GAUSSIAN MIXTURE MODELS DESIGN AND APPLICATIONS

A THESIS
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING
AND THE INSTITUTE OF ENGINEERING AND SCIENCES
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCES

By

Khaled Ben Fatma

January 2000

GAUSSIAN MIXTURE MODELS DESIGN AND APPLICATIONS

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND

ELECTRONICS ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCES

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Khaled Ben Fatma

January 2000

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

A. Enis Çetin, Ph. D.(Supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Billur Barshan, Ph. D.

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Yasemin Yardımcı, Ph. D.

Approved for the Institute of Engineering and Sciences:

Prof. Dr. Mehmet Baray
Director of Institute of Engineering and Sciences

ABSTRACT

GAUSSIAN MIXTURE MODELS DESIGN AND APPLICATIONS

Khaled Ben Fatma
M.S. in Electrical and Electronics Engineering
Supervisor: A. Enis Çetin, Ph. D.
January 2000

Two new design algorithms for estimating the parameters of Gaussian Mixture Models (GMM) are developed. These algorithms are based on fitting a GMM on the histogram of the data. The first method uses Least Squares Error (LSE) estimation with Gauss-Newton optimization technique to provide more accurate GMM parameter estimates than the commonly used Expectation-Maximization (EM) algorithm based estimates. The second method employs the matching pursuit algorithm which is based on finding the Gaussian functions that best match the individual components of a GMM from an overcomplete set. This algorithm provides a fast method for obtaining GMM parameter estimates.

The proposed methods can be used to model the distribution of a large set of arbitrary random variables. Application of GMMs in human skin color density modeling and speaker recognition is considered. For speaker recognition, a new set of speech feature parameters is developed. The suggested set is more

appropriate for speaker recognition applications than the widely used Mel-scale based one.

Keywords: Gaussian Mixture Models, Parameter Estimation, Expectation-Maximization Algorithm, Gauss-Newton Algorithm, Matching Pursuit Algorithm, Least Squares Error, Speaker Recognition.

ÖZET

GAUSS KARIŞIM MODELLERİNİN TASARIMI VE UYGULAMALAR

Khaled Ben Fatma
Elektrik ve Elektronik Mühendisliği Bölümü Yüksek Lisans
Tez Yöneticisi: Prof. Dr. A. Enis Çetin
Ocak 2000

Gauss Karışım Modellerini (GMM) parametrelerinin kestirimi amacıyla iki yeni tasarım algoritması geliştirilmiştir. Bu algoritmalar veri histogramı uydurma yoluna dayanmaktadır. Birinci yöntem, GMM parametre kestiriminde alışılagelmiş beklenti en büyükleme (EM) algoritması tabanlı kestirimlerden daha doğru sonuçlar sağlamak için Gauss-Newton eniyileme tekniğiyle en küçük kareler hata kestirimini (LSE) kullanmaktadır. İkinci yöntem, sözlük olarak adlandırılan, aşırı tamamlanmış Gauss modelleri kümesinden bir GMM'in her bir bileşenini en iyi eşleyen Gauss işlerlerini bulmak için kullanılan uyum izleme algoritmasına dayanmaktadır. Bu algoritma GMM parametre kestirimi için hızlı bir yöntem sunmaktadır.

Önerilen yöntem geniş bir rasgele değişken kümesini modellemekte kullanılabilir. GMM'lerin kullanım alanı olarak insan deri renk yoğunluğu modellemesi ve konuşmacı tanıma problemleri seçilmiştir. Konuşmacı tanıma için yeni bir konuşma öznitelik parametre kümesi geliştirilmiştir. Öngörülen

bu yeni küme, yaygın olarak kullanılan Mel-skala tabanlı kümeye kıyasla konuşmacı tanımaya daha uygundur.

Anahtar Kelimeler: Gauss Karışım Modelleri, Parametre Kestirimi, Beklenti En Büyükleme Algoritması, Gauss-Newton Algoritması, Uyum İzleme Algoritması, En Küçük Kareler Hatası, Konuşmacı Tanıma.

ACKNOWLEDGMENTS

I would like to express my deep gratitude to my supervisor Prof. Dr. A. Enis Çetin for his supervision, guidance, suggestions and patience throughout the development of this thesis.

I am grateful to Dr. Yasemin Yardımcı for her valuable suggestions and help which contributed a lot to the progress of this thesis. Special thanks to Dr. Billur Barshan for reading and commenting on the thesis.

I thank all of my friends for their sincere friendship throughout all these years.

It is a pleasure to express my special thanks to my father, mother, and sisters for their love, support, and encouragement.

Contents

List of Figures

List of Tables

To my Family, Friends, and the Reader ...

Chapter 1

Introduction

In nature, observed phenomena tend to have a wide variety of non-uniform distributions that often are very hard to estimate or model. In signal processing, modeling the distribution of an arbitrary phenomenon is a primordial step in understanding and analyzing the behavior of that phenomenon.

Gaussian Mixture Models (GMM) have been recently used in many applications as an efficient method for modeling arbitrary densities [1]. A Gaussian mixture density is defined as a weighted sum of different Gaussian component densities. GMMs were shown to provide a smooth approximation to the underlying long-term sample distribution of observations obtained from experimental measurements [1]. This is mainly due to the fact that a linear combination of Gaussian basis is capable of representing a large class of sample distributions, in addition to the observation that most natural phenomena tend to have a Gaussian distribution.

There are several techniques available for estimating the parameters of a GMM [2], [3], [4]. By far the most popular and well-established method is maximum likelihood (ML) estimation. The aim of ML estimation is to find the model parameters that maximize the likelihood of the GMM, given the training data. This usually leads to a nonlinear global optimization problem. ML parameter estimates can be obtained in an iterative manner using a special case of the Expectation-Maximization (EM) algorithm [5]. The EM algorithm is an iterative algorithm, which starts with an arbitrary model and tries to obtain a better model at each iteration until convergence in some sense is reached. The EM algorithm usually leads to good estimates of the GMM parameters. However, it does not always provide accurate estimates of the GMM parameters. Moreover, its computational complexity makes it unsuitable for applications where speed is important such as real time and adaptation applications.

New methods for estimating the parameters of a GMM by curve fitting to the histogram of the observation data are introduced. Two methods are described; one is based on least squares error estimation using Gauss-Newton algorithm and the other is based on the matching pursuit algorithm.

The least squares error method tries to obtain the best parameters by minimizing an error function over the unknown parameters. A parameter separation technique is used to simplify the optimization procedure [12]. The resulting error function is a highly nonlinear function of the parameters, the Gauss-Newton algorithm is used to obtain an iterative estimate to the problem. This method provides more accurate estimates resulting in a better model. Moreover, it needs a very few number of iterations to converge.

The second method is based on the matching pursuit algorithm. Pursuit algorithms are generally used to decompose arbitrary signals [15]. Decomposition vectors are chosen depending upon the signal properties. Vectors are selected one by one from a dictionary, while optimizing the signal approximation at each step. In this thesis, a modified version of the matching pursuit is used as an alternative method for estimating the parameters of a GMM. This method has a lower accuracy than the EM based method, but its low computational complexity makes much faster and more suitable for applications where speed is crucial such as adaptation algorithms [27], [28], and real time applications.

Speaker recognition is an important application where the use of GMMs has proven to be very efficient [1], [10]. Speaker recognition can be divided into two sub-fields: Speaker Identification which tries to identify the person speaking an utterance from a known set of speakers, and Speaker Verification which tries to check whether a speaker is that who he claims to be or is an impostor. For both of these tasks, many models like Hidden Markov models (HMMs), Multiple Binary Classifier Model (MBCM), Neural Networks, etc.. [10], are proposed. GMM is recognized as one of the most accurate models for Automatic Speaker Recognition (ASR), using telephone speech [1]. The speech spectrum based parameters are very effective for speaker modeling. The most widely used speech feature parameter set is based on the Mel-scale cepstrum. The Mel-scale based features produce excellent results for speech recognition, as the Mel-scale division of the spectrum is compatible with the human auditory system. This spectrum division may not be the best possible division for speaker recognition applications. In this thesis, we propose a new set of features that is more appropriate to speaker recognition applications.

This thesis is organized as follows. In Chapter 2, we describe briefly the general form of a GMM and the EM algorithm used for estimating its parameters. In Chapter 3, we develop the idea of using least squares data modeling implemented by the Gauss-Newton algorithm to derive more accurate GMM parameter estimates. An application of this idea is also described. Chapter 4 presents a fast GMM parameter estimation method based on a modified version of the matching pursuit algorithm. Speaker recognition is considered in Chapter 5, where a new set of speech feature parameters is proposed. Finally, conclusions are given in Chapter 6.

Chapter 2

Gaussian Mixture Models (GMM)

2.1 Description

Given an arbitrary D-dimensional random vector $\vec{x}$, a Gaussian mixture density of M components is defined as a weighted sum of individual D-variate Gaussian densities $b_i(\vec{x})$, $i = 1, ..., M$, as follows

$$p(\vec{x}|\lambda) = \sum_{i=1}^{M} p_i b_i(\vec{x}) \tag{2.1}$$

where p_i, $i = 1, ..., M$, are the weights of the individual components and are constrained by

$$\sum_{i=1}^{M} p_i = 1 \tag{2.2}$$

The D-variate Gaussian function $b_i(\vec{x})$ is given by

$$b_i(\vec{x}) = \frac{1}{(2\pi)^{D/2} |\Sigma_i|^{1/2}} \exp\left\{ -\frac{1}{2} (\vec{x} - \vec{\mu}_i)' \Sigma_i^{-1} (\vec{x} - \vec{\mu}_i) \right\} \tag{2.3}$$

where $\vec{\mu}_i$ is the mean vector and Σ_i is the covariance matrix. Therefore a GMM can be represented by the collection of its parameters λ as

$$\lambda = \{p_i, \vec{\mu}_i, \Sigma_i\}, \qquad i = 1, ..., M \tag{2.4}$$

The GMM can have different forms depending on the choice of the covariance matrices. The covariance matrices can be full or diagonal. Because the component Gaussians are acting together to model the overall probability density function, full covariance matrices are not necessary even if the observations are statistically dependent. The linear combination of diagonal covariance Gaussians is capable of modeling the correlations between the observation vector elements. The use of full covariance matrices can significantly complicate the GMM estimation procedure, while the effect of using M full covariance Gaussians can be approximated by using a larger set of diagonal covariance Gaussians.

2.2 Applications of GMM

Gaussian mixture models have been used in many applications as an efficient method for modeling arbitrary densities. Since a GMM is capable of modeling a broad range of probability densities, it has found use in a very large area of applications. In [7] for example, the probability density function of human skin color was estimated using a GMM. The estimated probability density function has many applications in image and video databases. These applications range from hand tracking to human face detection. Similarly in [8], an object tracking algorithm is developed using GMMs. Gaussian mixture models were used to estimate the probability densities of objects foreground and scene background

colors. Tracking was performed by fitting dynamic bounding boxes to image regions of maximum probabilities. GMMs have been also used very effectively in speaker recognition for modeling speaker identity [1], [10]. Short-term speaker-dependent feature vectors are obtained from the speech signal, then Gaussian mixture modeling is used to estimate the density of these vectors. The use of Gaussian mixture models for modeling speaker identity is motivated by the interpretation that the Gaussian components represent some general speaker-dependent spectral shapes and the capability of Gaussian mixtures to model arbitrary densities.

2.3 GMM Parameter Estimation

Given an observation sequence $\bar{X} = \{\underline{x}_1, \underline{x}_2, ..., \underline{x}_T\}$ from a random vector $\vec{x}$, the goal is to estimate the parameters of the GMM λ, which in some sense best matches the distribution of the observation data. This GMM can then be considered as a valid estimate to the distribution of the random vector $\vec{x}$.

There are several techniques available for estimating the parameters of a GMM [2], [3], [4]. By far, the most common and popular method is maximum likelihood (ML) estimation [5]. This method tries to find the model parameters that maximize the GMM likelihood

$$p(\bar{X}|\lambda) = \prod_{t=1}^{T} p(\underline{x}_t|\lambda) \tag{2.5}$$

given the training vectors $\bar{X}$. This leads to a nonlinear function of the parameters λ and direct minimization is not possible. However, ML estimates can

be obtained in an iterative manner using the Expectation-Maximization (EM) algorithm [5]. The EM algorithm is described in the next subsection.

Another possible method for the estimation of the GMM parameters λ, is to try to make a smooth fit to the histogram of the observation sequence $\bar{X}$ using a linear combination of Gaussian functions. This idea will be further investigated in later chapters. Two new methods using this idea will be presented along with some possible applications.

Usually there are two important factors in training of a GMM: model order selection and parameter initialization for iterative methods. We will not address these problems in this thesis.

2.4 Expectation Maximization (EM) Algorithm

The EM algorithm tries to find the estimates of the ML parameters iteratively. It begins with an initial model λ, and tries to estimate a better model until some convergence is reached. In each EM iteration, first a posteriori probability is estimated as

$$p(i|\underline{x}_t, \lambda) = \frac{p_i b_i(\underline{x}_t)}{\sum_{k=1}^{M} p_k b_k(\underline{x}_t)} \tag{2.6}$$

Based on this probability, mixture weights, means and variances are estimated using the following re-estimation formulas, which guarantee a monotonic increase in the model's likelihood value:

- *Mixture weights:*

$$\hat{p}_i = \frac{1}{T}\sum_{t=1}^{T} p\left(i|\underline{x}_t, \lambda\right) \qquad i = 1, ..., M \tag{2.7}$$

where $\hat{p}_i$ is the new estimate of the ith mixture weight and it is obtained by averaging all a posteriori probability estimates.

- *Means*:

$$\overrightarrow{\hat{\mu}} = \frac{\sum_{t=1}^{T} p(i|\underline{x}_t, \lambda)\underline{x}_t}{\sum_{t=1}^{T} p(i|\underline{x}_t, \lambda)} \tag{2.8}$$

where $\overrightarrow{\hat{\mu}}$ is the new estimated mean vector of the ith mixture.

- Variances:

$$\hat{\sigma}_i^2 = \frac{\sum_{t=1}^{T} p(i|\underline{x}_t, \lambda)x_t^2}{\sum_{t=1}^{T} p(i|\underline{x}_t, \lambda)} - \hat{\mu}_i^2 \tag{2.9}$$

where $\hat{\sigma}_i^2$ refers to new estimates of arbitrary entries on the diagonal of the covariance matrix and x_t, $\hat{\mu}_i$ refer to the corresponding elements of the vectors $\underline{x}_t$ and $\overrightarrow{\hat{\mu}}$.

In many applications, the EM algorithm has shown satisfying results . However, it does not always provide accurate estimates and it may converge to bad local maxima. Better estimates can usually be obtained for the same model under consideration. Moreover, the computational complexity of the EM algorithm is relatively high especially when the training set is large [6].

Chapter 3

Least Squares Error (LSE) Estimation

In the previous chapter, we described how ML estimation can be used to obtain estimates to the parameters of a GMM through the EM algorithm. In this chapter, we discuss the estimation of the parameters of a GMM by trying a least squares fit to the histogram of the observation data. Given an observation sequence $\bar{X} = \{\underline{x}_1, \underline{x}_2, ..., \underline{x}_T, \}$, we first obtain the normalized histogram $H(\vec{x})$. We want to use $H(\vec{x})$ to obtain a Gaussian mixture estimate to the unknown distribution of $\vec{x}$ of the form

$$p(\vec{x}|\lambda) = \sum_{i=1}^{M} p_i b_i(\vec{x}) \tag{3.1}$$

where λ represents the model parameters and the weights p_i are constrained by $\sum_{i=1}^{M} p_i = 1$.

We start by expressing the histogram as

$$H(\vec{x}) = \sum_{i=1}^{M} p_i b_i(\vec{x}) + w(\vec{x}) \tag{3.2}$$

where $w(\vec{x})$ is the error between the histogram and the Gaussian mixture density to be estimated. In other words, we express the histogram of the observation data as a linear combination of Gaussian functions $b_i(\vec{x})$.

We use the least squares data modeling to estimate the parameters of the GMM. This is done by minimizing a given function of the estimation error $w(\vec{x})$. We use the Least Squares Error (LSE) criterion. The Gauss-Newton algorithm is later used to obtain estimates to the unknown parameters in an iterative manner.

We first present a brief review of the basic concepts of least squares data modeling, then we proceed to its application in GMM parameter estimation.

3.1 Least Squares Data Modeling

In many applications, the observed signal or sequence is often assumed to be composed of a linear combination of "basis functions" which are characterized by a set of parameters, and additive noise [11]. The observation vector of length N is given by

$$\underline{h} = \sum_{i=1}^{M} p_i \underline{b}_i(\underline{\theta}_i) + \underline{w} \tag{3.3}$$

where p_i is the coefficient of ith basis vector $\underline{b}_i(\underline{\theta}_i)$, which depends on the parameter vector $\underline{\theta}_i$, while $\underline{w}$ is the additive error sequence. This expression

can also be written in the compact form

$$\underline{h} = B(\underline{\theta})\underline{p} + \underline{w} \tag{3.4}$$

where $B(\underline{\theta})$ is a $N \times M$ basis matrix given by

$$B(\underline{\theta}) = [\underline{b}_1(\underline{\theta}_1) \quad \underline{b}_2(\underline{\theta}_2) \ \ldots \ \underline{b}_M(\underline{\theta}_M)], \tag{3.5}$$

$\underline{p}$ is the vector containing the M coefficients and $\underline{\theta}$ is the composite parameter vector

$$\underline{\theta} = \left[\underline{\theta}_1^T \quad \underline{\theta}_2^T \ \ldots \ \underline{\theta}_M^T\right]^T. \tag{3.6}$$

The objective is to select the unknown parameter vectors $\underline{\theta}$ and the amplitude set $\underline{p}$ so that the linear combination of the basis functions best fits $\underline{h}$. Using the LSE criterion we have to minimize the functional

$$e(\underline{\theta}, \underline{p}) = \left\|\underline{h} - B(\underline{\theta})\underline{p}\right\|^2 \tag{3.7}$$

This is a highly nonlinear optimization problem with no closed from solution, therefore nonlinear programming techniques are necessary to achieve the optimization.

To make the optimization problem in (3.7) simpler, we note that the function to be minimized has two important properties:

- The unknown parameters $\underline{\theta}$ and $\underline{p}$ are separable.
- The least squares error criterion $e(\underline{\theta}, \underline{p})$ is a quadratic function of the amplitudes $\underline{p}$.

For problems with these properties, Gloub and Pereyra [12], proposed a parameter separation technique to ease the complexity of the problem. The

idea is to find the optimum amplitude vector $\underline{p}^o$ in terms of the unknown parameters $\underline{\theta}$. Then the set of unknowns reduces to the vector $\underline{\theta}$. Once these are found, the optimum amplitude vector $\underline{p}^o$ can then be obtained directly. This parameter separation technique simplifies the computations and significantly improves the speed of convergence.

To obtain an expression of the optimum amplitude vector in term of the unknown parameters $\underline{\theta}$, we first use the QR decomposition to write basis matrix $B(\underline{\theta})$ in the form

$$B(\underline{\theta}) = Q(\underline{\theta})R(\underline{\theta}) \tag{3.8}$$

where $Q(\underline{\theta})$ is a $N \times M$ orthonormal matrix and $R(\underline{\theta})$ is a $M \times M$ nonsingular upper triangular matrix. The expression for the optimum amplitude vector is formulated in [11] and given by

$$\underline{p}^o = R(\underline{\theta})^{-1}Q(\underline{\theta})^T\underline{h}. \tag{3.9}$$

The corresponding least squares error criterion's value for this optimum choice is given by

$$e(\underline{\theta}, \underline{p}^o) = \underline{h}^T\underline{h} - \underline{h}^TQ(\underline{\theta})Q(\underline{\theta})^T\underline{h}. \tag{3.10}$$

By minimizing criterion (3.10), we obtain the vector of the unknown parameters $\underline{\theta}$. Once this vector has been found, it is substituted into expression (3.9) to obtain the corresponding amplitude vector $\underline{p}^o$.

3.2 GMM Parameter Estimation

In our application, we want to estimate a density function that fits the distribution of a sequence of observed data, as a mixture of Gaussian functions. To

use least squares data modeling, we have to put expression (3.2) in the form of expression (3.4). For 1-D case, this is straightforward:

$$\underline{h} = \sum_{i=1}^{M} p_i b_i(\underline{\theta}_i) + \underline{w} \tag{3.11}$$

$$= B(\underline{\theta})\underline{p} + \underline{w} \tag{3.12}$$

where:

- $\underline{p} = [p_1 \quad p_2 \ \ldots \ p_M]^T$ is the mixture weights vector,
- $\underline{\theta}_i = [\mu_i \quad \sigma_i^2]^T$ is the parameter vector of the ith Gaussian component and
- $B(\underline{\theta}) = \begin{bmatrix} b_1(x_1, \underline{\theta}_1) & b_1(x_1, \underline{\theta}_2) & & b_1(x_1, \underline{\theta}_M) \\ b_1(x_2, \underline{\theta}_1) & b_1(x_2, \underline{\theta}_2) & & b_1(x_2, \underline{\theta}_M) \\ & & & \\ b_1(x_N, \underline{\theta}_1) & b_1(x_N, \underline{\theta}_2) & \cdots & b_1(x_N, \underline{\theta}_M) \end{bmatrix}$.

In our model, we use diagonal covariance matrices. Expression (3.9) gives us the optimum weights vector as

$$\underline{p}^o = R(\underline{\theta})^{-1} Q(\underline{\theta})^T \underline{h}. \tag{3.13}$$

where $Q(\underline{\theta})$ and $R(\underline{\theta})$ are obtained from QR decomposition of $B(\underline{\theta})$.

This approach can be extended to higher dimensions in a similar manner. The observation vector $\underline{h}$ can be obtained by putting the columns of $H(\vec{x})$ into one vector sequentially, $B(\underline{\theta})$ can be then obtained accordingly.

3.2.1 Gauss-Newton Algorithm

The minimization problem (3.10) is highly nonlinear in the unknown vector $\underline{\theta}$, therefore nonlinear programming techniques must be used to achieve the optimization. For this task, we use the Gauss-Newton algorithm developed in [11], which is a descent method that has proven to be very effective in solving highly nonlinear programming problems. In typical iterative optimization techniques, the parameter vector is incrementally perturbed so that the cost criterion takes lower values at each iteration. In other words, the current parameter vector $\underline{\theta}_k$ is perturbed to obtain

$$\underline{\theta}_{k+1} = \underline{\theta}_k + \underline{\delta}_k \tag{3.14}$$

where $\underline{\delta}_k$ is the perturbation vector which is chosen in such a way that a decrease in the cost criterion results.

In Gauss-Newton algorithm, the optimum perturbation vector $\underline{\delta}_k^o$, which results in the highest decrease in the cost criterion, is estimated at each iteration. This procedure ensures that quadratic or superlinear convergence rates are attained in a neighborhood of a relative minimum.

For the nonlinear optimization problem given in (3.4), the Gauss-Newton perturbation vector at the kth iteration is given by

$$\underline{\delta}_k^{(GN)} = -\left[J\left(\underline{\theta}_k\right)^T J\left(\underline{\theta}_k\right)\right]^{-1} J\left(\underline{\theta}_k\right)^T e(\underline{p}^o, \underline{\theta}_k) \tag{3.15}$$

where $J\left(\underline{\theta}_k\right)$ is the Jacobian matrix and the residual vector $e(\underline{p}^o, \underline{\theta}_k)$ is given by

$$e(\underline{p}^o, \underline{\theta}) = \left(I - Q(\underline{\theta})Q(\underline{\theta})^T\right)\underline{h} \tag{3.16}$$

The Jacobian matrix, $J(\underline{\theta})$ has the form

$$J(\underline{\theta}) = \left[\frac{\partial}{\partial \theta_1} e(\underline{\theta}) \quad \frac{\partial}{\partial \theta_2} e(\underline{\theta}) \quad \cdots \quad \frac{\partial}{\partial \theta_l} e(\underline{\theta}) \right]. \tag{3.17}$$

The partial derivative terms are approximated as

$$\frac{\partial}{\partial \theta_k} e(\underline{\theta}) = -\frac{\partial P_{\underline{\theta}}}{\partial \theta_k} \underline{h} \tag{3.18}$$

$$= -(I - P_{\underline{\theta}}) \frac{\partial B(\underline{\theta})}{\partial \theta_k} R^{-1}(\underline{\theta}) Q(\underline{\theta})^T \underline{h} \tag{3.19}$$

where $P_{\underline{\theta}}$ is a projection matrix defined as $P_{\underline{\theta}} = Q(\underline{\theta})Q(\underline{\theta})^T$.

The algorithm starts with an initial estimate $\underline{\theta}_0$ of the unknown parameters. At each iteration of the Gauss-Newton algorithm, the optimal perturbation vector is used to update the parameters vector $\underline{\theta}_0$ so that an improvement in the criterion (3.10) is obtained

$$\underline{\theta}_{k+1} = \underline{\theta}_k + \alpha_k \underline{\delta}_k. \tag{3.20}$$

The step size α_k is selected large at early iterations and reduced at later stages of the optimization procedure. Usually, α_k is chosen from the sequence

$$\alpha_k = 1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \ldots \tag{3.21}$$

until the first value of α_k which reduces the cost criterion is found. Once the parameter vector $\underline{\theta}$ is found, it is inserted in expression (3.13) and the amplitude set $\underline{p}^o$ is obtained.

3.2.2 Algorithmic Issues

Initialization

One critical factor in GMM parameter estimation is the initialization of the model parameters. The initialization procedure is very important for the performance of the Gauss-Newton algorithm. It was checked experimentally that a bad initialization can result in high estimation error and a poor model. One efficient initialization method consists of randomly choosing vectors from the training data as mean vectors followed by K-means clustering to initialize means, variances and mixture weights.

Optimum Step Size

In subsection 3.2.1, we discussed how the step size α_k used in expression (3.20) can be chosen from the sequence in Equation (3.21). This procedure can be effective in finding an appropriate step size that results in a decrease in the error criterion for a given perturbation vector. However, it does not find the best possible step size that results in the highest decrease in the cost criterion. Since the calculation of a perturbation vector δ_k is relatively costly in computation power, we want to get the most out of this perturbation vector once it is calculated by estimating the corresponding step size α_k^o that results in the highest decrease in the error criterion. Here, we introduce a procedure for estimating the optimum step size α_k^o for a given perturbation vector. We start by considering the error function to be minimized given in Equation (3.7), which is the squared error between the normalized histogram and the estimated distribution. Once a new perturbation vector is calculated the new value of this

error function depends on the step size α_k , denoted by $e(\alpha_k)$. We want to find the value of α_k^o for which $e(\alpha_k^o) < e(\alpha_k)$, for all $0 < \alpha_k^o \leq 1$. For these values of α_k, it was found experimentally that the error function can be approximated by a parabola as

$$\hat{e}(\alpha_k) = a.\alpha_k^2 + b.\alpha_k + c \qquad ; a \in \mathrm{IR}^+ \tag{3.22}$$

For this parabola the minimum value is given for

$$\alpha_k^o = -\frac{b}{2a} \tag{3.23}$$

To fit a parabola to $e(\alpha_k)$ we need its value at three different point of α_k between 0 and 1. We use the region $0 \leq \alpha_k \leq 1$, because we do not want to get too far from our current operating point ($\alpha_k = 0$), since far from this point the error function is unpredictable and our approximation becomes invalid. We already know $e(0)$ which is the error value at the previous iteration so need two more points. We use $e(1/2)$ and $e(1)$, this is enough to find the parameters a, b, and c of the parabola. We equate the values of $e(\alpha_k)$ and $\hat{e}(\alpha_k)$ at $\alpha_k = 0, 1/2, 1$. Then the three following equations gives us the solution:

$$\begin{array}{l} e(0) = c \\ e(\frac{1}{2}) = \frac{1}{4}a + \frac{1}{2}b + c \\ e(1) = a + b + c \end{array} \iff \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 2 & -4 & 2 \\ -3 & 4 & -1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} e(0) \\ e(\frac{1}{2}) \\ e(1) \end{bmatrix}. \tag{3.24}$$

We plug the values of a and b into Equation (3.13) to obtain

$$\alpha_k^o = -\frac{b}{2a} = \frac{3e(0) - 4e(1/2) + e(1)}{4\left[e(0) - 2e(1/2) + e(1)\right]}. \tag{3.25}$$

This gives us an estimate for the best step value α_k^o. Some typical experimental realization of this method are shown in Figure 3.1. The results in Figure 3.1 show that our approximation is valid around the region $0 \leq \alpha_k \leq 1$. The estimated optimum step size α_k^o is very close to the correct one.

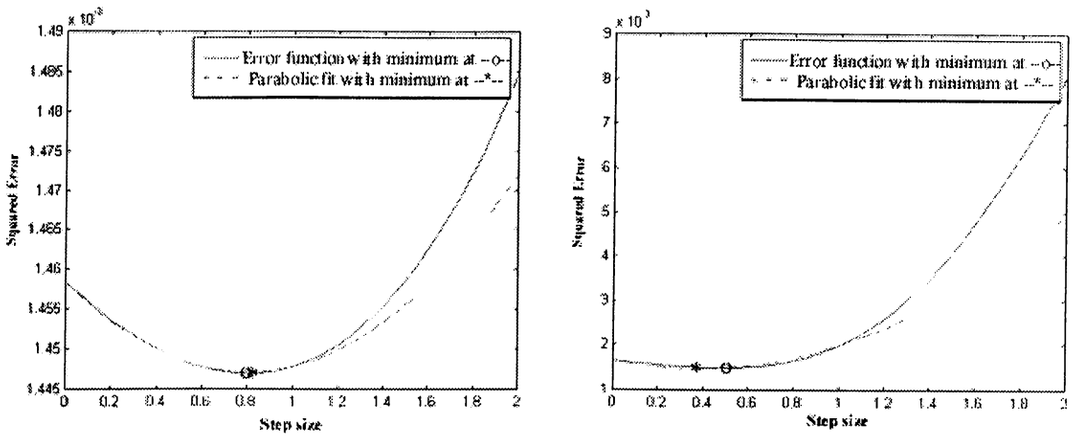

Figure 3.1: Two typical realization of the parabolic fit of the error function with the corresponding minimum value.

3.2.3 Simulation Studies

In this subsection, simulation studies are carried out to compare the performance of the suggested method with the EM algorithm. For this purpose, data from Gaussian mixture densities are generated at random. Simulations were carried out for both 1-D and 2-D GMMs. For each Gaussian mixture, 2000 observations are generated and used to obtain an estimate to the parameters of the original distribution. The estimation error criterion is defined as

$$e = \sum_x \left(p(x|\lambda) - p(x|\hat{\lambda}) \right)^2 \tag{3.26}$$

where λ represents the original Gaussian mixture density and $\hat{\lambda}$ represents the estimated one. Both 1-D and 2-D density cases are considered. For each case, 4-component and 12-component mixtures are used. For each experiment, 100 runs were made and the mean square error of the 100 runs is computed. The 100 runs are also divided into 10 groups of 10 runs such that in each group of 10 runs the GMM parameters are kept constant but 10 different sets of observations are generated, from which the histogram is computed. Figure 3.2 shows the

results for the 1-D and 2-D cases corresponding to 4 and 12 component GMMs.

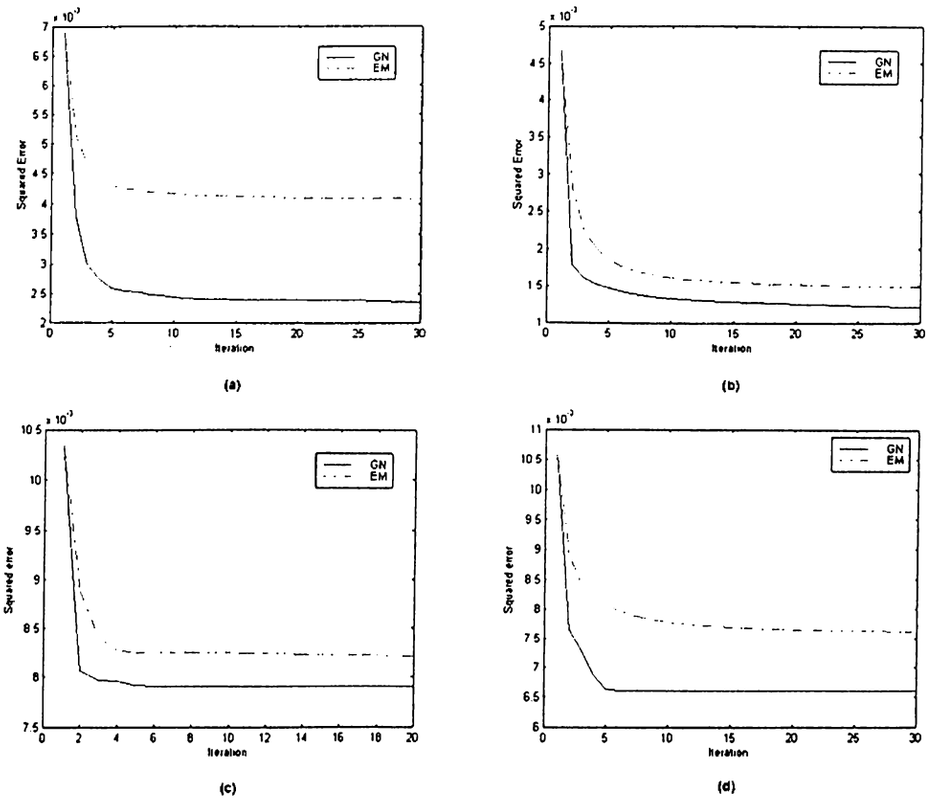

Figure 3.2: Mean square error obtained for GMM estimation using GN(straight line) and EM (dashed line) algorithms. (a) 1-D GMM, M=4, (b) 1-D GMM, M=12, (c) 2-D GMM, M=4, (d) 2-D GMM, M=12.

3.3 An Application

Probability density modeling using GMMs have been used in a wide range of applications ranging from speech and image processing to biology. In this

section, we consider some simple recent applications of GMMs to test the performance of the proposed Gauss-Newton based parameter estimation method compared to the EM-based one.

In [7], a probability density function of human skin color was estimated using a Gaussian mixture model whose parameters were estimated through the EM algorithm. The estimated density function has many applications in image and video databases. These applications range from human face detection to hand tracking.

A set of experiments similar to those described in [7] were carried out to estimate a probability density function of human skin color, but in our case the estimation of the GMM parameters was done using both the EM algorithm and the Gauss-Newton based method and the results are compared.

Typical human skin pictures were collected and human skin regions were extracted manually. A sample is shown in Figure 3.3. Each sample (skin color pixel) consists of three values (R,G,B). To reduce the dependence on the lighting condition, each sample is transformed from RGB to CIELUV color space and then the brightness component is discarded. The color space transformation from RGB to CIELUV in given in Appendix A. Figure 3.4 shows the resulting 2-D histogram of skin color (histogram of $x = (u, v)^T$).

As in [7], two Gaussian components were used to estimate the probability density function corresponding to the histogram shown in Figure 3.4. Figure 3.5 and Figure 3.6 show the estimated density function for EM and Gauss-Newton methods respectively. The corresponding estimation squared error is shown in Figure 3.7.

Figure 3.3: Human skin color pdf estimation, (a) original image, (b) skin pixels extracted for GMM training.

From Figure 3.5 and 3.6, we can see that the Gauss-Newton method results in a better model then the EM algorithm. This is confirmed by Figure 3.7 since the estimation squared error is smaller in the case of the Gauss-Newton estimation. Figure 3.7 also shows the improvements in performance obtained when using the optimum step size described in 3.2.2.

3.4 Conclusions

In this chapter, we described Gauss-Newton optimization technique based approach to obtain the parameters of a Gaussian Mixture Model. We experimentally demonstrated that this method provides a more accurate representation of the data compared to the widely used EM algorithm. Furthermore, this method often converges in less number of iterations.

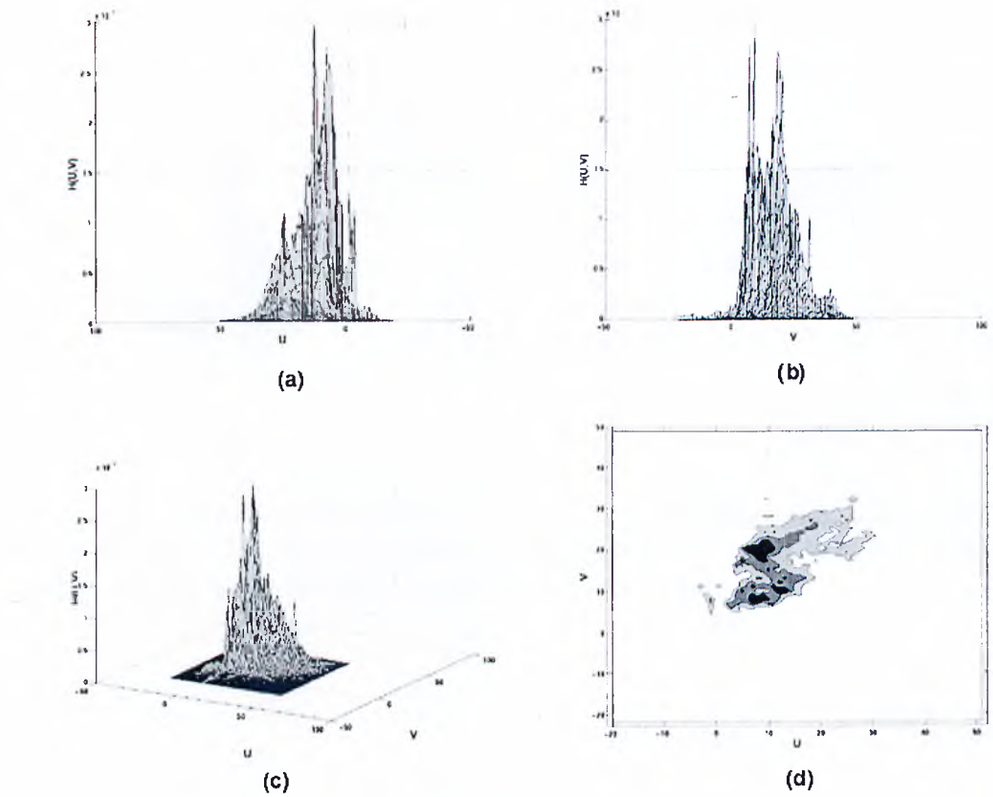

Figure 3.4: 2-D human skin color histogram in the UV space seen from different angles.

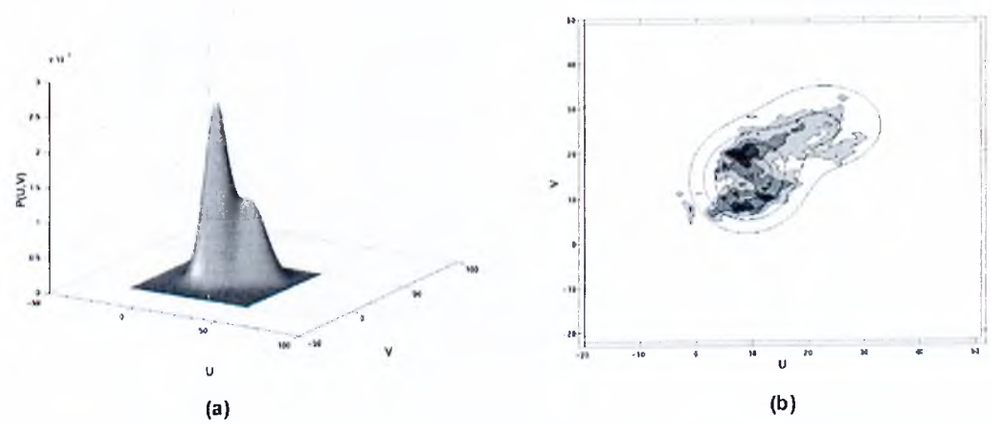

Figure 3.5: Estimated human skin color density function using EM algorithm, (a) 3-D view of the estimated histogram, (b) Top view of estimated histogram compared to the histogram shown in Figure 3.4d.

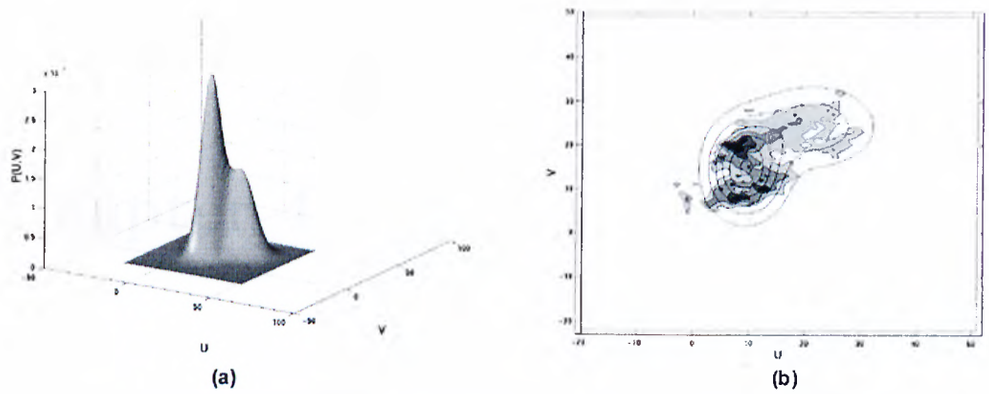

Figure 3.6: Estimated human skin color density function using Gauss-Newton algorithm, (a) 3-D view of the estimated histogram, (b) Top view of estimated histogram compared to the histogram shown in Figure 3.4d.

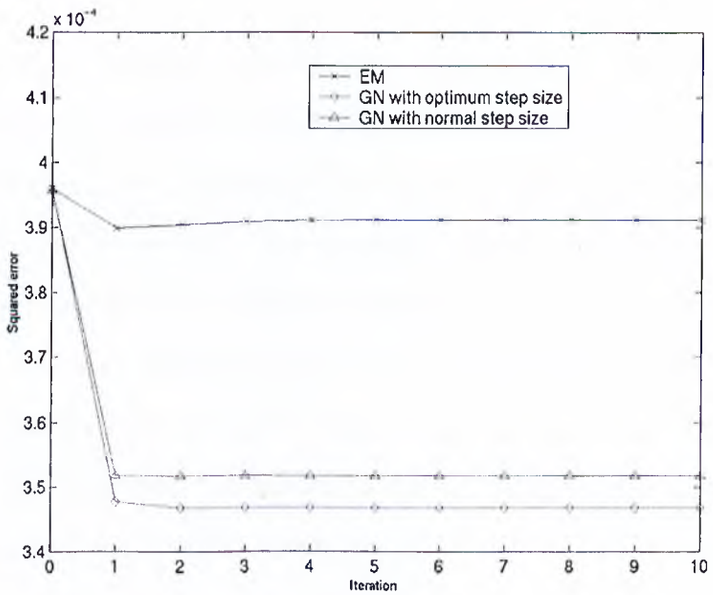

Figure 3.7: Skin color estimation squared error obtained at each iteration using EM algorithm, Gauss-Newton algorithm with normal perturbation step size and Gauss-Newton algorithm with estimated optimum perturbation step size.

Chapter 4

Matching Pursuit Based GMM Estimation

In this chapter, we develop a fast method for obtaining GMM parameter estimates for arbitrary probability densities. This method is based on the matching pursuit algorithm. As in Chapter 3, we use the histogram of the observation data for deriving our model. The matching pursuit algorithm is used to decompose the histogram into different Gaussian functions. This decomposition results in a Gaussian mixture density that can be used as an estimate to the probability density of the random vector under consideration. In section 4.1, we start by giving a brief description of the matching pursuit algorithm. The suggested method is presented in section 4.2.

4.1 Matching Pursuit Algorithm

Matching pursuit is a recently proposed algorithm for deriving signal-adaptive decompositions in terms of expansion functions chosen from an over-complete set called a dictionary -over-complete in the sense that the dictionary elements, also called atoms, exhibit a wide range of behaviors [13]. Roughly speaking, the matching pursuit algorithm is a greedy iterative algorithm which tries to determine an expansion for an arbitrary signal $x[n]$ given a dictionary of atoms, $g_\gamma[n]$, as follows

$$x[n] = \sum_{k=1}^{K} \alpha_k g_\gamma[n] \tag{4.1}$$

where the dictionary is a family of vectors (atoms) g_γ included in a Hilbert space $\mathbf{H}$ with a unit norm $\|g_\gamma\| = 1$ and γ is the set of parameters characterizing g_γ.

Matching pursuit algorithms are largely applied using dictionaries of Gabor atoms [14]. Gabor atoms are appropriate expansion functions for time-frequency signal decomposition, which are a scaled, modulated, and translated version of a single unit-norm window function, $g(.)$, which has the following form in continuous-time domain

$$g_\gamma(t) = \frac{1}{\sqrt{s}} g\left(\frac{t-\mu}{s}\right) e^{j\epsilon t} \tag{4.2}$$

where γ is the collection of parameters $\gamma = (s, \mu, \varepsilon) \in \Gamma = \mathrm{R}^+ \times \mathrm{R}^2$. Note that g_γ is centered in a neighborhood of μ whose size is proportional to s and its Fourier transform is centered at $\omega = \varepsilon$. This parametric model provides modification capabilities for time and frequency localization properties of signals.

4.2 Matching Pursuit Based Estimation

We want to use the matching pursuit algorithm with Gabor atoms to find a suitable decomposition to the speech features histogram. In our application, the modulation factor $e^{j\varepsilon t}$ in expression (4.2) is not necessary since the frequency localization has no meaning in this case, and thus it is dropped and we use

$$g_\gamma(t) = \frac{1}{\sqrt{s}} g\left(\frac{t-\mu}{s}\right) \tag{4.3}$$

Furthermore, if we choose $g(t)$ as a Gaussian function of zero mean and unit variance

$$g(t) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{t^2}{2}\right) \tag{4.4}$$

then we obtain

$$\begin{aligned} g_\gamma(t) &= \frac{1}{\sqrt{2\pi}\sqrt{s}} \exp\left(-\frac{(t-\mu)^2}{2s^2}\right) && (4.5) \\ &= \sqrt{s}.\mathcal{N}(\mu, s^2) && (4.6) \end{aligned}$$

which is a Gaussian function with mean μ and variance s^2 scaled by a factor $\sqrt{s}$. The discrete form of (4.5), is

$$g_\gamma[n] = \frac{1}{\sqrt{2\pi}\sqrt{s}} \exp\left(-\frac{(nN-\mu)^2}{2s^2}\right) \tag{4.7}$$

where and N is the sampling period. The resulting $g_\gamma[n]$ is a suitable decomposition function for our application.

In the following, we introduce a fast method for estimating the parameters of a GMM using the matching pursuit algorithm with decomposition functions derived in (4.7). Given an arbitrary D-dimensional random vector $\vec{x} = [x_1\ x_2\ \cdots\ x_D]^T$, we want to obtain a Gaussian mixture density

which approximates the distribution of $\vec{x}$, using a set of observation vectors $\bar{X} = \{\underline{x}_1, \underline{x}_2, ..., \underline{x}_T, \}$. Let us first write $\bar{X}$ as

$$\bar{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & & x_{1,T} \\ x_{2,1} & x_{2,2} & & x_{2,T} \\ & & & \\ x_{D,1} & & & x_{D,T} \end{bmatrix} = \begin{bmatrix} X_1 \\ X_2 \\ \\ X_D \end{bmatrix} \tag{4.8}$$

where X_i , $i = 1, ..., D$ are the sequences of training data corresponding to each of the D components of $\vec{x}$. For each X_i, we calculate the corresponding 1-D normalized histogram $H_i(x)$. If we can decompose $H_i(x)$ into a finite weighted sum of Gaussian components, we obtain a valid estimate to the distribution of x_i, the ith component of $\vec{x}$. The decomposition is done as follows. We first define our dictionary $\mathcal{D}$ as a family of vectors g_γ. The form of g_γ is given in (4.7). Each decomposition vector g_γ depends on the parameter $\gamma = (\mu, s)$. The range of μ can be obtained from the range of $H_i(x)$, while the range of s should be chosen experimentally. The dictionary should be large enough to cover a wide range of vectors. The algorithm starts by finding $g_{\gamma_{i,0}} \in \mathcal{D}$ that best matches $H_i(x)$ in the sense that the inner product $\left|\left\langle H_i, g_{\gamma_{i,0}} \right\rangle\right|$, which is a measure of similarity between $H_i(x)$ and $g_{\gamma_{i,0}}$, is maximized, i.e.,

$$\left|\left\langle H_i, g_{\gamma_{i,0}} \right\rangle\right| \geq \sup \left|\langle H_i, g_\gamma \rangle\right|. \tag{4.9}$$

Then, we can write

$$H_i = \left\langle H_i, g_{\gamma_{i,0}} \right\rangle g_{\gamma_{i,0}} + RH_i \tag{4.10}$$

where RH_i is the residual vector. The iteration then proceeds on RH_i as the initial vector. Suppose that $R^n H_i$ denotes the nth residual of H_i , at the nth iteration we get

$$R^n H_i = \left\langle R^n H_i, g_{\gamma_{i,n}} \right\rangle g_{\gamma_{i,n}} + R^{n+1} H_i \tag{4.11}$$

If we carry the iteration to order M, we obtain

$$H_i = \sum_{n=0}^{M-1} \left\langle R^n H_i, g_{\gamma_{i,n}} \right\rangle g_{\gamma_{i,n}} + R^M H_i \tag{4.12}$$

$$= \sum_{n=0}^{M-1} \alpha_{i,n} g_{\gamma_{i,n}} + R^M H_i \tag{4.13}$$

where $\alpha_{i,n} = \left\langle R^n H_i, g_{\gamma_{i,n}} \right\rangle$ and $\gamma_{i,n} = (\mu_{i,n}, s_{i,n})$. This gives us a decomposition of $H_i(x)$ as a weighted sum of Gaussian components. Let's examine the first term of the RHS of Equation (4.12). From Equation (4.5) we obtain

$$\sum_{n=0}^{M-1} \alpha_{i,n} g_{\gamma_{i,n}} = \sum_{n=0}^{M-1} \sqrt{s_{i,n}} \alpha_{i,n} . \mathcal{N}(\mu_{i,n}, s_{i,n}^2) \tag{4.14}$$

If we further define the weight $p_{i,n}$ as

$$p_{i,n} = \frac{\sqrt{s_{i,n}} \alpha_{i,n}}{\sum_{n=0}^{M-1} \sqrt{s_{i,n}} \alpha_{i,n}} \tag{4.15}$$

so that $\sum_{n=0}^{M-1} p_{i,n} = 1$, then we obtain a valid Gaussian mixture model for x_i

$$p(x_i|\lambda_i) = \sum_{n=0}^{M-1} p_{i,n} b_{i,n}(x_i) \tag{4.16}$$

where $b_{i,n}(x)$ is a Gaussian with mean $\mu_{i,n}$ and variance $s_{i,n}^2$ both obtained from the decomposition of $H_i(x)$.

If we carry out this procedure for all the individual components of $\vec{x}$, then we obtain D separate 1-dimensional models corresponding to the D components of $\vec{x}$: x_i , $i = 1, ..., D$. For 1-dimensional signal, this procedure results in one GMM that can be used as a model for the distribution of that signal. However, in higher dimensional cases the resulting 1-D models cannot be used to obtain the overall distribution directly, unless the individual components of the random vector are uncorrelated. For this case, the overall D-dimensional GMM can be obtained by multiplying the individual 1-D GMMs

$$p(\vec{x}|\lambda) = \prod_{i=1}^{D} p(x_i|\lambda_i). \tag{4.17}$$

For random variables with correlated components, this is not valid. One possible solution is to apply a transformation that decorrelates or at least minimizes the correlation between the individual components of $\vec{x}$. In speech processing for example, a Discrete Cosine Transform (DCT) is used to decorrelate the speech feature obtained from a speech signal. A DCT or a similar transform can be used to decorrelate (or at least minimize) the correlation between the individual components of the random vector. In this case, expression (4.17) can be used to obtain a good estimate to the overall distribution of the random vector.

In the next section, a fast calculation method is proposed to increase the algorithm's speed.

4.3 Fast Calculations

The matching pursuit can be implemented using a fast algorithm described in [23], that computes $\langle R^{n+1}H_i, g_\gamma \rangle$ from $\langle R^n H_i, g_\gamma \rangle$ with a simple updating formula. Consider Equation (4.11), which we can write as

$$R^{n+1}H_i = R^n H_i - \left\langle R^n H_i, g_{\gamma_{i,n}} \right\rangle g_{\gamma_{i,n}} \tag{4.18}$$

Take the inner product with g_γ on each side, we obtain

$$\left\langle R^{n+1}H_i, g_y \right\rangle = \langle R^n H_i, g_y \rangle - \left\langle R^n H_i, g_{\gamma_{i,n}} \right\rangle \left\langle g_{\gamma_{i,n}}, g_y \right\rangle \tag{4.19}$$

which is a simple updating formula for $\langle R^{n+1}H_i, g_y \rangle$. If we can calculate the inner product of all the atoms in the dictionary, $\langle g_\alpha, g_\beta \rangle$, and store it in a lookup table, then we can use this update formula to calculate $\langle R^{n+1}H_i, g_y \rangle$ at each iteration. The final algorithm is summarized below:

For each H_i , $i = 1, ..., D$

1. Set $n = 0$ and compute $\{\langle H_i, g_\gamma \rangle\}_{\gamma \in \Gamma}$
2. Find $g_{\gamma_{i,0}} \in \mathcal{D}$ such that: $\left|\langle H_i, g_{\gamma_{i,0}} \rangle\right| \geq \sup |\langle H_i, g_\gamma \rangle|$ for all $\gamma \in \Gamma$
3. Update for all $g_{\gamma_{i,n}} \in \mathcal{D}$:

   $\langle R^{n+1} H_i, g_y \rangle = \langle R^n H_i, g_y \rangle - \langle R^n H_i, g_{\gamma_{i,n}} \rangle \langle g_{\gamma_{i,n}}, g_y \rangle$
4. If $n < M - 1$ increment n and go to 2.

4.4 Experimental Results and Discussion

Even though the proposed method provides a less accurate model then the EM algorithm for random variables with correlated entries, its low computational complexity makes desirable. This method is especially useful for applications where speed is important. We have tried the matching pursuit based method in the application of speaker identification and the rates of recognition were compared to those obtained using the EM algorithm. Table 4.1 summarizes the results for a 30-speaker set with a training sequence of 40 sec for each speaker. The simulations were carried out on an Intel-Pentium based PC using MATLAB. The rates of identification in the matching pursuit case are less than those for the EM algorithm. However the model training time for the 30-speaker set is extremely lower than that required by the EM algorithm.

	Using EM	Using MP
Identification rate	77.8%	71.5%
Training time	42 min	8 min

Table 4.1: Speaker identification rate using EM algorithm and the matching pursuit algorithm. The model training time corresponds to a set of 30 speakers and 40 sec speech signals per speaker.

Chapter 5

Speaker Recognition Using GMM and a Nonlinear Frequency Scale

In general, the field of speaker recognition can be classified into two sub-areas: verification and identification. Recognition rate in both cases largely depends on extracting and modeling the speaker dependent nature of the speech signal, which can effectively distinguish one speaker from another. The most widely used speech feature parameter set is based on the Mel-scale Cepstrum: the Mel Frequency Cepstral Coefficients (MFCC) [16]. The MFCC features are obtained from a Mel-frequency division of the short-time speech spectra and produce very good results for speech recognition [22], as Mel-scale division of the spectrum is compatible with the human auditory system. In Section 5.1, we give a brief review of MFCC feature extraction and the use of Gaussian

Mixture Models (GMM) in speaker recognition for modeling the distribution of MFCC features.

In [17], we observed that the use of scales other than the Mel-scale may be more advantageous for speaker recognition applications. In this chapter, we propose the use of a nonlinear frequency scale for speaker recognition applications. In the modified Mel-scale, more emphasis is given to frequencies around 2 kHz. The idea of modifying the Mel-scale was originally proposed in [24] and used successfully in accent classification. In Section 5.2, we introduce a new nonlinear frequency scale for speaker recognition and its computation using the FFT domain filter bank. Section 5.3 describes the computation of the same frequency scale using a subband wavelet packet transform. Experimental results for speaker identification are given in Section 5.4.

5.1 Mel-Frequency Cepstral Coefficients

In speaker recognition, the use of speech spectrum has been shown to be very effective [21]. This is mainly due to the fact that the spectrum reflects vocal tract structure of a person which is the main physiological system that distinguishes one person's voice from another. Recently, cepstral features computed directly from the spectrum are found to be more robust in speech and speaker recognition, especially for noisy speech [19].

In speaker recognition systems, the Mel-frequency cepstral-coefficients (MFCC's) are usually used as features to characterize the speech signal [16]. Briefly, the MFCC's are computed by smoothing the Fourier transform spectrum by integrating the spectral coefficients within triangular bins arranged on

a non-linear scale called the Mel-scale shown in Figure 5.1. This scale tries to imitate the frequency resolution of the human auditory system which is linear up to 1 kHz and logarithmic thereafter. In order to make the statistics of the estimated speech power spectrum approximately Gaussian, logarithmic compression is applied to the energy obtained from each frequency bin. Finally, the Discrete Cosine Transform (DCT) is applied in order to compress the spectral information into the lower-order coefficients. Moreover, the DCT de-correlates these coefficients allowing the subsequent statistical modeling to use diagonal covariance matrices.

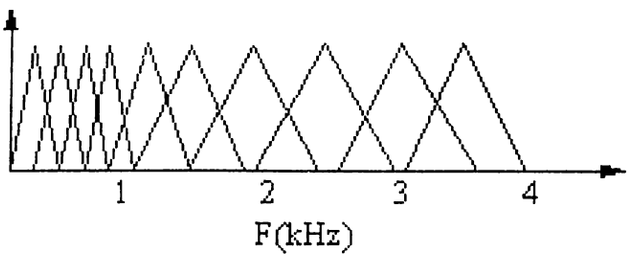

Figure 5.1: Triangular bins arranged on a Mel-scale for MFCC features extraction.

Gaussian Mixture Models (GMM) have been used very widely in speaker recognition applications for modeling speaker identity. Short-term (usually 20 ms) speaker-dependent feature vectors are first obtained from the speech signal, then GMM is used to model the density of these vectors. The individual Gaussian components of a GMM are shown to represent some general speaker-dependent spectral shapes that are efficient for modeling speaker identity. For speaker identification, each speaker is represented by a GMM and is referred to by his/her model.

5.2 New Nonlinear Frequency Scale

The Mel-scale, which is approximately linear below 1 kHz and logarithmic above, is more appropriate than linear scale for speech recognition performance across frequency bands. This scale tries to imitate the frequency resolution of the human auditory system which is linear up to 1 kHz and logarithmic thereafter. However, in [17], we observed that the use of scales other than the Mel-scale may be more advantageous for speaker recognition applications.

In [24], properties of various frequency bands in the range between 0-4 kHz was investigated for accent classification. It was shown that for speech recognition applications the lower range frequencies, mainly between 200-1500 Hz, have most of the relevant information. This explains the use of Mel-scale for speech recognition. While for accent classification applications, it was shown that the most relevant frequency band lies around 2 kHz. This suggests that mid-range-frequencies (1500-2500 Hz) contribute more to accent classification performance. Following these results, a new frequency axis scale was formulated for accent classification [24], which is shown in Figure 5.2. Since a large number of filter banks are concentrated in the mid-range frequencies, the output coefficients are better able to emphasize accent-sensitive features.

Similarly, the frequency range that is most relevant for speaker recognition is investigated in this paper. Then a scale that gives more emphasis to this frequency range is formulated.

In [24], a series of experiments were performed to investigate the accent discrimination ability of various frequency bands. We carried out similar experiments to investigate the importance of different frequency bands in speaker

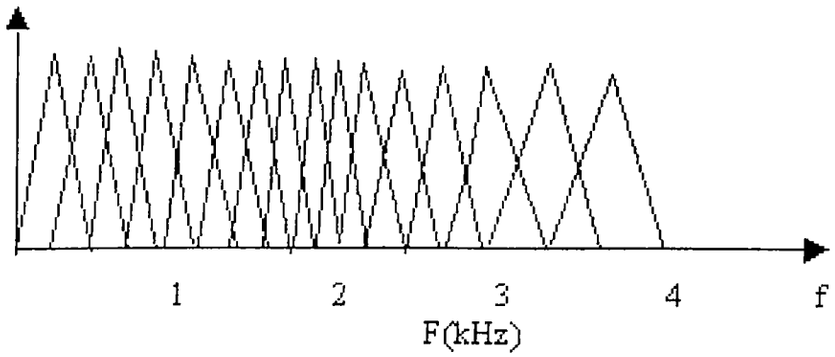

Figure 5.2: A sampling scheme for filter banks which is more sensitive to accent characteristics.

recognition. The frequency axis (0-4 kHz) was divided into 16 uniformly spaced frequency bands. The energy in each frequency band was weighted with a triangular window. The output of each filter bank was used as a single parameter in generating a GMM for each speaker. Figure 5.3 shows speaker identification performance across the 16 linearly spaced frequency bands. Unlike speech recognition, the most relevant frequency band lies slightly above 2 kHz.

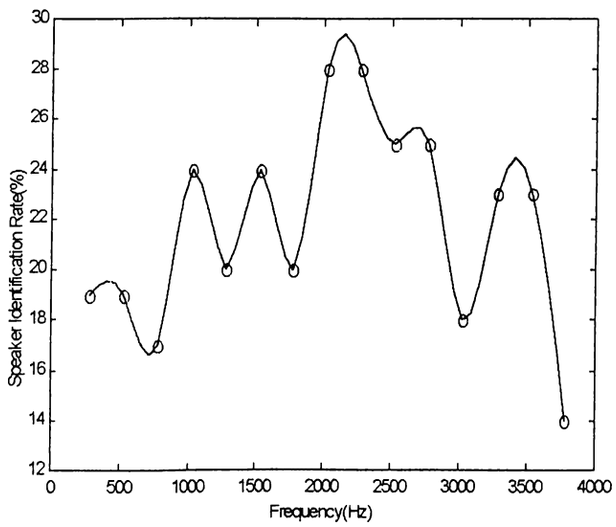

Figure 5.3: Speaker identification performance based on the energy in different frequency bands.

In accordance with the previous result, a new frequency axis scale is derived. The scale is shown in Figure 5.4. Since a relatively large number of filters (windows) are concentrated in the midrange frequencies, the output coefficients are better able to emphasize speaker-dependent features. The 16 center frequencies of the filters which range between 0-4 kHz are also given in Table 5.1.

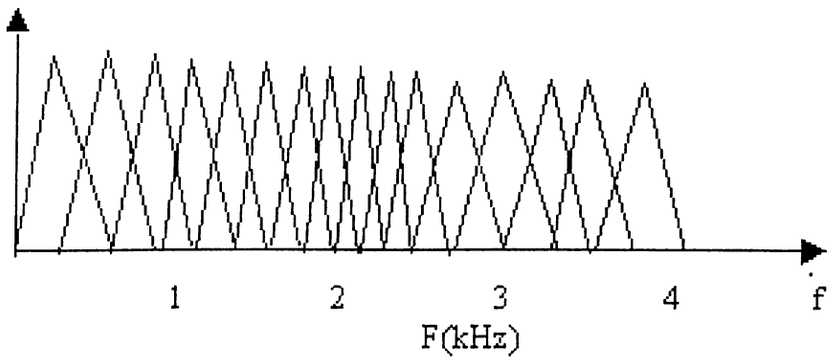

Figure 5.4: A new frequency axis division suitable for speaker recognition applications.

Filter #	Center frequency (Hz)	Filter #	Center frequency (Hz)
1	350	9	2100
2	700	10	2220
3	1000	11	2390
4	1250	12	2600
5	1450	13	3000
6	1650	14	3300
7	1850	15	3500
8	2000	16	3700

Table 5.1: Center frequencies (Hz) of triangular windows shown in Figure 5.4.

Usually a pre-emphasis, shown in Figure 5.5, is applied to the magnitude spectrum from each speech frame. This pre-emphasis gives more importance to mid-range and high frequencies and has proven to be effective in speaker recognition.

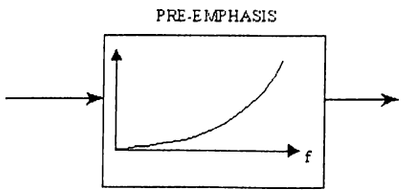

Figure 5.5: Pre-emphasis applied to speech frames.

After the pre-emphasis and log compression, the cepstral features are computed using the discrete cosine transform (DCT).

5.3 Subband Decomposition (Wavelet) Based Computation of Features

The wavelet analysis associated with a corresponding decomposition filterbank is proposed in [18] to obtain a scale similar to the one derived in the previous section. The implementation of the wavelet packet transform can differ according to the application. In this case, a tree structure which uses a single basic building block is used repeatedly until the desired decomposition is accomplished [19]- [20]. This single block, shown in Figure 5.6, divides the frequency range of the input into two half-bands.

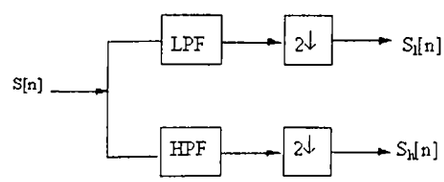

Figure 5.6: Basic block of a subband decomposition.

The pass-bands for the low-pass and high-pass filters are $[0, \frac{\pi}{2}]$ and $[\frac{\pi}{2}, \pi]$, respectively. One possible choice for these filters is the 7^{th} order Lagrange filters having transfer functions

$$H_l(z) = \frac{1}{2} + \frac{9}{32}(z^{-1} + z) - \frac{1}{32}(z^{-3} + z^3) \tag{5.1}$$

$$H_h(z) = \frac{1}{2} - \frac{9}{32}(z^{-1} + z) + \frac{1}{32}(z^{-3} + z^3) \tag{5.2}$$

Using subband decomposition, a frequency domain decomposition similar to Mel-scale can be obtained [18], the corresponding scale is shown in Figure 5.7. In [18], the resulting cepstral coefficients are called SUBCEP's.

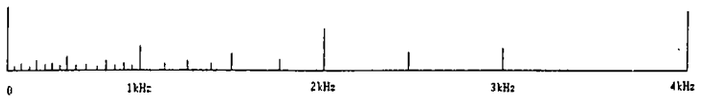

Figure 5.7: Subband decomposition approximation to Mel-scale.

In speaker recognition within the telephone bandwidth, the frequency range 0-4 kHz is decomposed in a manner to give more emphasis to mid-range frequencies between 2 and 2.75 kHz. The corresponding frequency domain decomposition is shown in Figure 5.8.

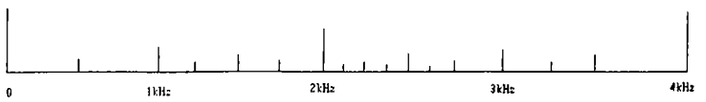

Figure 5.8: Frequency scale for speaker recognition using subband decomposition.

5.4 Experimental Study and Conclusions

5.4.1 Database Description

The experiments were carried out on the POLYCOST 250 database (v1.0). The POLYCOST database is dedicated to speaker recognition applications [26]. The main purpose behind it is to provide a common database on which speaker recognition algorithms can be compared and validated. The database was recorded from 134 subjects coming from 14 European countries. Around 10 sessions were recorded for each subject, each session contains 14 items. The recordings were made over the telephone network with an 8 kHz sampling frequency. In [26], a set of baseline experiments is defined for which results should be included when presenting evaluations made on this database. Our experiments follow the set of rules defined in [26] under "text-independent speaker identification" .

5.4.2 Experimental Results

A set of experiments was carried out to analyze the performance of the proposed frequency scale for speaker identification. The speech signal is first analyzed, and the silence periods are removed. Then the signal is divided into overlapping frames of approximately 20 ms length and a spacing of 10 ms. For each frame, 12 speech features are extracted. The experiments were done using features obtained from both methods described previously, i.e., cepstral features computed via Fourier analysis and wavelet analysis (SUBCEP) using frequency scales shown in Figure 5.4 and Figure 5.8, respectively. Table 5.1

shows the results obtained for both methods for different frequency scales. The first column is computed using the DFT while the second column is computed using wavelet analysis.

	Recognition rate using DFT analysis based cepstral features	Recognition rate using wavelet analysis based cepstral features
Mel-scale	77.8%	78.4%
Modified Mel-scale derived for accent classification	78.9%	79.5%
Frequency scale derived for speaker Identification	79.5%	80.7%

Table 5.2: Speaker identification performance for different frequency domain scales using MFCC and SUBCEP features.

The results obtained in Table II confirm that the Mel-scale is not appropriate for speaker recognition applications. In fact, Mel-scale performs slightly better than a uniform decomposition of the frequency domain. The frequency scale derived in [24] for accent classification performs better than the Mel-scale. This is mainly due to the fact that this scale emphasizes mid range frequencies which are important for speaker recognition. Finally, the new derived frequency scale for speaker identification performs the best among the three scales. This scale emphasizes exactly the frequency bands that are most significant for speaker identity.

The experimental results also show that the use of wavelet analysis for feature extraction performs slightly better than MFCC's which are computed using DFT. We finally conclude that the choice of the frequency domain scale should depend primarily on the type of application under consideration. For speaker recognition, it is shown that mid-range and some high frequency components are more important for representing speaker identity.

Chapter 6

Conclusion

In this thesis, the design of Gaussian mixture models for arbitrary densities was studied. Estimation of model parameters is one of the most important issues in GMM design. The Expectation-Maximization algorithm is widely in literature as a method for estimating these parameters. GMM parameters are usually estimated from a set of observed data. Since the density function to be estimated should be close to the histogram of the observed data in shape, the latter can be used to derive good model estimate. In this work, we proposed two new methods for estimating GMM parameters based on this approach, which overcome some drawbacks of the EM algorithm.

The first method is based on least squares estimation. The least squares criterion is used to minimize an error function based on the difference between the observation data histogram and the estimated density. The minimization is carried out using the Gauss-Newton optimization technique. This technique usually needs a very few number of iterations to converge. Simulations results

have shown that the model estimated using the proposed method is more accurate than the EM based model, in the sense that the mean squared error between the estimated density and the data histogram is lower. In the optimization procedure, the step size related to the perturbation vector used in the parameter update formula, has an important effect on the convergence speed and the final model error. We have provided a simple method for obtaining an estimate to the perturbation step size at each iteration of the Gauss-Newton algorithm. Experimental results showed an increase in model convergence speed and accuracy when this method was applied.

Human skin color distribution modeling was used as an experimental example to the application of the suggested method. The results showed a significant increase in the model accuracy when our method is used instead of the EM based method.

In the second method, we used the matching pursuit algorithm to decompose the histogram of the observation data with a proposed set of decomposition functions. The decomposition results in a set of weighted Gaussian functions which was used to obtain a GMM for the density function of the process under consideration. This method provides a fast way to obtain GMM parameter estimates. In the application of speaker identification, the proposed method resulted in a less accurate model then the EM algorithm but the required training time was remarkably lower. Still, the matching pursuit based method further needs to be investigated for applications where speed is important. The use of this method may be advantageous in applications like speaker adaptation.

In Chapter 5, we developed a new set of speech feature parameters that are more appropriate for speaker recognition applications than the commonly used Mel-scale based features. The proposed features are based on a nonlinear division of the frequency scale that gives more importance of mid-range frequencies around 2 kHz. In a set of experiments on speaker identification, we found that the suggested set of features results in some increase in the identification rate.

Each of the proposed GMM parameter estimation methods has its advantage. In fact, the two methods can be exploited together in one system. For example, in an application, the model obtained by the matching pursuit method can be used a good initial point for the Gauss-Newton based method. This can result in a faster convergence of the optimization procedure. Another interesting application is to use the Gauss-Newton method to obtain a starting model for our process, then whenever new data comes the matching pursuit method can be used to adapt the existing model to the new data in a fast manner. The adaptation procedure can be done using a method called modeling weighting. Briefly, what modeling weighting does is whenever there is new adaptation data the final model is calculated as a weighted sum of the original model and the model derived from the adaptation data. A smaller weight is given to the adaptation model, also a forgetting factor can be inserted with time.

Appendix A

RGB to CIELUV Color Space Conversion

A.1 CIELUV Color Space

This is based on CIE Yu'v' (1976) and is a further attempt to linearize the perceptibility of unit vector color differences. It is a non-linear color space, but the conversions are reversible. Coloring information is centered on the color of the white point of the system, (D65 in most TV systems). The non-linear relationship for Y* is intended to mimic the logarithmic response of the eye.

RGB color values cannot be transformed directly to CIELUV. Instead, they should be first converted to CIE XYZ, then CIELUV values can be computed from CIE XYZ.

A.2 RGB to XYZ Conversion

RGB values in a particular set of primaries can be transformed to and from CIE XYZ via a 3×3 matrix transform. These transforms involve tristimulus values, that is a set of three linear-light components that conform to the CIE color-matching functions. CIE XYZ is a special set of tristimulus values. In XYZ, any color is represented as a set of positive values.

To transform from RGB to XYZ, the matrix transform used is:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.189423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{A.1}$$

The range for valid R, G, B values is [0,1].

A.3 XYZ to CIELUV Conversion

CIE 1976 L*u*v* (CIELUV) is based directly on CIE XYZ and is another attempt to linearize the perceptibility of color differences. The non-linear relations for L*, u*, and v* are given below:

$$L^* = 903.3(Y/Y_n) \quad \text{for} \quad Y/Y_n \leq 0.008856 \tag{A.2}$$

$$L^* = 116(Y/Y_n)^{\frac{1}{3}} - 16 \quad \text{for} \quad Y/Y_n > 0.008856 \tag{A.3}$$

$$u^* = 13L^*(u' - u'_n) \tag{A.4}$$

$$v^* = 13L^*(v' - v'_n) \tag{A.5}$$

L* scales from 0 to 100 for relative luminance (Y/Y_n) scaling 0 to 1. Here X_n, Y_n and Z_n are the tristimulus values of the reference white. The quantities

u'_n and v'_n refer to the reference white or the light source; for the 2^o observer and illuminant C, $u'_n = 0.2009$, $v'_n = 0.4610$. Equations for u′ and v′ are given below:

$$u' = 4X/(X + 15Y + 3Z) \quad (A.6)$$

$$v' = 9Y/(X + 15Y + 3Z) \quad (A.7)$$

Bibliography

[1] D. A. Reynolds and R. C. Rose, "Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models," *IEEE Trans. Speech Audio Processing*, vol. 3, no. 1, pp. 72–83, Jan. 1995.

[2] S.J. Roberts, D. Husmeier, I. Rezek, and W. Penny, "Bayesian Approaches to Gaussian Mixture Modelling," *IEEE Transaction on Pattern Analysis and Machine Learning*, August 1998.

[3] F. Beaufays, M. Weintraub, and Y. Konig, "Discriminative Mixture weight Estimation for Large Gaussian Mixture Models," *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP95*, vol. 1, 1995.

[4] L. Liu and J. He, "On the Use of Orthogonal GMM in Speaker Recognition," *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP99*, vol. 2, 1999.

[5] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from Incomplete Data via the EM Algorithm," *J. Royal Stat. Soc.*, vol. 39, pp. 1–38, 1997.

[6] L. Xu and M.I. Jordan, "On Convergence Proprieties of the EM Algorithm for Gaussian Mixtures," *Neural Computation 8*, pp 129–151,1996.

[7] M.-H. Yang and N. Ahuja, "Gaussian Mixture Modeling of Human Skin Color and Its Applications in Image and Video Databases," *Proceedings of the SPIE99*, vol. 3656, pp. 458–466, San Jose, Jan. 1999.

[8] Y. Raja, S. J. McKenna and S. Gong, "Tracking and Segmenting People in Varying Lighting Conditions using Colour," *Proceedings of FG'98*, April 14–16, 1998 in Nara, Japan.

[9] Y. Raja, S.J. McKenna and S. Gong, "Segmentation and Tracking Using Colour Mixture Models," *Asian Conference on Computer Vision (ACCV)*, January 1998, Hong Kong, Lecture Notes in Computer Science 1351, Vol. I, pp. 607–614.

[10] P.J. Castellano, S. Slomka, and S. Sridharan, "Telephone Based Speaker Recognition Using Multiple Binary Classifier and Gaussian Mixture Models," *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP97*, pp 1075–1078, 1997.

[11] J.A. Cadzow, "Least Squares Error Modeling with Signal Processing Applications," *IEEE-ASSP Magazine*, pp. 12–31, October 1990.

[12] G.H. Gloub and V. Pereyra, "The Differentiation of Pseudo-inverse and Non-linear Least Squares Problems whose Variables Separate," *SIAM Journal Numerical Analysis*, pp. 413–432, April 1973.

[13] S. Mallat and Z. Zhang, "Matching Pursuits with Time-Frequency Dictionaries," *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.

[14] R. Gribonval, Ph. Depalle, X Rodet, E. Barcy, and S. Mallat, "Sound Signals Decomposition Using a High Resolution Matching Pursuit," *ICMC'96.*

[15] E. Gündüzhan, *Coding of Speech and Image Signals Using Gabor Decomposition,* M.S. Thesis, Bilkent University, 1994.

[16] S. B. Davis and P. Mermelstein, "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-28, pp. 357–366, Aug. 1980.

[17] J. Nam, A. Enis Cetin, A. H. Tewfik, "Speech and Visual Streams Analysis Based Video Sequence Segmentation," *IEEE International Conf. on Image Proc (ICIP)'97,* Santa Barbara, CA, October 1997.

[18] E. Erzin, A. Enis Cetin, and Y. Yardimci, "Subband Analysis for Robust Speech Recognition in the Presence of Car Noise," *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing 1995 (ICASSP '95),* May, 1995.

[19] F. Jabloun, A. Enis Cetin and E. Erzin, "Teager Energy Based Feature Parameters for Speech Recognition in Car Noise," accepted for publication, *IEEE Signal Processing Letters.*

[20] R. Sarikaya, B.L. Pellom, and J.H.L. Hansen, "Wavelet Packet Transform Features with Application to Speaker Identification," *IEEE Nordic Signal Processing Symposium (NORSIG'98),* Vigso, Denmark, pp. 81–84, June, 1998.

[21] M.S. Zilovic, R.P. Ramachandran, and R.J. Mammone, "Speaker Identification Based on the Use of Robust Cepstral Features Obtained from Pole-Zero Transfer Functions," *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 3, May 1998.

[22] "A Review of Large Vocabularly Continuous-speech Recognition," *IEEE Processing Magazine*, Spetember 1996.

[23] Stephane Mallat, *Wavelet Tour of Signal Processing,* Academic Press, 1998.

[24] L.M. Arslan, *Foreign accent classification in American English, PhD thesis*, Duke University, July 1996

[25] H. Altincay, "Experimental Work on Classifier Combination for Speaker Identification," *EUROSPEECH'99 Conference*, Budapest, Sep. 1999.

[26] H. Melin and J. Lindberg, "Guidelines for Experiments on the POLYCOST Database," Version 1.0, January 8th, 1997.

[27] W. Mistretta and K. Farell, "Model Adaptation Methods for Speaker Verification," *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP98*, pp 113–116, 1998.

[28] T.J. Hazen and J.R Glass, "A Comparison of novel Techniques for Instantaneous Speaker Adaptation".