A Novel Approach for Person Authentication and Content-based Tracking in Videos using Kernel Methods and Active Appearance Models*

Nimit Kumar†
Department of Electrical Engineering
Indian Institute of Technology,
Kanpur, UP, INDIA.
nimit.kumar@ieee.org

Vibhanshu Abhishek
Department of Computer Science & Engineering
Indian Institute of Technology,
Kanpur, UP, INDIA.
vibhanshu@acm.org

Gagan Gautam
Department of Computer Science & Engineering
Indian Institute of Technology,
Kanpur, UP, INDIA.
gagan@ieee.org

Abstract – *A novel integration of methods for person authentication and tracking is proposed for real time security systems. The implementation of the idea for this real time implementation follows a three step procedure – face detection, recognition and content-based tracking. Instead of analyzing continuous videos we sample the frame based on a method derived from Shannon's information theory model. The Face-detector detects multi-viewed faces in a video using feature-based kernel methods in a reduced feature space obtained using ICA. The identified "face regions" are then passed on to the face recognition system which is based on Active Appearance Models (AAM). Once the subject is recognized, it can be tracked in the video using kernel based object tracking method. Several space reduction techniques have been used like ICA, PCA and skin-color segmentation.*

Keywords: Kernel-based learning, SVM, ICA, Active Appearance Models, Information Theory, Mean-Shift Tracking.

1 Introduction

Face Authentication in security systems have been constantly improving in recent years. A Face Authentication system in real time was proposed in [12] by Sukthankar and Stockton. The authors have implemented an agent based method for integrating Face Detection and Recognition using a multiagent pipelined system. Though Argus is reported to perform well on regular visitors, it often misclassifies people. In the proposed work, we emphasize on the accuracy of proposed method and the scope of real time implementation. Kernel-based learning method, like SVM, has been shown to perform better than Neural Network and other similar methods in its ability to give better generalization ability [13]. This follows from the Structural Risk Minimization principle which minimizes both the *expected risk* and the *empirical risk*. The Face Detection System is explained in details in Section 2. We also discuss methods adopted to increase the computational speed of the face detection process. A reduced feature space using Independent Component Analysis [1] is implemented. In addition to this we apply a skin filter, which rejects most of the non-skin region. The skin-filter is applied in RGB color space and uses the properties associated with skin regions.

Active Appearance Models (AAM) was proposed by Cootes et al [4] and is one of the more sophisticated deformable template models. Active Appearance Model is a very robust approach which can be used to model deformable objects such as face. The primary advantage of AAMs is that a priori knowledge is learned through observations of both shape and texture variation in a training set. A brief discussion of AAMs and its improved version used in our work is given in Section 3.

In [2], the authors have introduced a kernel-based method for object tracking. In this work, kernels with Epanechnikov profile are recommended to be used. Section 4 gives a brief discussion for kernel-based face tracking.

The integration of above mentioned steps along with the notion of Information Theory [8] results in the final face authentication system. This system can be used either in real time or in videos and is discussed in Section 5.

2 Face Detection using SVM in a reduced feature space

Kernel-based learning algorithms, in particular Support Vector Machines (SVM), have been used for Face Detection with success [8]-[10]. They are mainly motivated by Vapnik's Statistical Learning Theory [13]. We construct our face detection system mainly using SVM as the classifier. In addition, ICA is used to identify independent basis for the face space, thus giving a reduced feature space. Another factor which serves in increasing the computational speed is application of a relevant skin-filter as discussed below.

* **0-7803-7952-7/03/$17.00 Ó 2003 IEEE.**

† Contact Author: Nimit Kumar, Senior Undergraduate, Dept. Of Electrical Engineering, Indian Institute of Technology, Kanpur, KANPUR 208016, UP, INDIA. Phone : (91) 512 2597198. Fax : (91) 512 2590063. email: nimit.kumar@ieee.org

2.1 Support Vector Machines (SVM)

A support vector machine (SVM) maps the input vectors of a sample space into the higher dimensional feature space through a nonlinear mapping so that an optimal separating hyperplane can be constructed. In this sense the SVM works in the same spirit as the Rosenblatt's perceptron introduced way back in 1956 by Frank Rosenblatt [5]. However, the success of the SVM lies in the selection of the aforementioned nonlinear mapping function, which must be selected prior to the optimization involved in the SVM method. To avoid this explicit dependence over the nonlinear mapping function, the idea of the *kernel function* was proposed inspired from the Hilbert-Schmidt theory. The kernel function here defines an inner product on a Hilbert space.

Given a training set of samples as in (1),

$$S = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_l, y_l)\} \subseteq (X \times Y)^l \quad (1)$$

The generalized Support Vector Machine solves the following dual quadratic optimization problem in order to obtain an optimal separating hyperplane.

$$\max_{a} W(\vec{a}) = \sum_{i=1}^{l} a_i - \frac{1}{2} \sum_{i,j=1}^{l} y_i y_j a_i a_j K(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

subject to $0 \le a_i \le C,\ i = 1,2,\ldots,l$ (2)

$$\sum_{i=1}^{l} y_i a_i = 0$$

Solution to (2) leads to the maximization of the margin of two classes and minimization of the training error simultaneously. This is made possible because of the Structural Risk Minimization [11] principle.

This results in the construction of the hyperplane given by (3).

$$f(x) = \sum_{j=1}^{s} y_j a_j K(x, z_j) + b \quad (3)$$

where z_j is the Support Vector belonging to class y_j with the Lagrange multiplier being equal to a_j. $K(x, z_j)$ is the kernel function which introduces the implicit mapping between the input space and feature space.

2.2 Feature space reduction using ICA

Independent Component Analysis (ICA) is a generalization of PCA by imposing statistical independence on the individual components of the output vector and has no orthogonality constraint. It must be noted that PCA is a special case of ICA with Gaussian source models.

Consider a linear mixture model of unknown independent sources, s as given in (4).

$$\mathbf{x} = \mathbf{A} \cdot \mathbf{s} \quad (4)$$

where A is unknown and x is observed signal. ICA gives an estimate of the demixing matrix W such that the original source signals can be recovered from the observed signals x using the reconstruction model in (5).

$$\mathbf{y} = \mathbf{W} \cdot \mathbf{x} \quad (5)$$

The objective is to obtain W through a learning algorithm so as to minimize the Kullback-Leibler divergence between the source signal vector s and its estimate y. This leads to the following learning algorithm:

$$\mathbf{Q} = \mathbf{I} - \mathbf{g}(\mathbf{y}(n))\mathbf{y}^T(n) \quad (6)$$

$$\mathbf{W}(n+1) = \mathbf{W}(n) + z(n)\mathbf{Q}\mathbf{W}^T(n) \quad (7)$$

where $\mathbf{I}$ is the identity matrix, $z(n)$ is the learning rate, and $\mathbf{g}(\mathbf{y}) = (g(y_1), \ldots, g(y_n))^T$ is a nonlinear function.

Using (6) and (7), we obtain statistically independent basis images. The basis images form the reduced feature space. The ICA method adopted for obtaining these basis images is same as that in [1]. In our system, we consider a masked face region of 20 x 20 pixels. Thus, we have a 400-dimensional. We consider 200 principal components of the face training set. Thus the dimensionality is reduced to half, speeding up the detection process considerably. A Matrix, *X,* with 200 rows and *n* columns, with *n* equal to the number of training faces, is constructed. Performing ICA on X gives us 200 independent basis images. This is same as Architecture I proposed in [1]. All faces are masked and then histogram normalized. The CMU Face Database is used for classifier construction. In addition to using the reduced feature space, we also use a novel skin-filtering method which rejects most of the non-skin regions, thus speeding up the process. This and some other heuristics implemented are discussed in the next sub-section.

2.3 Pre-processing using Skin-color segmentation

(a) (b)

Figure 1. (a) Shows the image on which the skin tone segmentation is applied. (b) shows the resulting image where non face regions have been removed.

It has been observed that the skin color of people from varied ethnicity clusters tightly in hue-saturation (HS) space. A pixel is classified as skin or non-skin based on the HSI parameters. Most of the skin tones lie between and $q_1 = 36^\circ$ and $q_2 = 6^\circ$ as shown in Figure 2.

Very often there is an overlap of the skin color space and background color space. Since skin color segmentation is used for reduction of the search space this does not present a major problem for us as the face

detection system elegantly discards the non face regions. The RGB pixels are converted to HSI space and all pixels having q value 36° to 6° are classified as skin and the mask is applied on the original image.

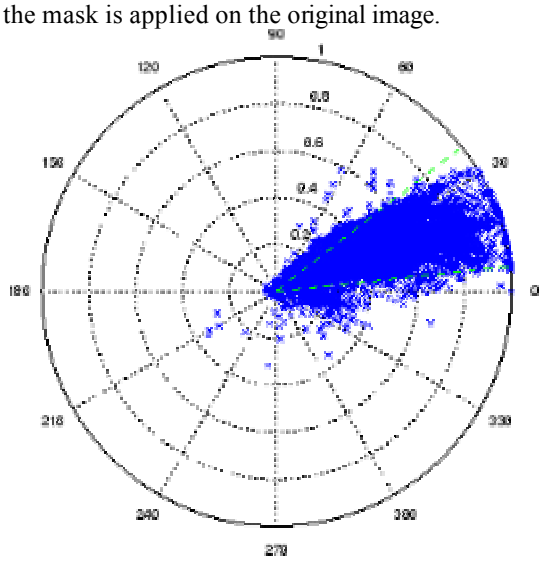

Figure 2.Skin Pixels plotted on HS-space. The radial coordinates represents saturation and the polar coordinates represents hue.

It is only the skin marked regions that are passed on to the face detection system. The effectiveness of the method depends on the nature of the image under consideration. However, on average applying the skin filter leads to 75-80% reduction in search space and there is an increase in speed by about two orders of magnitude in the detection and recognition process.

In addition to masking the scanned window image, we also make sure that a certain minimum of pixels lie in the window, in which case it is passed for detection. This considerably speeds up the detection system.

The methods described above are integrated into the IIT-Kanpur Face Analysis System.

3 Face Recognition using Active Appearance Models

Active Appearance Model is a very robust approach which can be used to model deformable objects such as face. The model is described as a set of parameters which can be varied between certain limits to obtain a viable instance of the model matching the input example.

Developing an AAM involves a general three step process - capturing the data, normalizing the data and finally formulating and describing the data in a statistical framework. The process is to first develop a statistical shape model then a statistical texture model then combining these two to get the Active Appearance Model. For statistical analysis Principal Component Analysis is performed to obtain a constrained and compact description of the data.

3.1 Shape Modeling

To obtain the shape information of an object we annotate its image with certain landmarks. Landmarks are chosen as those points which can easily differentiate between two different instance images of the same object, obtained under different deforming conditions. This method is completely manual.

Figure 3. Extracting the shape vector

This gives us a vector of Landmark coordinates describing the shape of the image. This is done for the whole example set giving us a set of vector. These vectors are then normalized and then principal component analysis is applied on them to obtain a set of eigen vectors describing the shape variation space. The PCA is done as shown below.

Compute the *Mean* of the data:

$$\bar{\boldsymbol{x}} = \frac{1}{s}\sum_{i=1}^{s} \boldsymbol{x}_i \tag{8}$$

where $\boldsymbol{x}_i$ is the shape vector for the ith image and $\bar{\boldsymbol{x}}$ is the mean.

Compute the *covariance* of the data :

$$S = \frac{1}{s-1}\sum_{i=1}^{s}\left(\boldsymbol{x}_i - \bar{\boldsymbol{x}}\right)\bullet\left(\boldsymbol{x}_i - \bar{\boldsymbol{x}}\right)^{\mathrm{T}} \tag{9}$$

where S is the covariance matrix.

Compute the eigenvectors Φ_i and corresponding eigenvalues l_i of S. If Φ contains $\mathbf{t}$ eigenvectors corresponding to $\mathbf{t}$ eigenvalues, then we can approximate any example of the training set by:

$$\boldsymbol{x} \approx \bar{\boldsymbol{x}} + \Phi \boldsymbol{b} \tag{10}$$

where $\Phi = \left[\Phi_1, \Phi_2, \ldots \Phi_t\right]$ and $\boldsymbol{b}$ is a t dimensional vector given by (11).

$$\boldsymbol{b} = \Phi^{\mathrm{T}}\left(\boldsymbol{x} - \bar{\boldsymbol{x}}\right) \tag{11}$$

3.2 Texture Modeling

Each of the example images is warped onto the mean shape obtained from the shape Model using the Delaunay Triangulation Method. Extracting the texture from the shape free warped image gives the texture vector for each of the example images. PCA is applied on these vectors to obtain a set of eigenvectors describing the texture variation space.

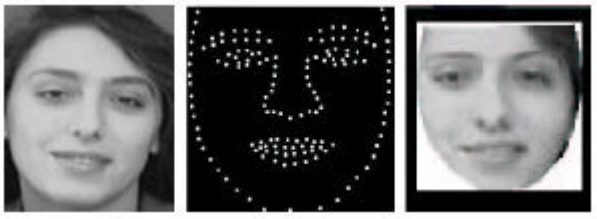

Figure 4. The figure shows the warping of image on the mean shape to obtain shape free texture vector. Left: Actual Image, Middle: Mean Shape, Right: Warped Image.

The samples are normalized to reduce the effect of global light variations as follows: Let $\mathbf{g}_{im}$ be the intensity vector that has been scanned from $\mathbf{i}^{th}$ image, we apply the following transformation to $\mathbf{g}_{im}$ to obtain the normalized vector,

$$\mathbf{g} = (\mathbf{g}_{im} - ß.I)/a \tag{12}$$

where a and ß are given by (13).

$$a = \mathbf{g}_{im}.\bar{\mathbf{g}}, \; ß = (\mathbf{g}_{im}.I)/n \tag{13}$$

where n is the number of elements in the texture vector, $\bar{\mathbf{g}}$ is the mean and I is the identity matrix.

Obtaining the mean of the normalized data is then a recursive process, as the normalization is defined in terms of the mean. A stable solution can be found by using one of the examples as the first estimate of the mean, aligning the others to it and re-estimating the mean and iterating.

3.3 Combined Model (AAM)

For each of the example images shape and texture model parameters are obtained. They are the combined to form a single vector for each of example image. PCA is then run on these set of vectors to obtain a set of eigenvectors describing the variation space of the Active Appearance models. The coefficients of these vectors describe an instance of the model.

3.4 AAM Search

Given an example image, we have to search for proper model parameters which can describe the example image in terms of the model as accurately as possible. Matching criteria is determined by difference in the normalized texture vector of the model and the actual image. This search is a high dimensional search problem which cannot be done in real-time. To tackle this problem we need some guidance mechanism in our search process.

The difference vector is defined as in (14).

$$dI = I_i - I_m \tag{14}$$

Where I_i is the vector of grey-level values in the image, and I_m is the vector of grey-level values for current model parameters.

To locate the best match between model and image, we wish to minimize the magnitude of the difference vector, by varying the model parameters C. For this we adopt prior knowledge approach. There are two parts to the problem: learning the relationship between dI and the error in the model parameters, dC, and using this knowledge in an iterative algorithm for minimizing the difference. For this the correlation between the texture difference vectors and the model parameters difference vectors between the actual and synthesized images is used. For this a linear relationship between the model parameter displacements and the Grey level texture difference vector is developed by using some training examples.

Taking a training image for which the shape and appearance parameters are known, we calculate its exact Combined Model parameters using the relation in (11). Φ is the eigenvector matrix calculated building the model. $\bar{\boldsymbol{x}}$ is the mean of all the shape and appearance parameters and $\boldsymbol{x}$ is the shape and mean parameters of the training image. Thus $\boldsymbol{b}$ contains the combined appearance parameters. We also have pose parameters for the training image.

Now each of these parameters are deliberately changed by some amounts, one by one and the corresponding texture difference vector between the Model generated image and the original image is calculated. Thus for each dC change in parameters we have a corresponding texture difference vector. Now we have to determine an empirical relationship between them.

$$C = R \cdot G \tag{15}$$

Where C is the Matrix formed by parameter change vectors, G is the Matrix of corresponding texture difference vectors, and R is the regression Matrix which is determined by solving (15).

In each iteration of the search process we calculate the texture difference vector between the model generated image and the input image. The corresponding change in parameters is obtained by (16).

$$dC = R \cdot dg \tag{16}$$

Thus using these parameters a new image is generated which is more close to the input image than the one in previous iteration. Thus several iterations are performed until the difference in texture values between the Model generated image and the input image is substantially small.

3.5 Classification

Given a new example of a face and extracted model parameters, the aim is to identify the individual in a way which is invariant to confounding factors such as lightning, pose and expression. Since we have a representative training set of face images, we can do this using Mahanolobis distance measure which enhances the effect of inter class variations (identity) whilst suppressing the effect of within class variations (pose, lightning, expression). This gives a scaled measure of the distance of an example from a particular class. The Mahanolobis distance D_i of the example from class i is given by (17).

$$D_i = (c - \bar{c}_i) C^{-1} (c - \bar{c}_i) \tag{17}$$

where c is the vector of extracted appearance parameters, $\overline{c}_i$ is the centroid of the multivariate distribution for class i, and C is the common within class covariance matrix for all the training examples.

3.6 Improved AAM

While starting the search for the new image, the AAM search initializes the model parameters to the mean-shape and pose parameters. Therefore if the image is initially misplaced and arbitrarily scaled with respect to the initial model generated image then the process may not converge at all. Therefore a good initialization of pose parameters can improve the accuracy and speed of AAM search.

For doing this we get a rough estimate of the Face boundary in the image. This is done using histogram segmentation and clustering. This gives us a good estimate of the initial translation and scale factor.

A good initializing technique like this has a tremendous effect on the accuracy and efficiency of the system. We found out an increase of 4% in accuracy by using this technique.

4 Kernel-Based Face Tracking

Kernel-Based Object Tracking has recently been introduced by Comaniciu et al. in [2]. An isotropic kernel is used to mask the target spatially, so that a spatially-smooth similarity function can be defined. The similarity between target model and target candidates in subsequent frame is measured using the metric derived from the Bhattacharya coefficient [7], which has the meaning of a correlation score.

4.1 Target Representation

The reference target model is represented by its pdf, q in the feature space, while the target candidate at location y is characterized by the pdf p(y). The pdf estimates are considered as the m-bin histograms. The Bhattacharya coefficient between $\hat{p}$ (estimate of p) and $\hat{q}$ (estimate of q) is used as the similarity metric as given in (18).

$$\hat{r}(\mathrm{y}) = r\left[\hat{p}_{\mathrm{i}}(\mathrm{y}), \hat{q}_{\mathrm{i}}\right] = \sum_{\mathrm{i}=1}^{\mathrm{m}} \sqrt{\hat{p}_{\mathrm{i}}(\mathrm{y})\hat{q}_{\mathrm{i}}} \qquad (18)$$

The similarity function given in (18) defines a distance among target model and candidates.

4.2 Target Localization

The location corresponding to the target in the current frame is obtained by minimizing the distance measure given in (19).

$$d(\mathrm{y}) = \sqrt{1 - r\left[\hat{p}_{\mathrm{i}}(\mathrm{y}), \hat{q}_{\mathrm{i}}\right]} \qquad (19)$$

The localization procedure starts at the location of the previous frame and searches in the neighbourhood. Use of the isotropic kernel makes the distance function a smooth function of position y, thus allowing usage of gradient information provided by mean shift vector as in [3].

Distance minimization is carried out by maximizing the Bhattacharya coefficient with the kernel recursively moving from the current location y_0 to the new location y_1.

5 Face Authentication and Tracking System

The Face Authentication and Tracking System includes the following :

- Face Detection System
- Face Recognition System
- Face Tracking System

Once faces are detected, the captured images are passed on to the face recognition system which locates the relevant person, thus authenticating the same. The located relevant face is then tracked using the kernel-based tracking.

5.1 Information-theoretic approach to speed up the frame analysis process in videos

In videos, it is often observed that two consequent frames represent similar information to a very high degree. Thus analyzing very similar frames leads to unnecessary processing time. Therefore, while no relevant face has been recognized in the video, one can skip frames which are very similar. The information content of a frame is represented using the m-bin histogram of the image. We use an approach similar to [8] to analyze two subsequent frames. The mutual information of two subsequent frames gives us a measure of their similarity. The similar frames are skipped thus making the process very fast.

6 Results

The IIT-Kanpur Face Analysis System is developed in Java and runs on Sun Microsystems's j2dsk1.4. There is no pipelined scheme at present and so the system first detects and then recognizes. However, use of the tracking method saves us from re-detecting faces in every frame. Also, a considerable section of image is rejected due to the skin-filter application.

In Figure 5, the frame is first passed through the skin-filter, this removing more than 50% of the image from the search space. A 20x20 window is then scanned over the image. The detected face is passed to the AAM Search module for recognition.

Note that the identified relevant face varies in size in both the frames, this confirming the robustness of the method. The face is then tracked in subsequent face. A frame with another interfering face is shown in Figure 6.

Figure 5. The Face Analysis System running on a video frame. The relevant face here has been recognized efficiently.

In the first frame, the relevant face is recognized and in the next frame, it is located again by tracking. The Face Detection and Recognition is carried out in Grayscale, while the Kernel-based face tracking is done in RGB format. However, we display the Grayscale image in all of the cases for homogeneity.

Figure 6. The Relevant face being tracked in subsequent frame. Note that, introduction of a new face has no significance on the efficiency of the system.

7 Conclusions

This work integrates three well known methods for Face Analysis. In addition, we have found out ways to minimize the computation time and increase the efficiency. The use of information-theoretic analysis for skipping frames forms a major direction to be used in future. At present we do not use agent based pipelined methods in the system. Using this as in [12], would lead to further increase in computational speed of the system.

References

[1] M. S. Bartlett, J. R. Movellan, and T. J. Sejnowski, "Face Recognition using Independent Component Analysis," *IEEE Trans. on Neural Networks,* Vol 13, No. 6, pp. 1450-1464, Nov. 2002.

[2] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-Based Object Tracking," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* Vol 25, No. 5, pp. 564-577, May 2003.

[3] D. Comaniciu and P. Meer, "Mean Shift: A Robust Approach Towards Feature Space Analysis," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* Vol 24, No. 5, pp. 603-619, May 2002.

[4] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active Appearance Models," *in Proceedings of European Conf. on Computer-Vision,* Vol 2, pp. 484-498, Springer, 1998.

[5] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods,* Cambridge University Press, 2000.

[6] G. J. Edwards, C. J. Taylor, and T. F. Cootes, "Interpreting Face Images using Active Appearance Models," *in Proceedings of Third IEEE Intl. Conf. on Automatic Face and Gesture Recognition,* pp. 300-305, 1998.

[7] T. Kailath, "The divergence and Bhattacharya distance measures in signal selection," *IEEE Trans. on Communication Technology,* Vol 15, pp. 52-60, 1967.

[8] M. S. Lew and N. Huijsmans, "Information Theory and Face Detection," *Proceedings of 13th Intl. Conf. on Pattern Recognition,* Vol 3, pp. 601-605, Aug, 1996.

[9] S. Z. Li., Q. Fu, L. Gu, B. Scholkopf, Y. Cheng, H. Zhang, "Kernel machine based learning for multi-view face detection and pose estimation," *in Proceedings of Eighth IEEE Intl. Conf. on Computer Vision,* Vol 2, pp. 674-679, July 2001.

[10] Y. Li, S. Gong, J. Sherrah, and H. Liddell, "Multi-view face detection using support vector machines and eigenspace modeling," *in Proceedings of Fourth Intl. Conf. on Knowledge-based Intelligent Engineering Systems and Allied Technologies,* Vol 1, pp. 241-244, 2000.

[11] H. Rowley, S. Baluja, and T. Kanade, "Neural-network based face detection," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* Vol 20. No. 1, pp. 23-38, Jan. 1998.

[12] R. Sukthankar and R. Stockton, "Argus: The Digital Doorman", *IEEE Intelligent Systems,* Vol 16, No. 2, pp. 14-19, March-April, 2001.

[13] V. N. Vapnik, *Statistical Learning Theory,* John Wiley and Sons, Inc., New York, 1998.

[14] CMU Face Database available at http://vasc.ri.cmu.edu/idb/html/face/index.html