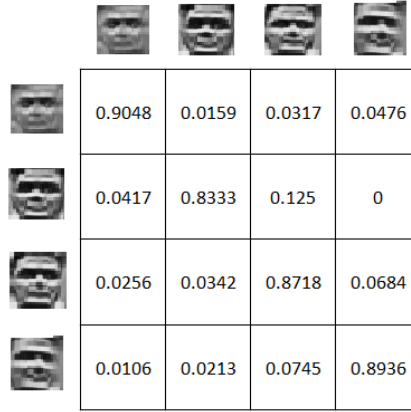
where α ensures that $\sum_{i=1}^P P(C_{k_t}^i|I_t, I_{0:t-1}) = 1$.

Here $P(I_t|C_k^i)$ is the probability that face image $I_t \in C_k^i$. Farther I_t is from subspace L_k^i , lower is the chance of its belonging to C_k^i . Hence this conditional probability can be conveniently calculated as

$$P(I_t|C_k^i) = \Lambda_k^t \exp\left(\frac{d_H^2(I_t, L_k^i)}{2 * \sigma^2}\right) \quad (4.8)$$

where Λ_k^t ensures that $\sum_{i=1}^P P(I_t|C_k^i) = 1$.

The term that actually captures the notion of temporal continuity in pose variations amongst frames is $P(C_{k_t}^i | C_{k_{t-1}}^j)$ which is the probability of x^* in the current frame lying in C_k^i given that in the previous frame it belonged to C_k^j . This can be easily obtained by counting all instances in training set S_k when an image in C_k^i followed another in C_k^j . Again normalization is essential to ensure $\sum_{i=1}^P P(C_{k_t}^i | C_{k_{t-1}}^j) = 1$ at every time instant.











				
	0.9048	0.0159	0.0317	0.0476
	0.0417	0.8333	0.125	0
	0.0256	0.0342	0.8718	0.0684
	0.0106	0.0213	0.0745	0.8936

Figure 4.2: Transition Matrix

4.2 Algorithm

Online tracking and recognition algorithm could be visualized in the form of a flow chart as shown in 4.3. The basic steps of this task are summarized in Algorithm 3 -

Algorithm 3 Online Tracking and recognition algorithm

```
1: for each frame in the video do
2:   Input: The position parameters,  $[x, y, size, \theta]$ , of the bounding box around the face de-
      tected in the previous frame, say at time step  $(t - 1)$ 
3:   Position parameters,  $[x_i, y_i, size_i, \theta_i]$  of the 20 random bounding boxes are generated
      from a normal distribution with mean as the input and an appropriate sigma
4:   A priori prediction of the state of face in current frame is done using the Kalman Filter's
      prediction stage.
5:   for each of the 20 bounding boxes generated in line 3 do
6:     Rotate the frame by an angle of  $90 - \theta_i$ 
7:     Apply a Viola Jones face detector on the rotated frame
8:     if face is detected then
9:       It would act as the measurement vector in the current frame
10:    else
11:      The random bounding box is used as the measurement vector in the current frame
12:    end if
13:    Using this measurement vector and the priori prediction from line 4, final prediction
      is done for the bounding box in consideration
14:    Illumination Correction is applied on this predicted face
15:    A score is generated by calculating the distance of this image from the closest pose
      subspace of the identified persons manifold in the previous frame
16:  end for
17:  Minimum of the 20 appearance manifold distances is found and the corresponding
      bounding box is the detected face in the  $t^{th}$  frame
18:  This detected face is recognized using the appearance manifold
19:  This detection is then used as the input for frame at  $(t + 1)^{th}$  time step
20: end for
```

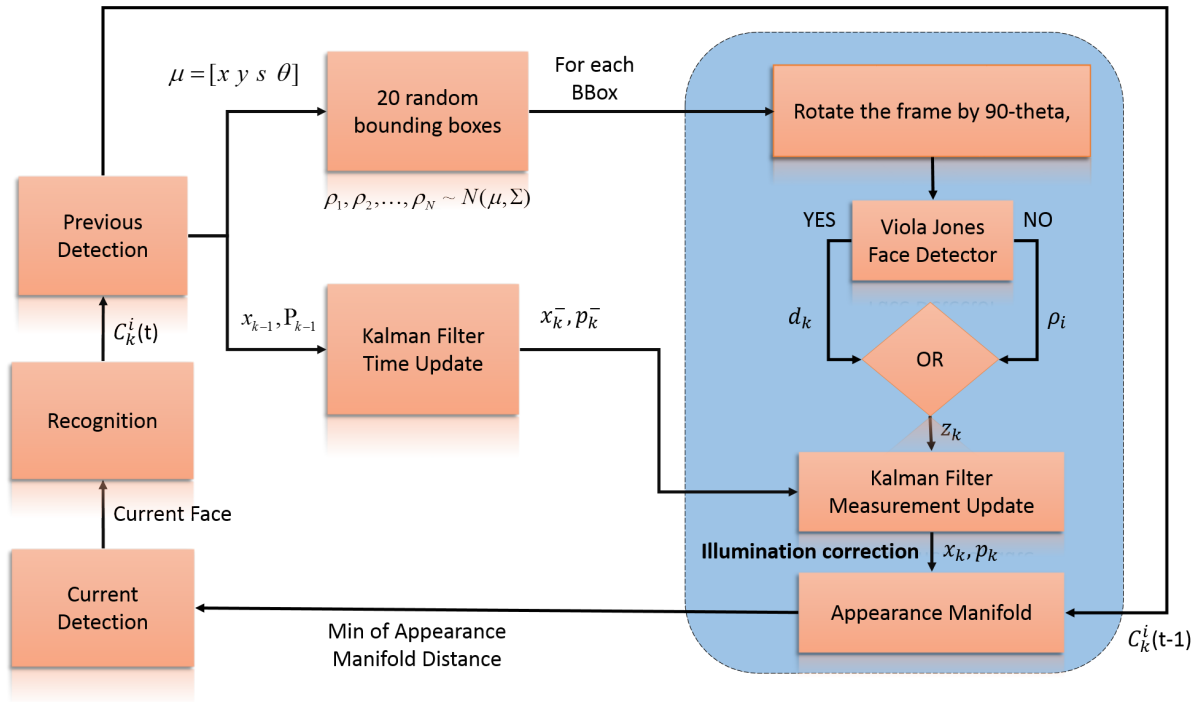


Figure 4.3: recognition

Chapter 5

Experiments and Results

5.1 Databases

5.1.1 Honda-UCSD database

It is a standard video database for evaluating face tracking/recognition algorithms. Video sequences are recorded at 15 frames per second in an indoor setup with a resolution of 640×480 . Each individual rotates his/her head at variable speeds and contains large variations in and out-of plane head movements and facial expressions. Every individual is recorded in at least 2 sequences, thus providing a training and a testing set of videos.

5.1.2 Indian Database

The goal of this Indian Database is to provide a video database for evaluating face detection/tracking algorithms on Indian faces. Indian faces generally show large variations especially in terms of skin color and facial hair. Videos are recorded for 30 different individuals in an indoor environment at 25 frames per second using a Sony CX-110 camera at ACES, IIT Kanpur. The resolution of each video sequence is 720×576 .

Similar to the Honda-UCSD database, each individual is recorded in at least 2 sequences, thus providing a training and a testing set of videos. This dataset contains large variations in terms of in-plane and out-of plane head movements. Unlike Honda-UCSD dataset, it contains structured variations in head movements in the training dataset for easy acquisition of face images. This allows us to get a wide range of different poses. The main motivation behind collecting this dataset is to evaluate the algorithms in an Indian setup as there is no such database available till date.

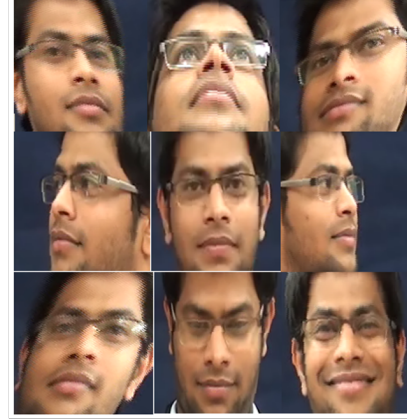
5.2 Results

5.2.1 Training Stage

From table 5.1 it can be seen that the standard viola-jones based algorithm for face detection yields a poor detection accuracy, i.e. it is able to correctly detect only about half of the total number of faces in the Honda-UCSD video. Further since it does not adapt to the orientation of the face, the actual number of faces that could be used in construction of appearance manifold



Honda\UCSD Video Database



Indian Video Database

Figure 5.1: Various face poses extracted using the proposed algorithm



(L) : Tracking without using
Template Matching

(R) : Tracking using
Template Matching

Indian Video Database



(L) : Tracking without using
Template Matching

(R) : Tracking using
Template Matching

Honda\UCSD Video Database

Figure 5.2: Effect of Template Matching on tracking performance

is actually even lower. Our proposed algorithm both with and without template matching beat this by a large margin. This means that our algorithm is able to extract more number of usable faces from the same video. Also the performance of all the three approaches improves on our proposed Indian Database. This is largely due to the structured nature of the head movements in the training videos of our database, that allow easier and reliable tracking without compromising on the range of pose variations captured. As per our intuition, our tracking algorithm employing template matching outperforms all others, because even when the face is not detected by the viola-jones detector, we are able to track it using the previous detector and the Kalman Filter.

Boxplot shown in Figure 5.3 is a representation for the variation in detection accuracy observed in different training videos for the different algorithms. Our algorithm with template matching achieves detection rates of more than 85% on 50% of the training videos in Honda-UCSD database and more than 90% on 50% of the training videos in Indian Database.

Table 5.1: Average value of tracking accuracy

Databases	Honda UCSD Database			Indian Database		
	Without Template Matching	Template Matching	Viola Jones Detector	Without Template Matching	Template Matching	Viola Jones Detector
Accuracy	74.79	82.37	55.98	79.27	86.41	70.39

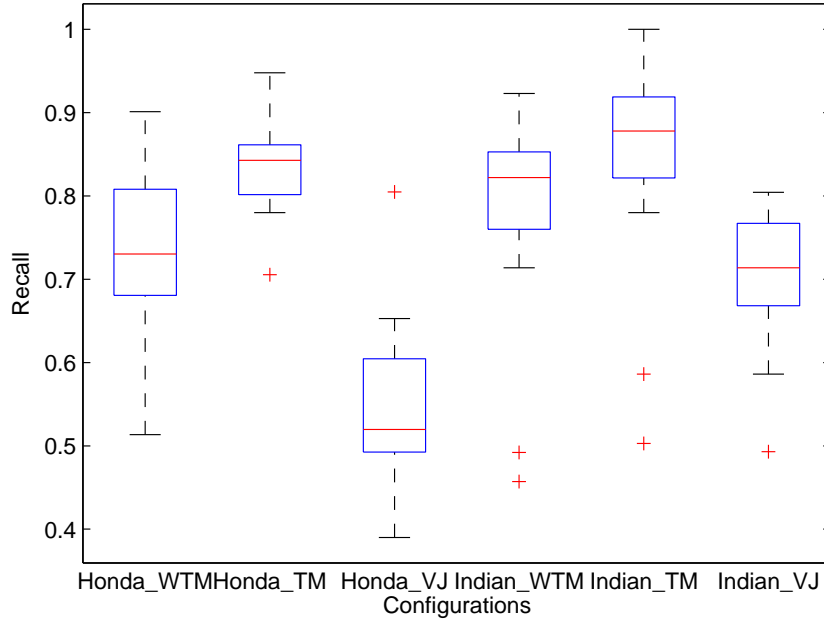


Figure 5.3: Box Plot for Tracking Accuracy of various configurations

5.2.2 Recognition Stage

Table 5.2 shows the recognition accuracies of the final system for different methods of face image acquisition from the training videos. Again since template matching resulted in extraction of more number of faces, it yields higher accuracy than the method without template matching. Note that the accuracy here is the fraction of good detections that are correctly recognized.

Box plot in Figure 5.6 shows that the template matching based approach yields a recognition accuracy of more than 90% for 50% of the test videos in the Honda-UCSD database and more than 85% for 50% of the test videos in the Indian Database. Thus there is a significant improvement of 5-10% from the approach without template matching.



Figure 5.4: Results of online tracking on Indian video database



Figure 5.5: Results of online tracking on HONDA/UCSD video database

Table 5.2: Average value of accuracy for recognition

Databases	Honda UCSD Database		Indian Database	
	Without Template Matching	Template Matching	Without Template Matching	Template Matching
Accuracy	82.2	85.86	73.19	77.49

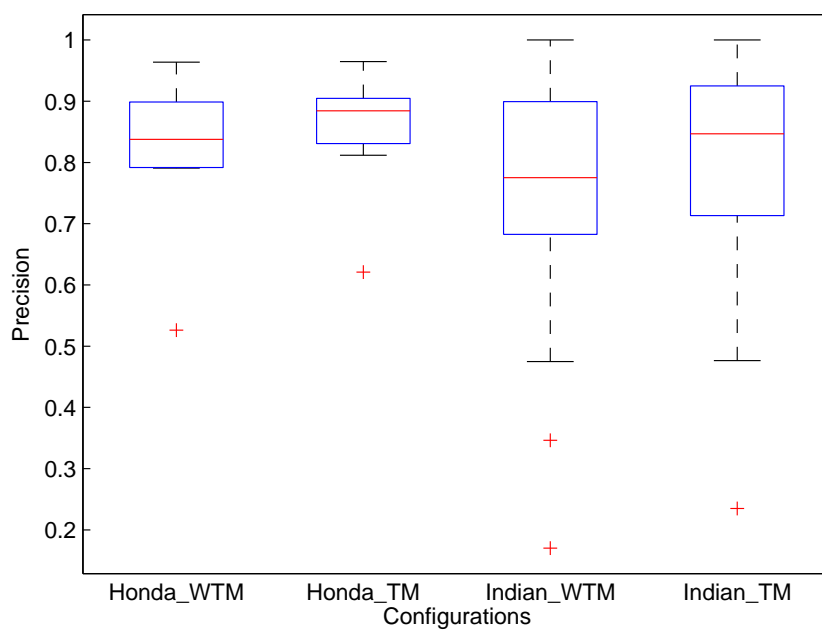


Figure 5.6: Box Plot for Recognition Accuracy of various configurations

Chapter 6

Conclusions

In this work we have built an end-to-end integrated face detection, tracking and recognition system. To the best of our knowledge this system outperforms the current state-of-the-art face detector and tracking systems in terms of accuracy. Most of the current systems require the training database for face recognition to be provided in the form of manually cropped images. However our system is capable of extracting the faces from the videos automatically, from which the false detections could be manually removed. This makes this system a potential candidate for small to medium surveillance security systems, specially for homes or personal work spaces. Further with our Indian Database we try to establish a new paradigm in database acquisition where special structure could be leveraged to simplify face image acquisition procedure for training database.

Bibliography

- [1] K.-C. Lee, J. Ho, M.-H. Yang, and D. Kriegman, "Video-based face recognition using probabilistic appearance manifolds," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 1, pp. I–313, IEEE, 2003.
- [2] G. Shakhnarovich, J. W. Fisher, and T. Darrell, "Face recognition from long-term observations," in *Computer Vision ECCV 2002*, pp. 851–865, Springer, 2002.
- [3] K.-C. Lee, J. Ho, M.-H. Yang, and D. Kriegman, "Visual tracking and recognition using probabilistic appearance manifolds," *Computer Vision and Image Understanding*, vol. 99, no. 3, pp. 303–331, 2005.
- [4] P. Viola and M. Jones, "Robust real-time object detection," *International Journal of Computer Vision*, vol. 4, pp. 34–47, 2001.
- [5] M. Jones and P. Viola, "Fast multi-view face detection," *Mitsubishi Electric Research Lab TR-20003-96*, vol. 3, p. 14, 2003.
- [6] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 125–141, 2008.
- [7] M. Kim, S. Kumar, V. Pavlovic, and H. Rowley, "Face tracking and recognition with visual constraints in real-world videos," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, IEEE, 2008.
- [8] C.-Y. Seong, B.-D. Kang, J.-H. Kim, and S.-K. Kim, "Effective detector and kalman filter based robust face tracking system," in *Advances in Image and Video Technology*, pp. 453–462, Springer, 2006.
- [9] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, vol. 2, pp. 142–149, IEEE, 2000.
- [10] K. Nummiaro, E. Koller-Meier, and L. Van Gool, "An adaptive color-based particle filter," *Image and vision computing*, vol. 21, no. 1, pp. 99–110, 2003.
- [11] J. Chen, S. Shan, C. He, G. Zhao, M. Pietikainen, X. Chen, and W. Gao, "Wld: A robust local image descriptor," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 9, pp. 1705–1720, 2010.

- [12] B. Wang, W. Li, W. Yang, and Q. Liao, "Illumination normalization based on weber's law with application to face recognition," *Signal Processing Letters, IEEE*, vol. 18, no. 8, pp. 462–465, 2011.
- [13] H. Wang, S. Z. Li, and Y. Wang, "Face recognition under varying lighting conditions using self quotient image," in *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, pp. 819–824, IEEE, 2004.
- [14] D. J. Jobson, Z.-u. Rahman, and G. A. Woodell, "A multiscale retinex for bridging the gap between color images and the human observation of scenes," *Image Processing, IEEE Transactions on*, vol. 6, no. 7, pp. 965–976, 1997.