

Analysis and Performance of Face Recognition System Using Log Gabor Filter Bank with HMM Model

Ankita Nigam and Rajeev Shrivastava

Abstract—In this paper I present a biometrics system performing identification, of automatic face recognition. This system is based on Gabor features extraction using Log Gabor filter bank construction. For feature extraction the input image is convolve with log Gabor filter bank to select a set of informative and non-redundant Gabor features. The extracted features are again subjected to Discrete Radom Transform (DRT) to extract a sequence of feature vectors. The HMM (Hidden Markov Models) is used for matching the input face image to the stored images. The purpose of this research is to develop a novel, accurate and efficient face verification system.

Index Terms—HMM, gabor, face.

I. INTRODUCTION

Automatic Face Recognition is a challenging problem in computer vision. One of its main goals is the understanding of the complex human visual system and the knowledge of how humans represent faces in order to discriminate different identities with high accuracy. Two basic and conceptually independent problems have to be addressed by this kind of systems: face detection and recognition of the detected face. Work on the recognition stage, takes the detected face values as input to the algorithm. This stage can be separated in two steps: feature extraction, where important information for discrimination is saved, and the matching step, where the recognition result is given with the aid of a face database. image. The extracted features are again subjected to Discrete Radom Transform (DRT) to extract a sequence of feature vectors from a image. The HMM-based system developed in this paper matches the feature set (observation sequence) for a test image with an HMM of the claimed image, through Viterbi alignment. A distance measure is obtained by calculating negative log likelihood.

In this paper an investigation into the use of Gabor filter bank representations for the task of recognition of both 2D and 3D images is presented. In Section 2 a brief introduction to Log Gabor filter bank construction, Log Gabor filters is given followed by an overview of the recognition process in 3. In section 4 describes the component wise HMM method towards the detection of human Face Section 5 reports on the performance of our system

II. FILTER BANK CONSTRUCTIONS

Manuscript received July 10, 2012; revised August 16, 2012.

F. A. Ankita Nigam is with the Sriram Institute of S Technology, Jabalpur, 482002 India (e-mail: rajeev441220@yahoo.com).

S. B. Rajeev Shrivastava was with TIETECH Jabalpur. He is now with the Department of Electronics & communication, GRKIST Jabalpur, M.P. 482003 India (e-mail:rajeev441220@rediffmail.com).

The Log-Gabor filter has a response that is Gaussian when viewed on a logarithmic frequency scale instead of a linear one. This allows more information to be captured in the high frequency areas and also has desirable high pass characteristics.

Field [1] defines the frequency response of a Log-Gabor filter as

$$G(f) = \exp - \frac{\log\left(\frac{f}{k}\right)}{2 \log\left(\frac{\sigma}{k}\right)} \quad (1)$$

where $k = [u_0, v_0, w_0, \dots \dots]^T$ is once again the centre frequency of the sinusoid and σ is a scaling factor of the bandwidth.

A. Filter Bank Construction

In order to cover the frequency spectrum effectively, a range of both scales and orientations of the Gabor filters must be considered. The overall aim is to provide an even coverage of the frequency components of interest while maintaining a minimum of overlap between filters so as to achieve a measure of independence between the extracted co-efficient.

In the experiment action which follows, the Log-Gabor filter bank was constructed with a total of 6 orientations and 3 scales. The shape parameter σ_y has chosen such that each filter had a bandwidth of approximately 2 octaves. The half magnitude profiles of the filters can be seen in Figure 1. Using such an array of filters it is possible to extract a set of co-efficient from an image which contains the significant portion of the information but transformed into less correlated features. The resulting series of co-efficient is also significantly larger than that of the original image (there are 6 orientations x 3 scales for each pixel in the original image). Methods for reducing the dimensionality of this feature vector and other aspects of the recognition process are described in the following section

III. FACE VERIFICATION

There have been many proposed approaches for automatically recognizing humans from digital images, however, it is fair to say that none have enjoyed the popularity of the Eigen-faces approach by Turk and Pentland [2]-[4].

Today it is a standard baseline against which prospective methods are contrasted and has been utilized in a myriad of systems.

A. Eigen Faces

The majority of proposed methods apply a Principal Component Analysis (PCA) directly to the intensity values extracted from the face region [5]. This generates a set of Eigen-faces which can be visually inspected to verify semblances of the human visage; however this approach can

be equally well applied to a transformed version of the face such as a Log-Gabor filter bank representation.

In the case of Log-Gabor representations this serves two main purposes

- To reduce the dimensionality of the data generated by the filtering process to a more compact form which can be used in standard statistical approaches,
- To further de-correlate the data. This is necessary because the filters still maintain some overlap in the frequency domain causing some interdependence between the co-efficient, as can be seen from Figure 2.

In [6] a standardized implementation of the Eigen-face approach was developed by researchers at Colorado State University (CSU). This package has been used to perform the following experiments and to provide a benchmark allowing them to be easily reproducible. The images from the database are first cropped and normalized using the included eye location metadata.

By using the Eigen faces approach it is possible to calculate a transformation T_{MN} , which maps the original data into a M-dimensional subspace where $M \ll N$. Instances from the training set are transformed into this subspace and Distance metrics calculated between them in the lower dimensional space. Turk originally used a Euclidean Based classifier; however, experimentation with a wide variety of distance measures has shown the Mahalanobis Cosine distance measure to provide superior performance. This metric is defined as [7]

$$D_{MahCosine} = -\frac{|m||n|\cos\theta_{mn}}{|m||n|} \quad (2)$$

$$= -\frac{m \cdot n}{|m||n|},$$

where m and n are two images, I_1 and I_2 transformed into Mahalanobis space by the transformation T_{MN} and λ . This is subtly different from the transform into the Eigen-space as it also involves a scaling of each axis m_i by a factor of $\frac{1}{\sqrt{\lambda_i}}$ such that the variance in each dimension is unity.

IV. IMAGES MODELING USING HMM

HMM-based system developed uses a continuous first order HMM to represent each image

The HMM-based and DTW-based systems use similar verification protocols A pattern recognition system, which is based on HMMs, typically uses an HMM to represent each pattern class. Each of these HMMs is used to model an observation sequence, as well as the relationship between the individual observations. HMMs are therefore constructed in such a way that time-evolution is assumed from one observation in the sequence to the next. Since speech signals and dynamic (on-line) images also contain temporal information, it is possible to extract a continuous observation sequence from these signals in a very intuitive way. For this reason HMMs are especially well-suited for modeling these type of signals. This is not the case for static (off-line) images. Consequently, feature vectors have to be extracted from off-line images in such a way that time-evolution is simulated from one observation to the next.

In this paper use a grid to segment an image into local

square cells. From each cell, the pixel density is computed, so that each pixel density represents a local feature. Each image is therefore represented by a sequence of feature vectors, where each feature vector represents the pixel densities associated with a column of cells. The HMM-based system developed in this dissertation simulates time-evolution from one observation to the next by calculating the DRT of each image during the feature extraction process. Before we discuss the HMM-based image model, we first present the notation in the following section.

Notation

We use the following notation for a sequence of T continuous observations,

$$XT_1 = \{x_1, x_2, \dots, x_T\}, \quad (1)$$

where x_i , $i = 1, 2, \dots, T$ denotes the i^{th} feature vector in the sequence.

We use the following notation for a continuous, first order HMM λ :

- We denote the N individual states as

$$S = \{s_1, s_2, \dots, s_N\}, \quad (2)$$

and the state at time t as q_t .

- The initial state distribution is denoted by

$$\pi = \{\pi_i\}, \text{ where } \pi_i = P(q_1 = s_i), i = 1, \dots, N. \quad (3)$$

- The state transition probability distribution is denoted by $A = \{a_{i,j}\}$, where

$$a_{i,j} = P(q_{t+1} = s_j | q_t = s_i), i = 1, \dots, N, j = 1, \dots, N. \quad (4)$$

- The PDF, which quantifies the similarity between a feature vector x and the state s_j , is denoted by

$$f(x|s_j, \lambda), j = 1, \dots, N. \quad (5)$$

- The similarity between an observation sequences X and a model λ is denoted by $f(X|\lambda)$. (6)

A. Image Modeling

The HMM-based system developed in simulates time-evolution from one observation in an observation sequence to the next by calculating the DRT of a raw image).The feature vectors are therefore obtained by calculating projections of a image at different angles, after which they are subjected to some further processing. The angle is therefore the dynamic variable. This enables us to construct an HMM for each image.

1) HMM Topology

The HMM-based system developed in represents each image with an HMM of which the states are organized in a ring (Figure 2).This model is equivalent to the popular left-to-right model, but a transition from the last state to the first state is allowed. Since the HMM is constructed in such a way that it is equally likely to enter the model at any state, and the feature vectors are obtained from all the projections, that is, the projections calculated at angles ranging from 0° to 360° , the ring topology of the HMM guarantees that the images are rotation invariant.

Each state in the HMM represents one or more feature vectors that occupy similar positions in a d-dimensional feature space. This implies that the HMM groups certain

projections (columns of the DRT) together. It is important to note that this segmentation process only takes place after some further image processing has been conducted on the original projections. The state transition probabilities and the parameters for the PDFs, that represent the individual states, are estimated during training.

When a test sequence is matched with a trained ring-structured HMM, it is possible that the HMM is exited before the entire ring has been traversed. In other words, it is possible that the HMM is exited before the state immediately preceding the emitting state, which was initially visited, is encountered for the definitions of emitting and non-emitting states. It is also possible that the ring is traversed more than once.

In order to ensure that the entire ring is traversed, and only once, one may for example use N HMMs, where each HMM has N states and left-to-right topology. Figure 4 illustrates this configuration for N = 6. Note that the first HMM, that is λ_1 , does not have a transition between S_6 (the last emitting state) and S_1 (the first emitting state).

The second HMM, that is λ_2 , does not have a transition between S_1 (the first emitting state) and S_2 (the second emitting state). Corresponding states within these HMMs, for example S_2 in λ_1 and S_2 in λ_2 , share a common probability density function. The initial and terminal (non-emitting) states are denoted by S_0 and S_7 , respectively. The probability to make a transition from the initial (non-emitting) state to S_1 in λ_1 , or to S_2 in λ_2 , or to S_3 in λ_3 , etc., is the same, that is $\pi_1 = \pi_2 = \pi_3 = \dots = \pi_6 = 1/6$.

When a transition is made from the initial (non-emitting) state to S_1 in λ_1 , it is only possible to reach the terminal (non-emitting) state from S_6 . Similarly, when a transition is made from the initial (non-emitting) state to S_2 in λ_2 , it is only possible to reach the terminal (non-emitting) state from S_1 , etc.

In this way it is still equally likely to enter the model at any state, but it is guaranteed that the entire ring will be traversed. However, when the above model is used with N HMMs and N states per HMM, we have N_2 states and $3N_2$ transitions, in contrast to the original N states and $3N$ transitions. This enlarges the computational requirements considerably. We therefore did not implement this model.

The HMM-based system developed in this dissertation utilizes a single ring-structured HMM, like the one shown in figure3.

2) Initial Estimates

The HMM-based system developed in this dissertation uses uniform estimates for the initial state and state transition probabilities. The PDFs, which represent the individual states, are estimated by first assigning an equal number of observations to each state. The average of the observations within each state is then calculated.

3) Training

Each model is trained using the Viterbi reestimation technique. The dissimilarity between an observation sequence X and a model λ can therefore be calculated as follows [8]

$$D(X, \lambda) = -\ln(f(X|\lambda)). \quad (7)$$

In real-world scenarios, each writer can only submit a small number of training samples when he or she is enrolled into the system. Since the algorithm uses feature vectors with a high dimension, the estimated covariance matrix of the PDF for each state is not reliable and may even be singular.

A Mahalanobis distance measure can therefore not be found. Consequently these covariance matrices are not estimated and are initially set to $0.5I$, where I is the identity matrix. Only the mean vectors are estimated, which implies that the dissimilarity values are based on an Euclidean distance measure. We assume that training images are available for only a limited number of image, that is for those image in the database used. Assuming that there are Ω image in the database, the training images for each writer are used to construct an HMM, resulting in Ω models, that is $\{\lambda_1, \lambda_2, \dots, \lambda_\Omega\}$.

When the training set for writer ω is denoted by

$$\{X_1^{(\omega)}, X_2^{(\omega)}, \dots, \dots, X_N^{(\omega)}\} \quad (8)$$

where N_ω is the number of samples in the training set, the dissimilarity between every training sample and the model is used to determine the following statistics for the image,

$$\mu_\omega = \frac{1}{N_\omega} \sum_{i=1}^{N_\omega} D(X_i^{(\omega)}, \lambda_\omega) \quad (9)$$

$$\sigma_\omega^2 = \frac{1}{N_\omega - 1} \sum_{i=1}^{N_\omega} (D(X_i^{(\omega)}, \lambda_\omega) - \mu_\omega)^2 \quad (10)$$

These statistics are used to obtain a threshold distance, which is subsequently used to authenticate an input (test) image. The mean provides a reference distance. The standard deviation σ_ω measures the variability of the image.

Note that the mean μ_ω represents the average dissimilarity between the observation sequences in the training set and the HMM, λ_ω . Also note that λ_ω was trained with these observation sequences. This implies that the mean μ_ω , in conjunction with providing a reference distance, also measures the variability of the image. The trained model for the image ω therefore consists of the trained HMM, λ_ω , in conjunction with one or both of the statistics defined in above equations.

B. Verification

When a system aims to detect only subsets of other image' training sets can be used to model system. This is called "impostor validation" and can be achieved through strategies like test normalization. These techniques enable one to construct verifiers that detect random false image very accurately. Our verifier is constructed as follows. When a claim is made that the test pattern $X(w)$ Test belongs to image w, the pattern is first matched with the model λ_w through Viterbi alignment. This match is quantified by $f(X(w)|\lambda_w)$. The dissimilarity between the test pattern and the model is then calculated as follows:

$$d(X_{Test}^{(\omega)}, \lambda_\omega) = -\ln(f(X_{Test}^{(\omega)}|\lambda_\omega)) \quad (11)$$

In order to use a global threshold for all image, Dolfing [9] suggests that every dissimilarity value in (11) is normalised, using the statistics of the claimed image, that is,

$$d_{Mah}(X_{T_{est}}^{(\omega)}, \lambda_{\omega}) = \frac{d(X_{T_{est}}^{(\omega)}, \lambda_{\omega}) - \mu_{\omega}}{\sigma_{\omega}} \quad (12)$$

where $d_{Mah}(X_{T_{est}}^{(\omega)}, \lambda_{\omega})$ denotes the normalized dissimilarity between the test pattern and the model of the claimed image. This normalization is based on the assumption that the dissimilarity value in [9] is based on a Mahalanobis distance measure.

When only the mean vectors are estimated though, the dissimilarity value in (11) is based on an Euclidean distance measure. When this is the case, we found that significantly better results are obtained when the standard deviation of the dissimilarities of the training set, that is, σ_w in (12), is replaced by the mean μ_w , that is,

$$d_{Eud}(X_{T_{est}}^{(\omega)}, \lambda_{\omega}) = \frac{d(X_{T_{est}}^{(\omega)}, \lambda_{\omega}) - \mu_{\omega}}{\mu_{\omega}} \quad (13)$$

A sliding threshold τ , where $\tau \in (-\infty, \infty)$, is used to determine the error rates for the test patterns. When

$$d_{Eud}(X_{T_{est}}^{(\omega)}, \lambda_{\omega}) < \tau \text{ that is}$$

$$d_{Eud}(X_{T_{est}}^{(\omega)}, \lambda_{\omega}) < \mu_{\omega}(1 + \tau) \quad (14)$$

The claim is accepted, otherwise, the claim is rejected. When $\tau = 0$, all the test patterns for which $d(X_{T_{est}}^{(\omega)}, \lambda_{\omega}) \geq \mu_w$ are rejected. When $\tau \rightarrow \infty$ all the test patterns, for which $d(X_{T_{est}}^{(\omega)}, \lambda_{\omega})$ is finite are accepted.

1) Finding the best state sequence using the Viterbi algorithm.

To find the single best state sequence $QT^* = \{q^*1, q^*2, \dots, q^*T\}$ for the given observation sequence $OT_1 = \{o_1, o_2, \dots, o_T\}$, we need to define the quantity

$$\delta t(i) = \max_{All Q_{1}^{t-1}} p(q_t = s_i, O_1^t | \lambda),$$

that is the best score along a single path, which accounts for the first t observations, and ends in state s_i . By induction we have that

$$\delta t(i) = \max_{i=1, \dots, N} [\delta_{t-1}(i) a_{ij}] b_j(O_{t+1}),$$

To retrieve the state sequence, we need to keep track of the argument which maximized (B.30) for each t and j . We do this via the array $\psi_t(j)$. The complete procedure for finding the best state sequence (Viterbi algorithm) can now be stated as follows:

Initialization

$$\delta_1(i) = \pi_i b_i(o_1), i = 1, 2, \dots, N$$

$$\psi_1(i) = 0$$

Recursion

$$\delta_t(j) = \max_{j=1 \dots N} [\delta_{t-1}(i) a_{ij}] b_j(o_t), t = 2, \dots, T, j = 1, \dots, N$$

$$\psi_t(j) = \operatorname{argmax}_{j=1 \dots N} [\delta_{t-1}(i) a_{ij}], t = 2, \dots, T, j = 1, \dots, N$$

Termination

$$P^* = \max_{i=1, \dots, N} [\delta_T(i)]$$

$$q_t^* = \operatorname{argmax}_{i=1, \dots, N} [\delta_T(i)]$$

Backtracking

$$q_t^* = \psi_{t+1}(q_{t+1}^*), t = T - 1, T - 2, \dots, 1$$

Here $P^* = P(OT_1, QT^* | \lambda)$ is an approximation to the forward or the backward method

V. RESULTS

Each state in the HMM used in this paper is represented by a PDF for which only the mean vector is estimated. The corresponding covariance matrix is kept fixed. The dissimilarity between an observation sequence and the HMM is therefore based on an Euclidean distance measure. Due to the high dimension of the feature vectors used in this paper. We have used number of observations i.e. the feature vector=256, number of states=64, feature length=512 when $d = 512$, the number of feature vectors ($T = 2N\Theta = 256$) is therefore four times the number of HMM states ($N = 64$).

For example, a threshold of $\tau = 0.16$ is selected, equation (4.14) of chapter 2 implies that all the test patterns for which $D(X_{T_{est}}^{(\omega)}, \lambda_{\omega}) \geq 1.16\mu_{\omega}$ are rejected the other pattern is accepted.

For various values of observations, states and feature lengths, it is clear that when the dimension of the feature vectors is decreased from $d = 512$ to $d = 256$ or even to $d = 128$, the performance of the system is not significantly decreases. The performance of the HMM-based system is generally enhanced when the number of feature vectors, or the number of states in the HMM, that is N , is increased. The best results are obtained when only one forward link is allowed in the HMM.

ACKNOWLEDGMENT

We are thankful to organization TITECH Jabalpur for their valuable support during my work. We would also like to thank Dr. M. M. Mushrif, Prof. Shweta Modi and Prof. Brajesh Patel for there valuable suggestions.

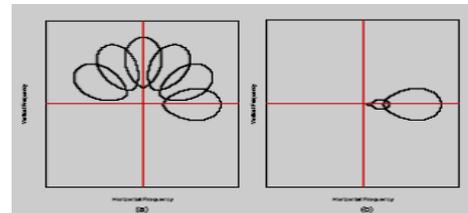


Fig. 1. The halfmagnitude profiles for varying orientations and (b) scales

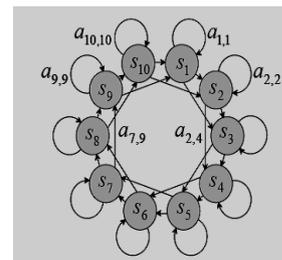


Fig. 2. An example of an HMM with a ring topology. This model has ten states with one state skip. One state skip is equivalent to two allotted forward links.

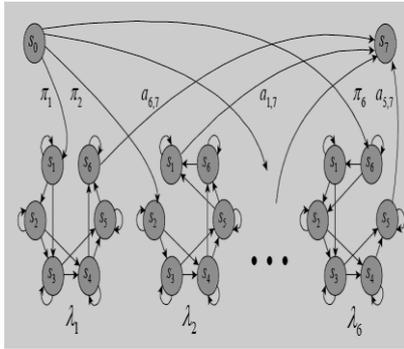


Fig. 3. An example of a model that consists of six individual HMMs. Each of these individual HMMs has six states with one state skip and a left-to-right topology. The initial (non-emitting) state is denoted by s_0 and the terminal (non-emitting) state by s_7 . With this configuration it is still equally likely to enter the model at any state, but it is guaranteed that the entire ring will be traversed.

REFERENCES

[1] D. Fields, "Relations between the statistics of natural images and the response properties of cortical cells," *Journal of Optical Society of America*, vol. 4, no. 12, pp. 2379–2394, 1987.

[2] R. Chellapa, C. L. Wilson, S. Sirohey, and C. S. Barnes, "Human and Machine Recognition of Faces: A Survey," Tech. Rep., University of Maryland and NIST, August 1994.

[3] Y. Saito, Y. Kenmochi, and K. Kotani, "Estimation of eyeglassless facial images using principal component analysis," *Proceedings of the International Conference on Image Processing*, vol. 4, pp. 197–201, October 1999.

[4] B. Moghaddam and A. Pentland, "Beyond euclidean recognition," in *Proceeding. of International Conference on Automatic Face and Gesture Recognition (FG'98)*, Nara, Japan, April 1998, pp. 30–35.

[5] Colorado State University. (2003). The CSU Face Identification EvaluationSystem. [Online]. Available: <http://www.cs.colostate.edu/evalfacerec/index.html>

[6] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuro Science*, vol. 3, no.1, pp. 71–86, 1991.

[7] Peter Kovesi, *Invariant Measures of Image Features From Phase Information*, Ph.D. thesis, The University of Western Australia, 1996.

[8] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," in *Proceedings of the IEEE*, vol. 77, pp. 257- 286,1989

[9] J. G. A. Dolfing, "Handwriting recognition and verification. A hidden Markov approach," Ph.D. thesis, Eindhoven University of Technology, 1998.



Ankitta Nigam was born in Kanpur on 27th April 1981. She has done MCA in compute application from IGNOU Delhi in 2007. She is presently doing M.E in Software System from SRIT Jabalpur .



Rajeev Shrivastava was born in Jabalpur on 5th July 1980. He has done M.E in Digital Communication from SRIT Jabalpur in 2009. He is presently doing PhD in Digital Image Processing from JVVU Jaipur.