

**STATISTICAL FACIAL FEATURE EXTRACTION AND LIP
SEGMENTATION**

by
MUSTAFA BERKAY YILMAZ

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of
Master of Science

Sabanci University

August 2009

STATISTICAL FACIAL FEATURE EXTRACTION AND LIP SEGMENTATION

APPROVED BY

Assist. Prof. Dr. Hakan ERDOĞAN
(Thesis Advisor)

Assoc. Prof. Dr. Mustafa ÜNEL
(Thesis Co-Advisor)

Assist. Prof. Dr. Kemalettin ERBATUR

Assist. Prof. Dr. Yücel SAYGIN

Assoc. Prof. Dr. Berrin YANIKOĞLU

DATE OF APPROVAL:

©Mustafa Berkay Yılmaz 2009

All Rights Reserved

to my family

Acknowledgements

I am sincerely grateful to my thesis advisor Hakan Erdoğan and my co-advisor Mustafa Ünel for their invaluable guidance, support, patience and encouragement throughout my thesis.

I would like to thank TÜBİTAK for providing the necessary financial support for my masters education.

I am thankful to the members of Sabanci University for their help and friendship.

I would like to thank my thesis jury members Kemalettin Erbatur, Yücel Saygın and Berrin Yanıkođlu for allocating their valuable time to my thesis for reading and reviewing it.

STATISTICAL FACIAL FEATURE EXTRACTION AND LIP SEGMENTATION

MUSTAFA BERKAY YILMAZ

ME, M.Sc. Thesis, 2009

Thesis Advisor: Hakan Erdoğan

Thesis Co-Advisor: Mustafa Ünel

Keywords: facial features, probability model, optimization, joint distribution, Gaussian mixture models, shape priors

Abstract

Facial features such as lip corners, eye corners and nose tip are critical points in a human face. Robust extraction of such facial feature locations is an important problem which is used in a wide range of applications including audio-visual speech recognition, human-computer interaction, emotion recognition, fatigue detection and gesture recognition.

In this thesis, we develop a probabilistic method for facial feature extraction. This technique is able to automatically learn location and texture information of facial features from a training set. Facial feature locations are extracted from face regions using joint distributions of locations and textures represented with mixtures of Gaussians. This formulation results in a maximum likelihood (ML) optimization problem which can be solved using either a gradient ascent or Newton type algorithm. Extracted lip corner locations are then used to initialize a lip segmentation algorithm to extract the lip contours. We develop a level-set based method that utilizes adaptive color distributions and shape priors for lip segmentation. More precisely, an implicit curve representation which learns the color information of lip and non-lip points from a training set is employed. The model can adapt itself to the image of interest using a coarse elliptical region. Extracted lip contour provides detailed information about the lip shape.

Both methods are tested using different databases for facial feature extraction and lip segmentation. It is shown that the proposed methods achieve better results compared to conventional methods. Our facial feature extraction method outperforms the active appearance models in terms of pixel errors, while our lip segmentation method outperforms region based level-set curve evolutions in terms of precision and recall results.

İSTATİSTİKSEL YÜZ ÖZNİTELİK ÇIKARIMI VE DUDAK BÖLÜMLEMESİ

MUSTAFA BERKAY YILMAZ

ME, Yüksek Lisans Tezi, 2009

Tez Danışmanı: Hakan Erdoğan

Yardımcı Danışman: Mustafa Ünel

Anahtar Kelimeler: yüz öznitelikleri, olasılık modeli, optimizasyon, ortak dağılım, Gauss karışım modeli, şekil öncelikleri

Özet

Yüz öznitelikleri; insan yüzündeki dudak köşeleri, göz köşeleri ve burun ucu gibi kritik noktalar. Bu tür yüz özniteliklerinin konumlarının sağlam çıkarımı; görsel-işitsel konuşma tanıma, insan-bilgisayar etkileşimi, duygu tanıma, yorgunluk algılaması ve hareket tanımayı da kapsayan geniş bir uygulama alanına sahiptir.

Bu tezde, yüz özniteliklerinin çıkarılması için bir olasılık modeli geliştirilmiştir. Bu teknik, yüz özniteliklerinin konum ve doku bilgisini bir eğitim kümesinden otomatik olarak öğrenebilir. Yüz özniteliklerinin konumları, konum ve doku bilgisinin ortak dağılımını temsil eden Gauss karışımlarını kullanarak yüz bölgelerinden çıkarılır. Bu formülasyon, rampa tırmanışı veya Newton türü bir algoritma ile çözülebilecek bir maksimum ihtimal problemi ile sonuçlanır. Çıkarılmış dudak köşeleri daha sonra dudak hatlarını çıkarmak amacıyla bir dudak bölümlemesi algoritmasını başlatmak için kullanılır. Dudak bölümlemesi için, uyarlamalı renk uzayı ve şekil önceliklerinden yararlanan düzey-kümesi tabanlı bir yöntem geliştirilmiştir. Daha detaylı olarak, dudak ve dudak dışı noktaların renk bilgilerini bir eğitim setinden öğrenebilen örtük bir eğri temsilcisi kullanılmıştır. Bu model, kaba bir eliptik bölge kullanarak kendini ilgili resme uyarlayabilir. Çıkarılan dudak hatları, dudak şekliyle ilgili detaylı bilgi sağlar.

Yüz öznitelikleri çıkarımı ve dudak bölümlenmesi yöntemleri farklı veritabanları kullanılarak test edilmiştir. Önerilen yöntemlerin, geleneksel yöntemlere göre daha iyi sonuçlar verdiği gösterilmiştir. Yüz öznitelikleri çıkarma yöntemi, piksel hatası bazında aktif görünüm modellerinden daha iyi sonuç vermiştir. Dudak bölümlenme yöntemi ise, duyarlık ve akletme bazında bölge tabanlı düzey-kümesi eğri gelişiminden daha iyi sonuç vermiştir.

Table of Contents

Acknowledgments	v
Abstract	vi
Ozet	viii
1 Introduction	1
1.1 Motivation	1
1.2 Literature Review	2
1.3 Contributions	8
1.4 Outline	8
2 Facial Feature Extraction	9
2.1 Face Detection and Normalization	10
2.1.1 Face Detection	10
2.1.2 Face Normalization	11
2.2 Probabilistic Facial Feature Extraction	11
2.2.1 Independent Features Model	12
2.2.2 Dependent Locations Model	12
2.2.3 Representing Locations and Texture	13
2.2.4 Modeling Locations and Texture	15
2.2.5 Algorithm	16
2.2.6 Experimental Results	19
3 Lip Segmentation	27
3.1 Lip Segmentation Using Adaptive Color Space Training	27
3.1.1 Probabilistic Modeling	28
3.1.2 Adaptation	28
3.1.3 Testing	29
3.1.4 Curve Evolution	30
3.1.5 Color Spaces	31
3.2 Addition of Shape Priors	32
3.2.1 Shape Priors Using Linear Subspace	32
3.3 Experimental Results	34
3.3.1 Database	34
3.3.2 Performance Metric	34

3.3.3	Results	35
4	Conclusions and Future Work	39
4.1	Conclusions	39
4.2	Future Work	40
	Bibliography	41

List of Figures

2.1	Facial features used in this work	9
2.2	Example face detection result	10
2.3	Obtained multiple elliptic face regions	10
2.4	Search regions for facial features	23
2.5	Facial feature locations obtained using independent and dependent locations models, with a good independent model initialization	26
2.6	Facial feature locations obtained using independent and dependent location models, with an inaccurate independent model initialization .	26
3.1	Potential field surface and segmented lip boundary	30
3.2	Sampling and score mapping	30
3.3	Binary image and corresponding ROI	33
3.4	Example face image and its corresponding binary lip image	34
3.5	Segmented boundary and its precision-recall image	35

List of Tables

2.1	PCA subspace dimension and window size parameters used for facial features	22
2.2	Number of mixtures used for different GMMs	23
2.3	Pixel errors for independent model	25
2.4	Pixel errors for dependent model with gradient ascent optimization .	25
2.5	Pixel errors for dependent model with Newton optimization	25
2.6	Pixel errors for AAM	26
3.1	Precision and recall results	36
3.2	Adaptive color space segmentation results	37
3.3	Region based gray-level segmentation results	38

Chapter 1

Introduction

1.1 Motivation

Extraction of facial feature locations is an important problem in computer vision. There are many uses of facial feature extraction from a face image such as automatic visual emotion detection, gesture recognition, pupil tracking, driver fatigue detection for safe driving, face detection and recognition using a bottom-up approach, facial image compression and low-bit video coding.

Our main motivation to work on this problem is to improve the performance of audio-visual automatic speech recognition. Especially in noisy environments, visual information is complementary to auditory information. In order to extract visual features, one either needs to find the lip region or lip boundary. We worked on both finding the lip corners and extraction of lip boundaries in this thesis. Since other facial features such as eye corners can help in finding lip corners, we decided to develop an algorithm for general facial feature extraction.

Among many different approaches aiming at robust facial feature extraction in literature, we followed a statistical model to learn appearance and shape of facial features in a natural way. This way we hope to learn other dependencies between facial features such as geometric constraints, symmetry properties, expected locations on the face and texture representations, implicitly.

Our lip segmentation approach uses color space information and shape information in a level-set framework to detect lip boundaries.

1.2 Literature Review

Facial feature extraction from a face image has been an intensive research area. It is an appropriate application area of various image processing and pattern recognition techniques. There are many different approaches, most of them being hybrid approaches that are combinations of various methods.

Human face has an impressive special symmetry giving an ideal look. Many works in literature actively make use of this symmetry. In [1], a generalized symmetry operator is used. Method proposed in [2] makes a simple segmentation based on pixel darkness and then searches for minima pairs for eyes and mouth. A symmetry based cost function is introduced in [3] and then it is optimized to find perfect facial feature locations. [4] also uses this symmetry property to find eye locations.

Geometry-based approaches use some rule based a-priori information which are usually determined visually by people. Their most widespread use is to find the location of a special facial feature using a detailed and robust search, and then decide where to look and search for other facial features. They are especially useful for limiting the search area but they are not sufficient for a robust facial feature extraction system. Detailed geometric models are defined to indicate relative positions of features in a typical human face. Some example works incorporating strict geometric models are [4–15].

Low level image features such as corners and edges are also intensively used in the literature. Some previous works relying on simple edge information are [1,11,16]. An example work using simple corner detection is [17].

Works especially published recently using low level image features are utilizing a method “Smallest Univalued Segment Assimilating Nucleus” or its abbreviation SUSAN, proposed in [18]. This is a completely novel approach for corner detection in an intensity image, with the ability to behave like an edge detector tuning an input parameter called “geometrical threshold”. There are many works using this method to extract low level image features of a face image, for instance [19].

Shape based approaches statistically learn relative distribution of facial feature point locations, usually by introducing a point distribution model (PDM). PDM is then used as a local optimizer for a detailed localization. Shape based approaches are also used for facial feature extraction. This kind of methods are usually classified un-

der the name deformable templates. That term includes both statistical approaches and several kinds of other non-rigid template matching approaches. The most famous methods under this class of algorithms are Active Contour Models (ACM) or snakes [20] and Active Shape Models (ASM) also known as smart snakes [21, 22]. Difference of the ASM from ACM lies under the fact that ASM can only deform to fit the data in ways consistent with the training set, in other words it knows where to search for the contours. ASMs are statistical models of the shape of objects which iteratively deform to fit to an example of the object in a new image. The shape of an object is represented by a set of points (controlled by the shape model). Instead of single point extraction for each facial feature, those algorithms work in a manner of segmentation extracting many points representing a single facial feature such as eyes or the lip. In [23]; eyebrow, nostril and face are modeled using ACMs. In [24], an improvement called the jumping snake is used with manual initialization. Unlike classical snakes, it can be initialized far from the final edge and the adjustment of its parameters is easy and intuitive. In [16], multiple snakes are used synchronously after the initialization stage using color similarity map and low level image features. Recent works making use of ASMs for facial feature extraction are [25–28]. In [26], red-green-blue (RGB) color information is integrated into ASM. In [27], two separate strategies are proposed to improve the ASM. One is called “asymmetric sampling”: Information outside the contour and near the background is complex and with large variance due to illumination, background and hairstyle, so it does not carry much useful information, hence it is not suitable to use in training; on the contrary, information contained in the pixels inside the contour and hence on the face, is much more stable and deserves more attention. Therefore, instead of the symmetric strategy used in the standard ASM, they adopt asymmetric sampling. Other one is the “multi-template ASM”: It basically uses the fact that some facial features such as eyes and mouth have different appearances when they are closed and open, so a multi-template ASM is introduced to handle these kinds of different states. The work in [28] also brings some extensions to ASM: fitting more landmarks than are actually needed; selectively using two instead of one-dimensional landmark templates; adding noise to the training set; loosening up the shape model as the iterations advance; trimming covariance matrices by setting most entries to zero

and stacking two ASMs in series.

Appearance based methods use various transformations to represent the texture information learnt from a database. They are supposed to generalize to any unseen example as long as the database is large enough. Active appearance model (AAM) is first introduced in [29]. It is based on ASM however there are radical differences. A good comparison of ASM and AAM is given in [30]. ASM searches along profiles about the current model point positions to update the current estimate of the shape of the object. AAM samples the image data under the current instance and uses the difference between the model and the sample to update the appearance model parameters. ASM matches the model points to a new image using an iterative technique which is a variant of the Expectation Maximization (EM) algorithm. A search is made around the current position of each point to find a point nearby which best matches a model of the texture expected at the landmark. The parameters of the shape model controlling the point positions are then updated to move the model points closer to the points found in the image. AAM manipulates a full model of appearance, which represents both shape variation and the texture of the region covered by the model. This can be used to generate full synthetic images of modeled objects. AAM uses the difference between the current synthesized image and the target image to update its parameters. In [31], a coarse to fine approach is followed. First a coarse global AAM is used for initialization, then local detailed AAM models are used for fine localization. The whole procedure is formulated into a Maximum-A-Posteriori (MAP) framework.

Other typical examples involving deformable templates are [23, 32, 33]. In [32], statistical information is generated by estimating the distributions of the model parameters. Approximate location, scale and orientation of the head is found by repeatedly deforming the whole template at random by scaling, rotation and translation; until it matches best with the image whilst remaining a feasible head shape. In [23], the contours of some facial features such as eye and mouth are captured by a deformable template model.

There are various examples involving classical rigid template matching methods. Design of the template is the critical part of this method. A local patch is used to learn the search template from a database, using texture information. In [34],

an elliptic template is applied to the lip region, to find most probable lip position. The template's size is chosen proportional to the face ellipse size. In [35], extended templates are used which represent both local appearance and a relative positional relationship between sampling points and feature points. An extended template consists of three elements: a reference pattern at a sampling point, directional vectors from the sampling point to the feature points, and local likelihood patterns at the feature points.

To represent the texture of a facial feature in a lower dimensional subspace, there are different dimension reduction techniques used in previous works. Principal component analysis (PCA) is probably the most well known technique. Some examples using it are [11, 36–40]. A PCA subspace is trained in [36] with an initial training set. As the facial features from new test images are extracted, training set is extended by adding the results to the training set. However this may lead to a degraded PCA model due to some erroneous test results. Method proposed in [37] uses shape model to describe the outer and inner lip contour and a deformable grey level model to describe intensity values around the lip contours. It is closely related to ASM. PCA is performed on all profile examples of a training set to reduce the feature space and to obtain the principal modes of profile variation. Any profile of the training set can be approximated by a linear combination of the mean profile and the first few modes of profile variation. The profile model deforms with the contour model and therefore always represents the same object features. Kernel PCA was first introduced in [41]. It is an extension of PCA using techniques of kernel methods. Using a kernel, the originally linear operations of PCA are done in a reproducing kernel Hilbert space with a non-linear mapping. Non-linearity comes from an individually selected kernel function that enable to operate in the feature space without ever computing the coordinates of the data in that space, but rather by simply computing the inner products between the images of all pairs of data in the feature space. An example utilizing polynomial functions as kernel function in KPCA for face and facial features recognition is [42]. In [43], an improved version of PCA is introduced to use with missing data which is called Robust PCA (RPCA). In [44], RPCA is applied to capture the statistics of shape variations. Discrete cosine transformation (DCT) is another dimension reduction technique used for instance

in [7, 9, 45]. In [44], training face samples are automatically labeled by local image search using Gabor wavelet features. In [46], Gabor filters are used with 24 filters consisting of 6 different orientations and 4 different wavelengths. Linear discriminant analysis (LDA) is a useful dimension reduction tool. A good review on LDA is published in [47] and it summarizes different LDA based facial feature extraction methods. It further proposes some post processing ideas on discriminant images to vindicate the LDA that it is a suitable way of dimension reduction for facial feature extraction.

There are so many different classifiers used for facial feature extraction that it is not possible to list all of them in this thesis. Some example classifiers are: Neural networks (NN) [10], support vector machine (SVM) [42], adaboost [48], gentleboost [14], hidden Markov model (HMM) [49], Gaussian mixture model (GMM) [39], k-means clustering [50], bayesian network (BN) [40], binary decision tree (BDT) [40], self-organizing feature map (SOFM) based NNs [51].

Scale-invariant feature transform (SIFT) is an image feature extractor published in [52] and it is robust to changes in illumination, noise, and minor changes in viewpoint. An example utilizing it for facial feature extraction is [53].

Genetic algorithm (GA) is a search technique used in computing to find exact or approximate solutions to optimization and search problems. In [6], GA is used to search for facial feature locations instead of the traditional template matching. The approach in [42] used GA to select optimal feature set. GA is used in [8] to find the face region.

Besides the key points extraction from a face image, lip segmentation is the next task for detailed lip contours extraction. There are various approaches, most of them based on the snake method. In [54], a spatiotemporal Markov random field (MRF) framework is used for mouth colored pixels detection, using red hue predominant region and motion in a spatiotemporal neighborhood. In [55], a novel color transformation is proposed as a preprocessing step for lip segmentation process. In various works, Eveno et. al. proposed a new idea which is based on a parametric model composed of several cubic curves fitted on the lips. First one composed of two cubic curves and a broken line in case of a closed mouth, and a second one composed of four cubic curves in case of an open mouth. These parametric models

give a flexible and accurate final inner lip contour. This model is flexible enough to reproduce the specificities of very different lip shapes [56–61]. It is considered that the lip boundary is composed of several independent cubic polynomial curves, instead of a single closed curve. In [62, 63], classical snake algorithm is applied to extract lip contour points after key points detection. In [64], both the color information and the spatial distance are taken into account while most of the methods focus on the former. A new dissimilarity measure, which integrates the color dissimilarity and the spatial distance in terms of an elliptic shape function, is introduced, which is able to differentiate the pixels having similar color information but located in different regions. It provides good differentiation between the lip region and the non-lip region. In [65], a fuzzy clustering method is used. It is a “one object, multiple background” approach. Since the non-lip region becomes inhomogeneous in the presence of beards, multiple background clusters can produce better fitting to a rather complex background region than one single cluster. Spatial information in terms of the physical distance towards the lip center is incorporated to enhance the differentiation between the lip and the background region. In [66], a method based on a statistical model of shape (ASM) with local appearance Gaussian descriptors is used. Idea is that the response of the local descriptors can be predicted from the shape. This prediction is achieved by a non-linear neural network. In [67], a specific parametric model is defined for each deformable feature. Segmentation is done using a jumping snake model. In [68], a lip versus non-lip classification is done using SVM. Then, an energy functional is minimized using level-set methods. In [69], a novel multi-class and shape-guided fuzzy c-means (MS-FCM) clustering algorithm is introduced and used for lip segmentation. In [70], lip contours are extracted using edge information directly. A special edge detection method, wavelet multi-scale edge detection across the discrete Hartley transform is performed. In [71], level-set method is used for lip segmentation. An internal shape constraint energy is incorporated into the evolution equation. Difference between the shape constraint and current curve is used for minimization. Predetermined parametric curves are used for imposing shape constraints instead of a shape subspace built from a training set. In [72], level set framework is used for segmentation. The contribution is, instead of employing signed distance function resulting in partial differential equation for

the temporal evolution, implicit polynomial function is utilized and the temporal evolution equation is obtained using ordinary differential equation. This approach has not been tried for the lip contour segmentation purpose in that work, but the sample results with and without missing image data are promising.

1.3 Contributions

Our contributions in this thesis are:

1. A statistical facial feature extraction method which is able to automatically learn location and texture information from a training set is introduced.
2. Joint optimization problem is formulated in such a way as to be solved using gradient ascent algorithm or Newton's method.
3. We develop a lip segmentation algorithm that uses adaptive color space and shape prior information together.

1.4 Outline

In Chapter 2, we explain the facial feature extraction system in detail, which finds facial feature point locations in a face image. The face detection system, normalization of rotation, scaling, translation and side illumination effects are explained. We propose two models; independent features model and dependent locations model. We actually use both models in cooperation for our system. We describe how to realize our models for implementation. At the end, we show experimental results.

In Chapter 3, we propose a novel lip segmentation method for detailed lip contours extraction. A novel level set formulation is proposed. Then we integrate a shape prior term into our level set framework. At the end we describe how to measure the performance of segmentation algorithm and show the experimental results.

In Chapter 4, conclusion of the thesis is made and some further improvements are proposed.

Chapter 2

Facial Feature Extraction

Facial features are critical points such as lip corners, eye corners and nose tip in a human face. For the cases where the image of an object is represented by a finite set of landmark points, it is convenient to use flexible shape modeling. Usually, critical points on flexible shapes are detected and then the shape of the object can be deduced from the location of these key points. Face can be considered as a flexible object and critical points on a face can be easily identified.

In this thesis, our goal is to detect the location of facial features. We use nine facial features which we think are helpful for finding exact lip corner locations. Facial features used in this work are shown in Figure 2.1. Number of facial features to extract can be increased and determined individually by providing a training set having necessary facial features annotated.

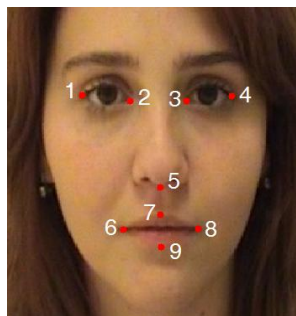
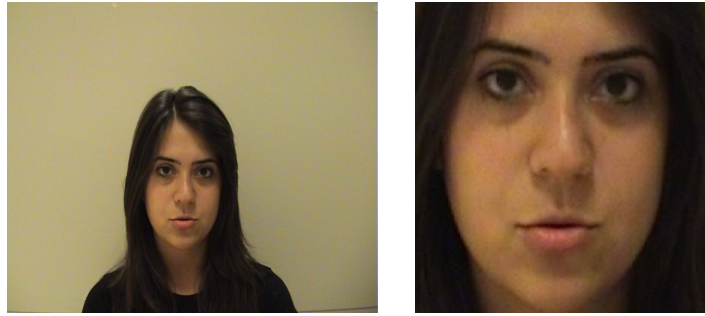


Figure 2.1: Facial features used in this work



(a) Input image

(b) Detected face image

Figure 2.2: Example face detection result



(a) Face regions, without miss

(b) Face regions, one missed miss

Figure 2.3: Obtained multiple elliptic face regions

2.1 Face Detection and Normalization

2.1.1 Face Detection

For the detection of face or separate faces in an image, the same approach in [73] is followed without further improvements. This algorithm is currently state of the art and is based on efficiently extracting Haar-like features and using those features in an Adaboost classification - feature selection framework.

An example video frame from our database and its corresponding detected face image are shown in Figure 2.2.

Two complex scenes with multiple faces are shown in Figure 2.3. Found face regions are shown using ellipses in various colors, instead of the rectangular face region in Figure 2.2b.

2.1.2 Face Normalization

Face image has to be processed after face detection to normalize various effects such as scaling, rotation, translation and side illumination. We assume that the face detection output is correct, so there is no translation in face image. Face image is resized to a fixed dimension during both training and test steps to overcome the effects of different scalings. Our system is capable of learning the texture information from face images with minor rotations.

To normalize side illumination effects, a four-region adaptive histogram equalization is applied as described in [74]. It is based on dividing the face image into 4 big rectangles and doing histogram equalization in each rectangle separately. Then each pixel is affected by 4 histogram equalization functions by its distance to each 4 regions.

2.2 Probabilistic Facial Feature Extraction

¹Every facial feature is expressed with its location and texture components. Let vector $\mathbf{l}_i = [x_i, y_i]^T$ denote the location of the i th feature in a 2D image². $\mathbf{t}_i = \mathbf{t}_i(\mathbf{l}_i)$ is the texture vector associated with it. We use $\mathbf{f}_i = [\mathbf{l}_i^T, \mathbf{t}_i^T]^T$ to denote the overall feature vector of the i th critical point on the face. The dimension of the location vector is 2, and the dimension of the texture vector is p for each facial feature. Define $\mathbf{l} = [\mathbf{l}_1^T, \mathbf{l}_2^T, \dots, \mathbf{l}_N^T]^T$, $\mathbf{t} = [\mathbf{t}_1^T, \mathbf{t}_2^T, \dots, \mathbf{t}_N^T]^T$ and $\mathbf{f} = [\mathbf{f}_1^T, \mathbf{f}_2^T, \dots, \mathbf{f}_N^T]^T$ as concatenated vectors of location, texture and combined parameters respectively.

Our goal is to find the best facial feature locations by maximizing the joint distribution of locations and textures of facial features. We define the joint probability of all features as follows:

$$P(\mathbf{f}) = P(\mathbf{t}, \mathbf{l}). \quad (2.1)$$

In this thesis, we will make different assumptions and simplifications to be able to calculate and optimize this objective function. The optimal facial feature locations can be found by solving the following optimization problem:

$$\hat{\mathbf{l}} = \operatorname{argmax}_{\mathbf{l}} P(\mathbf{t}, \mathbf{l}). \quad (2.2)$$

¹This section is based on [75].

²The location vector could be three dimensional in a 3D setup

It is not easy to solve this problem without simplifying assumptions. Hence, we introduce some of the possible assumptions in the following section.

2.2.1 Independent Features Model

We can simplify this formula by assuming independence of each feature from each other. Thus, we obtain:

$$P(\mathbf{t}, \mathbf{l}) \approx \prod_{i=1}^N P(\mathbf{t}_i, \mathbf{l}_i). \quad (2.3)$$

We can calculate the joint probability $P(\mathbf{t}_i, \mathbf{l}_i)$ by concatenating texture and location vectors; obtaining a concatenated vector \mathbf{f}_i of size $p+2$. We can then assume a parametric distribution for this combined vector and learn the parameters from training data. One choice of a parametric distribution is a Gaussian mixture model (GMM) which provides a multi-modal distribution. With this assumption, we can estimate each feature location independently, so it is suitable for parallel computation. Since

$$\hat{\mathbf{l}}_i = \operatorname{argmax}_{\mathbf{l}_i} P(\mathbf{t}_i, \mathbf{l}_i), \quad (2.4)$$

each feature point can be searched and optimized independently. The search involves extracting texture features for each location candidate (pixels) and evaluating the likelihood function for the concatenated vector at that location. The pixel coordinates which provide the highest likelihood score will be chosen as the sought feature location $\hat{\mathbf{l}}_i$. Although this assumption can yield somewhat reasonable feature points, since the dependence of locations of facial features in a typical face are ignored, the resultant points are not optimal.

2.2.2 Dependent Locations Model

Another assumption we can make is to assume that the locations of features are dependent while the textures are independent. First, we write the joint probability as follows:

$$P(\mathbf{t}, \mathbf{l}) = P(\mathbf{l})P(\mathbf{t}|\mathbf{l}). \quad (2.5)$$

Next, we approximate the second term in the equation above as:

$$P(\mathbf{t}|\mathbf{l}) \approx \prod_{i=1}^N P(\mathbf{t}_i|\mathbf{l}) \approx \prod_{i=1}^N P(\mathbf{t}_i|\mathbf{l}_i),$$

where we assume (realistically) that the textures of each facial feature component is only dependent on its own location and is independent of other locations and other textures. Since the locations are modeled jointly as $P(\mathbf{l})$, we assume dependency among locations of facial features. With this assumption, the equation of joint probability becomes:

$$P(\mathbf{t}, \mathbf{l}) = P(\mathbf{l}) \prod_{i=1}^N P(\mathbf{t}_i | \mathbf{l}_i). \quad (2.6)$$

We believe this assumption is a reasonable one since the appearance of a person's nose may not give much information about the appearance of the same person's eye or lip unless the same person is in the training data for the system. Since we assume that the training and test data of the system involve different subjects for more realistic performance assessment, we conjecture that this assumption is a valid one. The dependence of feature locations however, is a more dominant dependence and it is related to facial geometry of human beings. The location of the eyes is a good indicator for the location of the nose tip for example. Hence, we believe it is necessary to model the dependence of locations.

Finding the location \mathbf{l} that maximizes equation (2.2) will find optimal locations of each feature on the face.

2.2.3 Representing Locations and Texture

The location parameters can be represented as x and y coordinates directly.

The texture parameters are extracted from rectangular patches around facial feature points. We train subspace models for them and use p subspace coefficients as representations of textures. We use different dimension reduction techniques to build the subspaces and compare their performances at the end.

Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is an orthogonal linear transformation that transforms the data to a new coordinate system. The greatest variance by any projection of the data comes to lie on the first coordinate called the first principal component, the second greatest variance on the second coordinate, and so on.

PCA can be used for dimensionality reduction by keeping lower-order principal components and ignoring higher-order ones. Such low-order components represent

most of the data. It is possible to select individual number of lower-order principal components.

We collect the texture data of a facial feature using rectangular patches around that feature's point location, stacking them as column vectors of a data matrix. Each rectangular patch contains M pixels. Suppose \mathbf{X} is the mean subtracted data matrix. The covariance matrix of mean subtracted data is calculated by

$$\mathbf{C} = \frac{1}{N} \mathbf{X} \mathbf{X}^T. \quad (2.7)$$

An eigenvalue decomposition is applied to the covariance matrix by the formula

$$\mathbf{V}^{-1} \mathbf{C} \mathbf{V} = \mathbf{D}, \quad (2.8)$$

where \mathbf{V} is the $M \times M$ square matrix with an eigenvector in each column and \mathbf{D} is a diagonal matrix containing the corresponding eigenvalues of eigenvectors. Eigenvectors corresponding to larger eigenvalues are more important for the representation of the data. All of the eigenvectors form an orthogonal basis. Dimensionality reduction is possible ignoring the eigenvectors with lower eigenvalues. Suppose that the dimension is to be reduced to L where $1 \leq L \leq M$, then L eigenvectors with the highest corresponding eigenvalues are placed in columns of the transformation matrix \mathbf{W} of size $M \times L$. When we obtain the transformation matrix, it is possible to express the M dimensional feature vector \mathbf{x}_i as L dimensional feature vector \mathbf{y}_i by the formula

$$\mathbf{y}_i = \mathbf{W}^T \mathbf{x}_i. \quad (2.9)$$

The elements of \mathbf{y}_i are the coefficients of the orthogonal basis vectors.

Discrete Cosine Transformation (DCT)

DCT is an energy compaction technique to concentrate most of the signal information in a few low frequency components. DCT does not need a training to build the subspace as it has fixed basis vectors. DCT is an orthonormal transformation. The pixel (n_1, n_2) of DCT basis \mathbf{B}_{k_1, k_2} is calculated as:

$$\mathbf{B}_{k_1, k_2}(n_1, n_2) = \sqrt{\frac{2}{N_1}} \sqrt{\frac{2}{N_2}} \Lambda(k_1) \Lambda(k_2) \cos\left(\frac{\pi(n_1 + 0.5)k_1}{N_1}\right) \cos\left(\frac{\pi(n_2 + 0.5)k_2}{N_2}\right), \quad (2.10)$$

where $\Lambda(k) = \frac{1}{\sqrt{2}}$ when $k = 0$ and $\Lambda(k) = 1$ for other values of k . DCT coefficient $D(k_1, k_2)$ is found by the inner product $D(k_1, k_2) = \langle \mathbf{X}, \mathbf{B}_{k_1, k_2} \rangle$. We can invert this transformation and obtain the original input image \mathbf{X} as

$$\mathbf{X}_{k_1, k_2}(n_1, n_2) = \sum_{k_1=0}^{N_1} \sum_{k_2=0}^{N_2} D(k_1, k_2) \mathbf{B}_{k_1, k_2}(n_1, n_2). \quad (2.11)$$

For feature extraction from a region of interest, we use top few 2D DCT coefficients extracted from that region. We used zig-zag-scan ordering of 2D DCT coefficients for selecting the top coefficients to be used in dimension reduction.

Gabor Transformation (GT)

A real Gabor filter is defined as follows:

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{(x')^2 + \gamma^2(y')^2}{2\sigma^2}\right) \cos\left(\frac{2\pi x'}{\lambda} + \psi\right), \quad (2.12)$$

where $x' = x \cos \theta + y \sin \theta$, $y' = -x \sin \theta + y \cos \theta$, λ is the wavelength of the cosine factor, θ is the orientation of the normal to the parallel stripes of a Gabor function, ψ is the phase offset, σ is the sigma of Gaussian envelope and γ is the spatial aspect ratio specifying the ellipticity of the support of the Gabor filter.

Gabor features can be obtained by taking the inner product of an image patch with a Gabor filter. Notice that, this does not constitute an orthogonal transformation in general.

2.2.4 Modeling Locations and Texture

A multivariate Gaussian distribution is defined as follows:

$$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{N/2} |\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right), \quad (2.13)$$

where \mathbf{x} is the input vector, N is the dimension of \mathbf{x} , $\boldsymbol{\Sigma}$ is the covariance matrix and $\boldsymbol{\mu}$ is the mean vector.

For the model defined in 2.2.1, probability for each concatenated feature vector \mathbf{f}_i , $P(\mathbf{f}_i)$ is modeled using a mixture of Gaussian distributions. GMM likelihood can be written as follows:

$$P(\mathbf{f}_i) = \sum_{k=1}^K w_i^k \mathcal{N}(\mathbf{f}_i; \boldsymbol{\mu}_i^k, \boldsymbol{\Sigma}_i^k). \quad (2.14)$$

Here K is the number of mixtures, w_i^k , $\boldsymbol{\mu}_i^k$ and $\boldsymbol{\Sigma}_i^k$ are the weight, mean vector and covariance matrix of the k^{th} mixture component. \mathcal{N} indicates a Gaussian distribution with specified mean vector and covariance matrix.

For the model defined in 2.2.2, probability $P(\mathbf{t}|\mathbf{l})$ of texture parameters \mathbf{t} given location \mathbf{l} is also modeled using a GMM as in equation (2.14).

During testing, for each facial feature i , a GMM texture log-likelihood image is calculated as:

$$I_i(x, y) = \log(P(\mathbf{t}_i | \mathbf{l}_i = [x \ y]^T)). \quad (2.15)$$

Note that, to obtain $I_i(x, y)$, we extract texture features \mathbf{t}_i around each candidate pixel $\mathbf{l}_i = [x \ y]^T$ and find its log-likelihood using the GMM model for facial feature i .

Our model for $P(\mathbf{l})$ is a Gaussian model, resulting in a convex objective function. Location vector \mathbf{l} of all features is modeled as follows:

$$P(\mathbf{l}) = \mathcal{N}(\mathbf{l}; \boldsymbol{\mu}, \boldsymbol{\Sigma}). \quad (2.16)$$

Candidate regions for each feature i can be found by marginalizing the above multi-dimensional Gaussian distribution learned from data. Marginal Gaussian distribution of a feature location is thresholded and a binary ellipse region is obtained for that feature. GMM scores are calculated only inside those ellipses for faster computation.

The model parameters are learnt from training data using maximum likelihood. Expectation maximization (EM) algorithm is used to learn the parameters for the GMMs [76].

2.2.5 Algorithm

For independent features model, we calculate $P(\mathbf{f}_i)$ in equation (2.14) using GMM scores for each candidate location \mathbf{l}_i of feature i and decide the location with maximum GMM score as the location for feature i .

For dependent locations model, we propose an algorithm as follows. We obtain the log-likelihood of equation (2.6) by taking its logarithm. Because the texture of each feature is dependent on its location, we can define an objective function which

only depends on the location vector:

$$\phi(\mathbf{l}) = \log(P(\mathbf{t}, \mathbf{l})) = \log(P(\mathbf{l})) + \sum_{i=1}^N \log(P(\mathbf{t}_i | \mathbf{l}_i)). \quad (2.17)$$

Using the Gaussian model for location and GMM for texture defined in section 2.2.4, we can write the objective function ϕ as:

$$\phi(\mathbf{l}) = \frac{-\beta}{2}(\mathbf{l} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{l} - \boldsymbol{\mu}) + \sum_{i=1}^N I_i(x_i, y_i) + \text{constant}. \quad (2.18)$$

Here, $\boldsymbol{\mu}$ is the mean location vector, and $\boldsymbol{\Sigma}^{-1}$ is the precision (inverse covariance) matrix, learnt during the training. β is an adjustable coefficient. $I_i(x, y)$ is the score image of feature i defined in equation (2.15).

So the goal is to find the location vector \mathbf{l} giving the maximum value of $\phi(\mathbf{l})$:

$$\hat{\mathbf{l}} = \operatorname{argmax}_{\mathbf{l}} \phi(\mathbf{l}). \quad (2.19)$$

To find this vector, we use two different optimization algorithms and compare their performances at the end.

Gradient ascent optimization

Evolution formula for the gradient ascent optimization is given as follows:

$$\mathbf{l}^{(n)} = \mathbf{l}^{(n-1)} + k_n \nabla \phi(\mathbf{l}^{(n-1)}). \quad (2.20)$$

Here, n denotes the iteration number. We can write the location vector \mathbf{l} as:

$$\mathbf{l} = [x_1, y_1, x_2, y_2, \dots, x_N, y_N]^T. \quad (2.21)$$

Then we can find the gradient of ϕ as:

$$\nabla \phi(\mathbf{l}) = [\partial \phi / \partial x_1, \partial \phi / \partial y_1, \dots, \partial \phi / \partial y_N]^T. \quad (2.22)$$

For a single feature i :

$$\partial \phi / \partial x_i = \frac{\partial}{\partial x_i} \log P(\mathbf{l}) + \sum_{i=1}^N \frac{\partial}{\partial x_i} \log P(\mathbf{t}_i | \mathbf{l}_i), \quad (2.23)$$

and

$$\partial \phi / \partial y_i = \frac{\partial}{\partial y_i} \log P(\mathbf{l}) + \sum_{i=1}^N \frac{\partial}{\partial y_i} \log (P(\mathbf{t}_i | \mathbf{l}_i)). \quad (2.24)$$

The gradient for the location part can be calculated in closed form due to the modeled Gaussian distribution and the gradient for the texture part can be approximated from the score image using discrete gradients of the score image. Plugging in the values for the gradients, we obtain the following gradient ascent update equation for the algorithm:

$$\mathbf{l}^{(n)} = \mathbf{l}^{(n-1)} + k_n(-\beta\Sigma^{-1}(\mathbf{l}^{(n-1)} - \boldsymbol{\mu}) + \mathbf{G}), \quad (2.25)$$

where

$$\mathbf{G} = \begin{bmatrix} G_x^1(\mathbf{l}_1^{(n-1)}) \\ G_y^1(\mathbf{l}_1^{(n-1)}) \\ \dots \\ G_x^N(\mathbf{l}_N^{(n-1)}) \\ G_y^N(\mathbf{l}_N^{(n-1)}) \end{bmatrix}. \quad (2.26)$$

Here, G_x^i and G_y^i are the two-dimensional numerical gradients of $I_i(x, y)$ in x and y directions respectively. The gradients are computed only for every pixel coordinate (integers) in the image. \mathbf{G} is the collection vector of gradients of all current feature locations in the face image. k_n is the step size which can be tuned in every iteration n . Since $\mathbf{l}^{(n)}$ is a real-valued vector, we use bilinear interpolation to evaluate gradients for non-integer pixel locations. Iterations continue until the location difference between two consecutive iterations is below a stopping criterion.

Newton optimization

Evolution formula for the Newton optimization is given as follows:

$$\mathbf{l}^{(n)} = \mathbf{l}^{(n-1)} + H_\phi(\mathbf{l}^{(n-1)})^{-1} \nabla \phi(\mathbf{l}^{(n-1)}), \quad (2.27)$$

where $H_\phi(\mathbf{l}^{(n-1)})$ is the $2N \times 2N$ Hessian matrix of vector function ϕ .

Hessian matrix of the objective function is defined as:

$$H_\phi(\mathbf{l}^{(n-1)}) = -\beta\Sigma^{-1} + A, \quad (2.28)$$

where $A = \frac{\partial^2}{\partial x_i \partial y_i} \sum_{i=1}^N I_i(x_i, y_i)$.

It turns out that A is a block diagonal matrix and it is computed using second

order gradients of the score image I_i as follows:

$$\mathbf{A} = \begin{bmatrix} G_{xx}^1(\mathbf{l}_1^{n-1}) & G_{xy}^1(\mathbf{l}_1^{n-1}) & 0 & 0 & \dots \\ G_{yx}^1(\mathbf{l}_1^{n-1}) & G_{yy}^1(\mathbf{l}_1^{n-1}) & 0 & 0 & \dots \\ 0 & \dots & & & \\ \dots & & & & \\ 0 & 0 & \dots & G_{xx}^N(\mathbf{l}_N^{n-1}) & G_{xy}^N(\mathbf{l}_N^{n-1}) \\ 0 & 0 & \dots & G_{yx}^N(\mathbf{l}_N^{n-1}) & G_{yy}^N(\mathbf{l}_N^{n-1}) \end{bmatrix}. \quad (2.29)$$

The gradients G_{yx}^i and G_{xy}^i are expected to be the same. However, because of the interpolation process, they may end up having different values. So we take the average $\hat{G}_{yx}^i = \hat{G}_{xy}^i = \frac{G_{yx}^i + G_{xy}^i}{2}$ to overcome this problem.

Other optimization schemes based on the idea of improving the performance of Newton optimization method by approximation such as Gauss-Newton, Levenberg-Marquardt, efficient second order minimization (ESM) [77] are present, however the search in this optimization problem is not so complex, so the performance of the optimization is not crucial.

2.2.6 Experimental Results

Databases

We used various face video databases and their combinations for training and testing of our method. We used a selection of extracted frames from videos. Databases used for facial feature extraction are:

1. Sabanci University Turkish Audio Visual Database (SUTAV): Consists of many male and female individuals' frontal videos counting from zero to nine in Turkish. There are two tapes with same individuals and same number of videos. Individuals are in different conditions in two different tapes in terms of dressings, facial hair and illumination. No excessive head rotations, scalings and translations of the face region is present. It is a relatively challenging database because it includes different illumination conditions. We hand-marked the locations of necessary facial features in selected frames manually to make use of this database.

2. Multi Modal Verification for Teleservices and Security Applications Face Database (M2VTS - [78]): It is made up from 37 different faces and provides 5 shots for each person. These shots were taken at one week intervals or when drastic face changes occurred in the meantime. During each shot, people have been asked to count from zero to nine in their native language (most of the people are French speaking). There are videos including various head rotations. We used the videos without head rotations. We selected random frames per video and hand-marked them as in SUTAV database.
3. Technical University of Denmark Department of Informatics and Mathematical Modeling Face Database (IMMDB - [79]): The IMM Face Database comprises 240 still images of 40 different human faces, all without glasses. The gender distribution is 7 females and 33 males. The following facial structures were manually annotated using 58 landmarks: eyebrows, eyes, nose, mouth and jaw. We used only nine facial features of our interest. Each person has six different images:
 - (a) Full frontal face, neutral expression, diffuse light.
 - (b) Full frontal face, happy expression, diffuse light.
 - (c) Face rotated approximately 30 degrees to the person's right, neutral expression, diffuse light.
 - (d) Face rotated approximately 30 degrees to the person's left, neutral expression, diffuse light.
 - (e) Full frontal face, neutral expression, spot light added at the person's left side.
 - (f) Full frontal face, joker image (arbitrary expression), diffuse light.

Some facial features are unseen in some face images with much rotation. We found and put out those images.

We used some subsets and combinations of the databases explained above and made different experiments with each set:

1. Set:

- Training: All IMMDB face images except 24 images having some facial features occluded because of head rotation (216). In addition, randomly selected 4 images of 15 females and 10 males from SUTAV (100). Totally 316 frontal face images are used.
- Testing: We randomly selected 4 images of 14 females and 11 males from SUTAV (100). No identity in the test set is used in the training set.

2. Set:

- Training: In addition to the training part of Set 1, randomly selected images of 27 subjects from each 5 tape of M2VTS are used. 28 face images are eliminated because of faulty face detection results. That makes a training set of 423 face images.
- Testing: Same as the test part of Set 1.

3. Set: Only M2VTS frames are used for this set. Training and testing groups are shown below:

- Training: 10 random frames from each 5 tape of 10 subjects are selected. Those subjects are distinct from the 27 subjects used in Set 2 training. There are in total 500 face images in this training set.
- Testing: 2 random frames from each 5 tape of the same 10 subjects are selected. Those frames are distinct from the ones selected in training. There are in total 100 face images in this testing set.

Parameters

There are miscellaneous critical parameters used during our experiments. We found the optimal parameter values experimentally. We used 320×300 face images, then down-sampled them to 80×75 preserving the aspect ratio, for faster computation while improving the performance in terms of pixel errors. PCA subspaces of different dimensions are obtained for different facial features by using the texture information inside rectangular patches around facial features. Four-region adaptive histogram equalization based side-illumination normalization method described in Section 2.1.2 resulted in better facial feature extraction results. We used this method in two

Table 2.1: PCA subspace dimension and window size parameters used for facial features

Feature	PCA and DCT dimensions	Window size	Hist. eq.
1	30	17×17	2
2	30	17×17	1
3	30	21×21	2
4	30	11×11	2
5	20	11×11	2
6	50	25×25	1
7	50	27×27	2
8	50	25×25	2
9	50	39×39	2

different ways: Features having this histogram equalization method as 1; histogram equalization is applied for red, green and blue channels separately and the resulting image is converted to gray-level. For features having this histogram equalization method as 2; image is converted to gray-level and then histogram equalization is applied to the resulting image. Those training parameters are found experimentally. Optimal PCA and DCT subspace dimensions of each facial feature, window sizes used around facial feature points and side illumination normalization methods used are shown in Table 2.1. For facial features having large variability between different people, like jaw and lip; we had to train larger dimensional PCA subspaces and had to use larger windows.

Search locations for facial features are limited using separate Gaussian models trained using locations of each feature. Search areas are shown in Figure 2.4. Ellipses denote the boundaries of search regions and pentagrams denote the mean facial feature locations.

For the independent model explained in Section 2.2.1, texture coefficients and location vectors are used to build a GMM model to obtain scores. We tried two different types of GMM models when using PCA: First one is the concatenated vector choice where texture coefficients and location vectors are combined as required

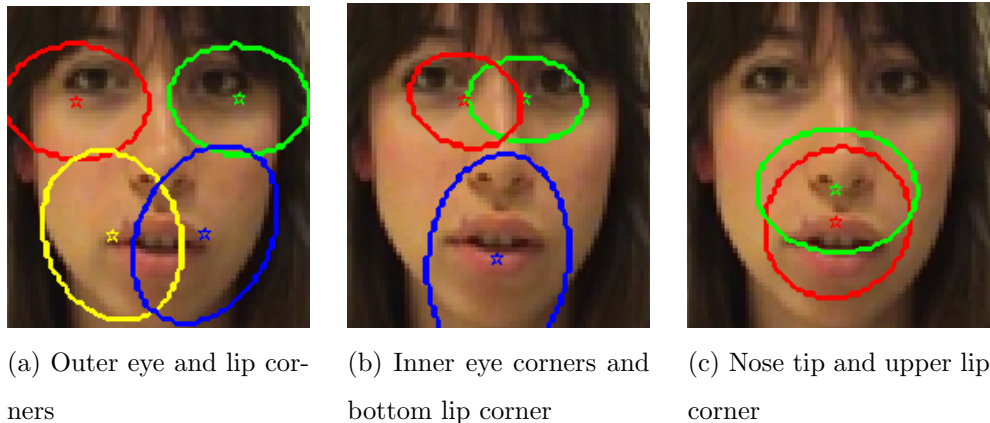


Figure 2.4: Search regions for facial features

Table 2.2: Number of mixtures used for different GMMs

GMM	Number of mixtures
GMM_{PCA}	2
GMM_{COMBI}	2
GMM_{DCT}	1
GMM_{GT}	2

in our independent model. In the second choice, we only used texture coefficients as our dependent locations model implies. For other texture feature extraction methods such as GT or DCT, we only used texture coefficients.

In total, we tried 5 different GMM models:

1. GMM built using PCA coefficients (GMM_{PCA})
2. GMM built using PCA coefficients and location parameters (GMM_{COMBI})
3. GMM built using DCT coefficients (GMM_{DCT})
4. GMM built using GT coefficients (GMM_{GT})

Two mixtures for GMMs gave the best results in most of the experiments which means that there are basically two clusters of texture data for each feature. Number of mixtures used for different GMMs are shown in Table 2.2.

For each feature, the pixel giving the highest GMM score is selected as the initial location. These locations are then used to solve the dependent locations model in

Section 2.2.2. Using the method explained in Section 2.2.5, locations and textures of the features are refined iteratively.

Results and Comparison

Pixel error of a single facial feature on a face image is the Euclidean distance between the location found for that feature and the manually labeled location. We find the mean pixel error of all facial features on a single face image. We take the mean of all such means over all test images. We also find the maximum pixel error of each facial feature over all test images, then we take the mean of such maximum errors for all facial features. This maximum value gives an idea about the performance of a method in the worst case scenario. We discuss the performance of different feature extraction algorithms, different optimization methods in different data sets in following. We also compare the performance of our method with the AAM method using the AAM implementation AAM-API [80].

Pixel errors for the independent model are shown in Table 2.3, pixel errors for the dependent model with gradient ascent optimization are shown in Table 2.4, pixel errors for the dependent model with Newton optimization are shown in Table 2.5 and pixel errors for AAM are shown in Table 2.6. PCA1 denotes the GMM where we use only PCA texture coefficients and PCA2 denotes the GMM where we use the concatenated vector of PCA texture coefficients and location parameters. All pixel errors are calculated in the 320×300 face images.

For data sets involving different people in training and testing parts, both independent and dependent models outperformed AAM. For data sets involving same people in training and testing parts like Set 3, our method gives closer results to AAM. For that case; dependent locations model gives slightly better results compared to AAM, but AAM is slightly better in terms of maximum errors.

Newton optimization usually gives better results than gradient ascent optimization in terms of maximum errors. But they give very similar results in terms of mean errors. However, Newton optimization converges in fewer iterations which makes it faster than gradient ascent optimization.

Among all texture representation methods, PCA is the best in terms of pixel errors. PCA1 is better for dependent model and PCA2 is better for independent

Table 2.3: Pixel errors for independent model

	PCA1		PCA2		DCT		GT	
Data Set	Mean	Max	Mean	Max	Mean	Max	Mean	Max
Set 1	5.805	23.672	5.538	18.879	5.981	24.616	10.978	34.517
Set 2	5.680	22.401	5.467	20.560	5.980	25.297	11.073	34.110
Set 3	4.580	51.588	4.505	51.522	5.075	54.755	11.156	69.539

Table 2.4: Pixel errors for dependent model with gradient ascent optimization

	PCA1		PCA2		DCT		GT	
Data Set	Mean	Max	Mean	Max	Mean	Max	Mean	Max
Set 1	5.251	17.756	5.211	16.399	5.774	21.077	9.407	27.737
Set 2	5.049	17.120	5.078	17.478	5.669	20.713	9.672	27.903
Set 3	4.381	51.119	4.555	51.220	4.720	51.985	9.227	64.773

model which is convenient to our formulations. DCT also gives good results. Both PCA and DCT outperformed GT.

Example facial feature extraction results are shown in Figure 2.5 and Figure 2.6.

Table 2.5: Pixel errors for dependent model with Newton optimization

	PCA1		PCA2		DCT		GT	
Data Set	Mean	Max	Mean	Max	Mean	Max	Mean	Max
Set 1	5.235	16.382	5.300	16.831	5.831	19.826	9.551	34.765
Set 2	5.021	15.688	5.124	16.495	5.656	18.574	10.029	29.632
Set 3	4.629	53.269	4.577	50.759	4.764	52.007	9.337	64.687

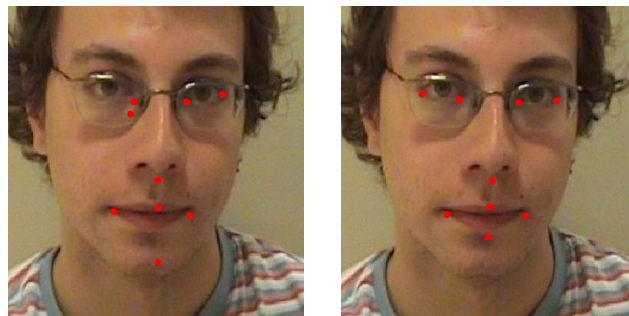
Table 2.6: Pixel errors for AAM

	AAM	
Data Set	Mean	Max
Set 1	8.126	17.845
Set 2	5.501	15.267
Set 3	4.547	50.396



(a) Independent locations (b) Dependent locations

Figure 2.5: Facial feature locations obtained using independent and dependent locations models, with a good independent model initialization



(a) Independent locations (b) Dependent locations

Figure 2.6: Facial feature locations obtained using independent and dependent location models, with an inaccurate independent model initialization

Chapter 3

Lip Segmentation

Lip segmentation can be an important part of audio-visual speech recognition (AVSR), lip-synching, modeling of talking avatars and facial feature tracking systems. In audio-visual speech recognition, it has been shown that using lip texture information is more valuable than using the lip boundary information [81,82]. However, this result may have been partly due to inaccurate boundary extraction as well, since lip segmentation performance was not independently evaluated in earlier studies. In addition, it is possible to use lip segmentation information complementary to the texture information. Lip boundary features can be utilized in addition to lip texture features in a multi-stream Hidden Markov model (HMM) framework with an appropriate weighting scheme. Thus, we conjecture it is beneficial to use lip boundary information to improve accuracy in AVSR. Once the boundary of a lip is found, one may extract geometric or algebraic features from it. These features can be used in audio-visual speech recognition systems as complementary features to audio and other visual features.

Section 3.1 follows the work of Ozgur et. al. [83] for lip segmentation using adaptive color space training. In Section 3.2, we propose the idea of imposing shape priors into our level set framework.

3.1 Lip Segmentation Using Adaptive Color Space Training

We use statistical color distributions represented by GMMs to obtain region based fields to be employed in a level-set curve evolution framework. The GMMs are first trained in a speaker independent way by taking lip and non-lip examples from multiple subjects. Then, for each test subject, we obtain initial lip and non-lip

examples using good initial guesses and adapt lip and non-lip GMMs to the subject by using the maximum a-posteriori probability (MAP) algorithm. We compare the performance of different color spaces for lip segmentation. Our level-set formulation enables fusion of different regional fields. We assess different combinations of color spaces using our approach as well.

3.1.1 Probabilistic Modeling

We would like to learn color distributions of the pixels in the lip and non-lip regions by modeling them using GMMs. The training data we used has the lip region hand-marked. We take a region of interest around the marked lip region and label the pixels as lip or non-lip within that region of interest. We randomly select a fixed number of pixels from each region in our training data. Next, we extract color-space features from each chosen pixel. We use these features to train a GMM for each region.

Let \mathbf{x} denote x and y coordinates and $c = c(\mathbf{x})$ denote the color space feature(s) associated with a pixel. Probability $P(c|R)$ is the probability of a pixel c being a lip or a non-lip pixel where R is an indicator of the region (lip (L) or non-lip (N)). This probability is modeled using a GMM distribution for c as in 2.2.4. We call these distributions generic region models in the following discussion.

3.1.2 Adaptation

During testing, we first adapt the generic region models to the subject of interest by initially choosing conservative regions in the test image making use of the extracted (or assumed) lip corners. We find two concentric ellipses that pass through the lip corners as shown in Figure 3.2a. Inside of the smaller ellipse is assumed to be a part of the lip region and outside of the outer ellipse is assumed to be a part of the non-lip region. We choose the ellipses such that for almost all subjects the assumptions are correct. We then randomly take a fixed number of samples from two assumed regions to adapt the generic models to the subject. This adaptation step yields models that are well-suited to the subject of interest. We use MAP adaptation as described in [84] with a relevance parameter ρ . After adaptation of GMMs, we obtain adapted regional models for the subject of interest.

3.1.3 Testing

For testing, we first find the region of interest using the lip corner points. For each pixel within the region of interest, we calculate a detection score based on the likelihood ratio as follows:

$$\hat{S}(\mathbf{x}) = \log p(\mathbf{c}(\mathbf{x})|N) - \log p(\mathbf{c}(\mathbf{x})|L) + \log \frac{P(N)}{P(L)}. \quad (3.1)$$

Here $P(N)$ and $P(L)$ denote the probability that a pixel belongs to non-lip and lip regions, respectively. This score is precisely the logarithm of the likelihood ratio plus a prior imbalance term. The range of the score function is $(-\infty, \infty)$. We can assume $P(N)/P(L)$ to be in the range 5-10 since there are more non-lips than lips in a typical region of interest. In order to remove regional discontinuities, we median filter this score field using a 9×9 window.

Since the logarithm of a small likelihood value tends towards $-\infty$, it may be beneficial to limit the dynamic range of the score function. To achieve this, we obtain a clipped score as follows:

$$\tilde{S}(\mathbf{x}) = \begin{cases} \hat{S}(\mathbf{x}) & \text{if } |\hat{S}(\mathbf{x})| < S_M \\ -S_M & \text{if } \hat{S}(\mathbf{x}) < -S_M \\ S_M & \text{if } \hat{S}(\mathbf{x}) > S_M \end{cases}, \quad (3.2)$$

where S_M is the maximum score allowed.

This score needs to be thresholded and we expect the pixel \mathbf{x} to belong to the lip region if the score is less than the threshold and vice versa. This threshold can be varied to adjust precision-recall trade-off (or ROC curve). However, we would like to choose a single optimal threshold value in this work. In order to choose a single best threshold value, we make use of two ellipses that go through lip and non-lip regions as shown in Figure 3.2b.

We calculate the means μ of the clipped score field $\tilde{S}(\mathbf{x})$ values on the boundary pixels of each ellipse. We then choose the single best threshold value to be a value in between these two means as follows:

$$t_{opt} = k\mu_{lip} + (1 - k)\mu_{nonlip}. \quad (3.3)$$

One might think of choosing the threshold as the middle point between two means ($k = 0.5$) but, we experimentally found that using a k value larger than 0.5 worked better.

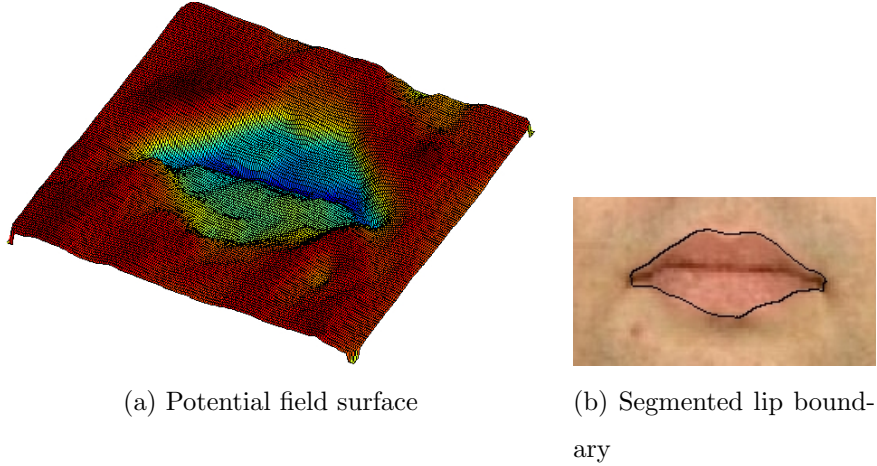


Figure 3.1: Potential field surface and segmented lip boundary

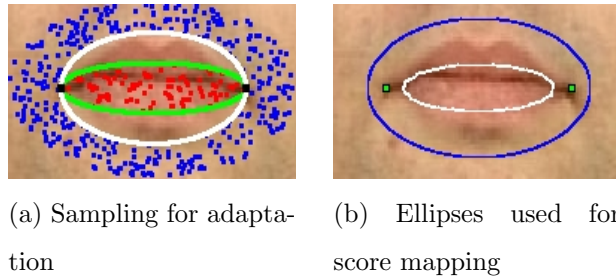


Figure 3.2: Sampling and score mapping

Our level-set formulation requires a score field which is between -1 and +1 and we would like to have a negative value within the lip region and a positive value within the non-lip region ideally. So, we need to map the score value to the range $(-1, 1)$. For this purpose, we *linearly* map the score value $\tilde{S}(\mathbf{x}) - t_{opt}$ to the desired range by dividing the score by the maximum absolute score Z :

$$S(\mathbf{x}) = (\tilde{S}(\mathbf{x}) - t_{opt})/Z. \quad (3.4)$$

We also experimented with some more sophisticated score mapping techniques with no improvement in results.

Figure 3.1a shows an example potential field surface $S(\mathbf{x})$ and the final lip contour which is governed by it.

3.1.4 Curve Evolution

We consider a level set formulation [85] based on both region and image gradients information. For this purpose, we construct color based potential fields to be used in level set equation to drive the interface to the boundaries of a lip and stop there

by the help of a stopping function which uses image gradients. The initial curve is chosen to be an ellipse which passes through the extracted lip corners around the mouth. The evolution of the level set function is given by:

$$\Phi_t + F|\nabla\Phi| = 0, \quad (3.5)$$

where the speed function F on a pixel \mathbf{x} is designed as:

$$F(\mathbf{x}) = -g(\mathbf{x})(\epsilon\kappa + \sum_{i=1}^n \alpha_i S_i(\mathbf{x})), \quad (3.6)$$

where g , κ and S denote stopping function, curvature and the potential field constructed from a color space, respectively. The coefficients ϵ and α_i are positive scalars. The number of color spaces used is given by n . We have used up to three color spaces together in this paper. The potential field $S_i(\mathbf{x})$ is computed using equation (3.4) for the i^{th} color feature.

The stopping function is designed in terms of the gradient of the Gaussian smoothed image as follows,

$$g(\mathbf{x}) = \frac{1}{1 + |\nabla((G * I)(\mathbf{x}))|^p}, \quad p \geq 1. \quad (3.7)$$

Reinitialization

Signed distance function (SDF) Φ represents the distance of pixels from the zero level set. For the purpose of segmentation, only the zero level set is meaningful. Distance of other level sets may blow up during the evolutions. This situation may lead to incorrect calculations for the evolution. To overcome this problem, a periodic reinitialization of SDF is done: We stop the evolution calculation periodically in time and discretize the zero level set holding only the isocontour $\Phi = 0$. We then measure the signed distances of other pixels to the zero level set isocontour.

3.1.5 Color Spaces

In our experiments, we used 4 types of color space: $\{RGB\}$, $\{\frac{R}{R+G}\}$, $\{Hue\}$ and $\{rg\}$, to train GMMs and to construct potential fields. We define $r = R/(R+G+B)$ and $g = G/(R+G+B)$ as red and green ratios independent of illumination, and $\{rg\}$

denotes this normalized chromatic space. The $\{\frac{R}{R+G}, Hue^*\}$ and $\{\frac{R}{R+G}, Hue, rg\}$ combined color spaces are also employed. In the first combined color space, $\{Hue^*\}$ represents hue image itself, used as potential field, after mapping to the range $(-1,1)$.

3.2 Addition of Shape Priors

In Section 3.3, it is shown that the method proposed in Section 3.1 achieves satisfying results on its own. Human lip shape has discriminative characteristics of shape, and it is perceptual to integrate this shape prior information into our evolution formulation.

Heaviside function $H\Phi$ is defined as follows:

$$\begin{aligned}\Phi(x, y) \geq 0 &\Rightarrow H\Phi(x, y) = 1, \\ \Phi(x, y) < 0 &\Rightarrow H\Phi(x, y) = 0,\end{aligned}\tag{3.8}$$

where Φ is the signed distance function (SDF).

3.2.1 Shape Priors Using Linear Subspace

For shape subspace, we prefer to build a linear subspace as in [86] considering computational issues together with the urge of palatable segmentation outputs.

Training

Suppose we have a training set $\tau = \{I_1, I_2, \dots, I_N\}$ of N binary images as $m \times n$ matrices. I_i 's represent the possible shapes of objects of interest. They have the value of 1 inside the object and 0 outside. Extraction of the region of interest (i.e. lip region) from those binary images is straightforward, we just cut the part of the image from top and left starting with 1's and to bottom and right ending with 1's. We resize this ROI to a fixed dimension to get rid of scaling differences. An example binary image and the corresponding ROI can be seen in Figure 3.3. We perform PCA directly on those ROIs. First the mean shape is found as $\mu = \frac{1}{N} \sum_{i=1}^N I_i$. Subtract from each I_i this mean shape to create a mean-offset map \tilde{I}_i . These \tilde{I}_i 's are written as column vectors and collected in a matrix $\mathbf{M} = [\tilde{I}_1^c, \tilde{I}_1^c, \dots, \tilde{I}_1^c]$. Covariance matrix is then $\mathbf{C} = \frac{1}{N} \mathbf{M} \mathbf{M}^T$. \mathbf{C} is decomposed using singular value decomposition as $\mathbf{C} = \mathbf{U} \mathbf{\Sigma} \mathbf{U}^T$ where \mathbf{U} is a matrix whose column vectors represent the set of

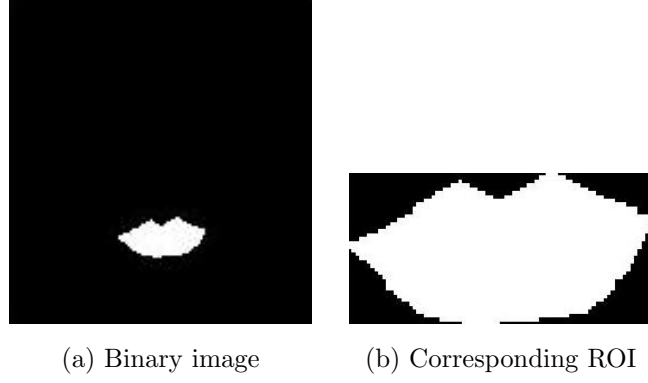


Figure 3.3: Binary image and corresponding ROI

orthogonal modes of shape variation (principal components) and Σ is a diagonal matrix of corresponding eigenvalues. Each column \mathbf{u}_i of \mathbf{U} can be rearranged as the original lip region image inverting the stacking process.

Let S be a given binary map, representing an arbitrary Heaviside function of lip region. The coordinates α^k of the projection of S onto the first k components of the space of shapes can be computed as

$$\alpha^k = \mathbf{U}_k^T (S^c - \mu^c), \quad (3.9)$$

where \mathbf{U}_k is the dimension reduction of \mathbf{U} to its first k columns, S^c and μ^c are column vectors obtained by stacking S and μ . The projection $P^k(S)$ of S is then obtained by:

$$P^k(S) = \sum_{i=1}^k \alpha_i^k \mathbf{u}_i + \mu. \quad (3.10)$$

Curve Evolution

The shape energy is defined as follows:

$$E_{shape}(\Phi) = \|H\Phi - P^k(H\Phi)\|^2. \quad (3.11)$$

Here, $P^k(H\Phi)$ is the projection of Heaviside function onto the shape subspace. Our aim is to minimize the shape energy.

For the implementation of the Heaviside function, following regularization is used:

$$H_\epsilon \phi(x, y) := \left(\frac{1}{2} + \frac{1}{\pi} \arctan \frac{\phi(x, y)}{\epsilon} \right), \quad (3.12)$$

and

$$\delta_\epsilon \phi(x, y) := \frac{1}{\pi} \left(\frac{\epsilon}{\phi^2(x, y) + \epsilon^2} \right), \quad (3.13)$$

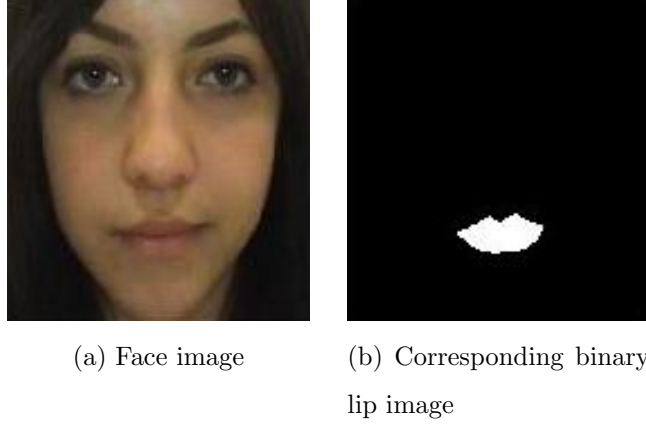


Figure 3.4: Example face image and its corresponding binary lip image

where ϵ is a parameter such that as $\epsilon \rightarrow 0$, $H_\epsilon \rightarrow H$ and $\delta_\epsilon \rightarrow \delta$ with $\delta = H'$.

The new curve evolution term is the following:

$$\Phi_t = (\beta_1(-F|\nabla\phi|) + \beta_2(P^k(H_\epsilon\phi) - H_\epsilon\phi))2\delta_\epsilon\phi, \quad (3.14)$$

where β_1 is the image term weight and β_2 is the data term weight.

3.3 Experimental Results

3.3.1 Database

For testing, we used 100 face images from our own audio-visual database SUTAV. For each test image, we prepared a hand-marked binary face image having 1 inside the lip and 0 outside. Comparing the binary ground truth lip image with the segmented lip region supplies us the segmentation errors. An example face image and its corresponding binary lip image is shown in Figure 3.4.

3.3.2 Performance Metric

In order to assess the segmentation performance we used the following *precision* (p) and *recall* (r) metrics,

$$p = \frac{t_p}{t_p + f_p}, \quad r = \frac{t_p}{t_p + f_n}, \quad (3.15)$$

where t_p , f_p and f_n denote the true positives, the false positives and the false negatives with respect to ground truth binary image of the lip, respectively. The closer p and r are to 1, the better the segmentation. Segmented lip region is equal

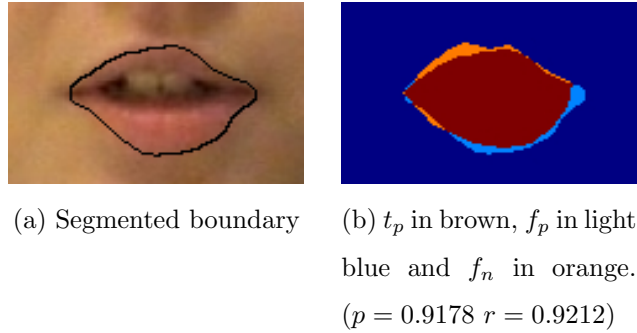


Figure 3.5: Segmented boundary and its precision-recall image

to $t_p + f_p$. Figure 3.5 shows a segmented lip and its t_p in brown, f_p in light blue and f_n in orange colors.

3.3.3 Results

To make a comparison of the performance of adaptive color space training image term, we also made the same experiments using the simple gray-level region based image term in [86] with both shape priors and without shape priors. That region based image term basically uses the difference between mean intensity inside the bounded object and mean intensity outside of the bounded object.

Precision and recall results of our method for different color spaces with and without shape priors, and that of conventional gray-level region based method are shown in Table 3.1. For adaptive color space segmentation, we take the mean precision and recall results over all color spaces. At the end, we compare all methods with respect to their mean precision and recall results.

Example adaptive color space segmentation results are shown in Table 3.2. Left column shows the results without shape priors, right column shows the results using shape priors. Each row represents a different color space, from top to bottom; RGB, R/(R+G), HUE, rg, Combination 1 and Combination 2. Region based gray-level segmentation results for the same region of interest are shown in Table 3.3. Again left column shows the result without shape priors, right column shows the result using shape priors.

Table 3.1: Precision and recall results

	Precision	Recall	Mean
Region based gray-level without shape priors	0.6151	0.9139	0.7645
Adaptive color space without shape priors:	—	—	—
RGB	0.7316	0.7970	0.7643
R/(R+G)	0.8380	0.8574	0.8477
HUE	0.7955	0.8611	0.8283
rg	0.8521	0.7866	0.8194
Combination 1	0.8444	0.8389	0.8417
Combination 2	0.8952	0.6487	0.7720
Mean of all color spaces	0.8261	0.7983	0.8122
Region based gray-level with shape priors	0.5410	0.9947	0.7679
Adaptive color space with shape priors:	—	—	—
RGB	0.7075	0.9405	0.8240
R/(R+G)	0.8095	0.9315	0.8705
HUE	0.7576	0.9469	0.8522
rg	0.7544	0.9600	0.8572
Combination 1	0.8282	0.9086	0.8684
Combination 2	0.8813	0.7764	0.8288
Mean of all color spaces	0.7898	0.9106	0.8502

Table 3.2: Adaptive color space segmentation results













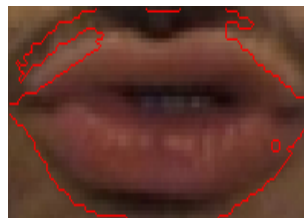
	Without using shape priors	Using shape priors
RGB		
R/(R+G)		
Hue		
rg		
Combination 1		
Combination 2		

Table 3.3: Region based gray-level segmentation results

Without using shape priors



Using shape priors



Chapter 4

Conclusions and Future Work

4.1 Conclusions

It is experimentally shown that our facial feature extraction method using both the independent locations model and the dependent locations model outperforms AAM for the same experimental setup in terms of pixel errors. Approaches like AAM and ASM are widely used for the purpose of facial feature extraction. These are very popular methods, however they give favorable results only if the training and test sets consist of a single person. They can not perform as well for person-independent general models. AAM uses subspaces of location and texture parameters which are learned from training data. However, this learning is not probabilistic and every point in the subspace is considered equally likely. This is highly unrealistic since we believe some configurations in the subspace may have to be favored as compared to other configurations. An advantage of AAM is that it takes into account global pose variations. Our algorithm is modeling the probability distributions of facial feature locations arising from inter-subject differences when there are no major global pose variations. It is critical for our algorithm that it takes as the input, the result of a good face detector.

Dependent locations model gives better results than independent locations model as expected. Newton optimization usually gives relatively better results than gradient ascent optimization as it additionally takes the second order gradients of the objective function into account.

Among texture feature extraction methods, DCT and PCA give good results. In our dependent locations formulation, GMMs are supposed to be trained using only texture coefficients. When we use only PCA texture coefficients for training

the GMM, it gives better results as it is appropriate to our formulation. We tried using combined PCA texture coefficients and location parameters, this combination achieved better results for independent features model but did not provide much improvement. We trained the GMMs only with texture coefficients for other feature extraction techniques.

As it is reasonable, using a data set which is constituted by a more homogenous selection, that is the selection from same kinds of databases, provides better results. This way our method can more easily learn the illumination conditions and personal appearances of a special database.

Our lip segmentation algorithm works well both using shape priors and without using shape priors, it outperforms conventional gray-level region based segmentation in all cases. It is shown that $R/(R+G)$ is the best color space when the object of interest is the lip. Using the gray-level region based segmentation without shape priors can give visually unpleasant results when facial hair is present in the region of interest. Imposing shape priors, improves the results visually, but it can only make little improvement in terms of precision and recall. Imposing shape priors to adaptive color space lip segmentation algorithm also improves our method. Shape priors are especially useful when there are local discontinuities other than lip boundary in a noisy lip region image.

4.2 Future Work

We were able to get promising facial feature extraction results from independent and dependent locations assumptions offered in this work. Dependent locations model improves the independent one. It is in our plans to find better texture parameters. Using global-pose variation compensation is expected to improve our approach. Using color information in addition to gray-level intensities can also improve facial feature extraction results. It is possible to fine tune special parameters experimentally but it would not be fair when comparing our method with general purpose approaches like AAM. Our method is better utilized for point extraction from face image but it is possible to use it for other purposes, such as medical imaging. Other distributions for locations and texture can be used.

For the lip segmentation part, an obvious improvement can be done using other

color spaces, such as non-linear ones. An implicit polynomial function may be employed instead of the current signed distance function.

We currently use a linear shape space built with the help of PCA. However, as stated in [87], the space of signed distance functions is a nonlinear manifold and is not closed under linear operations. We can improve upon this approach, for instance by extending a Parzen density estimator to the space of shapes.

One can develop an adaptive weighting between image term and shape term for our segmentation framework. The weight of the image term may decrease as the level-set iterate, while the weight of the shape term increases. So that, a faster convergence is expected in previous iterations. In later iterations, shape priors will gain more importance and local improvements in lip boundary may give more pleasant results.

Another possible improvement is to consider inner and outer lip boundaries as separate objects and employ a multiple object segmentation for inner and outer bounds of the lip. For this, we need a binary training set labeled both for inner and outer lip boundaries to build the necessary shape spaces.

Bibliography

- [1] D. Reisfeld and Y. Yeshurun, “Robust detection of facial features by generalized symmetry,” *ICPR*, vol. A, pp. 117–120, 1992.
- [2] K. Sobottka and I. Pitas, “Face localization and facial feature extraction based on shape and color information,” *ICIP*, vol. C, pp. 483–486, 1996.
- [3] E. Saber and A. M. Tekalp, “Frontal-view face detection and facial feature extraction using color, shape, and symmetry based cost functions,” *Pattern Recogn. Lett.*, vol. 19, no. 8, pp. 669–680, 1998.
- [4] J.-S. Oh, D. W. Kim, J. T. Kim, Y.-I. Yoon, and J.-S. Choi, “Facial component detection for efficient facial characteristic point extraction,” in *ICIAR*, 2005, pp. 1125–1132.
- [5] S. Jeng, H. Liao, Y. Liu, and M. Chern, “An efficient approach for facial feature detection using geometrical face model,” *ICPR*, vol. C, pp. 426–430, 1998.
- [6] C. Lin and J. Wu, “Automatic facial feature extraction by genetic algorithms,” vol. 8, no. 6, pp. 834–845, June 1999.
- [7] M. Zobel, A. Gebhard, D. Paulus, J. Denzler, and H. Niemann, “Robust facial feature localization by coupled features,” in *4th International Conference on Automatic Face and Gesture Recognition*, 2000, pp. 28–30.
- [8] Y. H. S. H. Hyoung Woo Lee, SeKee Kil, “Automatic face and facial features detection,” *IEEE International Symposium on Industrial Electronics*, vol. 1, pp. 254–259, 2001.
- [9] C. Sanderson and K. Paliwal, “Fast feature extraction method for robust face verification,” *Electronics Letters*, vol. 38, no. 25, pp. 1648–1650, Dec 2002.

- [10] S. Phimoltares, C. Lursinsap, and K. Chamnongthai, “Locating essential facial features using neural visual model,” in *Machine Learning and Cybernetics, 2002. Proceedings. 2002 International Conference on*, vol. 4, Nov. 2002, pp. 1914–1919 vol.4.
- [11] L. Zhi-fang, Y. Zhi-sheng, A. Jain, and W. Yun-qiong, “Face detection and facial feature extraction in color image,” in *Computational Intelligence and Multimedia Applications, 2003. ICCIMA 2003. Proceedings. Fifth International Conference on*, Sept. 2003, pp. 126–130.
- [12] A. Gunduz and H. Krim, “Facial feature extraction using topological methods,” in *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, vol. 1, Sept. 2003, pp. I-673–6 vol.1.
- [13] V. Perlibakas, “Automatic detection of face features and exact face contour,” *Pattern Recogn. Lett.*, vol. 24, no. 16, pp. 2977–2985, 2003.
- [14] D. Vukadinovic and M. Pantic, “Fully automatic facial feature point detection using gabor feature based boosted classifiers,” in *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, vol. 2, Oct. 2005, pp. 1692–1698 Vol. 2.
- [15] C. Boehnen and T. Russ, “A fast multi-modal approach to facial feature detection,” in *Application of Computer Vision, 2005. WACV/MOTIONS '05 Volume 1. Seventh IEEE Workshops on*, vol. 1, Jan. 2005, pp. 135–142.
- [16] H. Wu, T. Yokoyama, D. Pramadihanto, and M. Yachida, “Face and facial feature extraction from color image,” in *FG '96: Proceedings of the 2nd International Conference on Automatic Face and Gesture Recognition (FG '96)*. Washington, DC, USA: IEEE Computer Society, 1996, p. 345.
- [17] R. Elham Bagherian and N. I. Udzir, “Extract of facial feature point,” in *IJC-SNS*, 2009.
- [18] S. M. Smith, “A new class of corner finder,” in *Proc. 3rd British Machine Vision Conference*, 1992, pp. 139–148.

- [19] C. D. Hua Gu, Guangda Su, “Feature points extraction from faces,” in *Image and Vision Computing NZ*, 2003.
- [20] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active contour models,” *International Journal of Computer Vision*, vol. V1, no. 4, pp. 321–331, January 1988. [Online]. Available: <http://dx.doi.org/10.1007/BF00133570>
- [21] T. Cootes and C.J.Taylor, “Active shape models - smart snakes,” in *In British Machine Vision Conference*. Springer-Verlag, 1992, pp. 266–275.
- [22] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, “Active shape models—their training and application,” *Comput. Vis. Image Underst.*, vol. 61, no. 1, pp. 38–59, 1995.
- [23] C.-L. Huang and C. W. Chen, “Human facial feature extraction for face interpretation and recognition,” *Pattern Recognition*, vol. 25, no. 12, pp. 1435–1444, 1992.
- [24] H.E.Cetingul, “Discrimination analysis of lip motion features for multimodal speaker identification and speech-reading,” Master’s thesis, Koc University, Electrical and Computer Engineering, July 2005.
- [25] V. Zanella, H. Vargas, and L. Rosas, “Automatic facial features localization,” in *Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on*, vol. 1, Nov. 2005, pp. 290–264.
- [26] M. H. Mahoor and M. Abdel-Mottaleb, “Facial features extraction in color images using enhanced active shape model,” in *FGR '06: Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 144–148.
- [27] Y. Li, J. H. Lai, and P. C. Yuen, “Multi-template asm method for feature points detection of facial image with diverse expressions,” in *FGR '06: Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 435–440.

- [28] S. Milborrow and F. Nicolls, “Locating facial features with an extended active shape model,” in *ECCV '08: Proceedings of the 10th European Conference on Computer Vision*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 504–513.
- [29] T. F. Cootes, G. J. Edwards, and C. J. Taylor, “Active appearance models,” *Lecture Notes in Computer Science*, vol. 1407, pp. 484–??, 1998.
- [30] T. F. Cootes, G. Edwards, and C. Taylor, “Comparing active shape models with active appearance models,” in *in Proc. British Machine Vision Conf.* BMVA Press, 1999, pp. 173–182.
- [31] F. Tang, J. Wang, H. Tao, and Q. Peng, “Probabilistic hierarchical face model for feature localization,” in *WACV '07: Proceedings of the Eighth IEEE Workshop on Applications of Computer Vision*. Washington, DC, USA: IEEE Computer Society, 2007, p. 53.
- [32] A. Bennett and I. Craw, “Finding image features using deformable templates and detailed prior statistical knowledge,” in *In British Machine Vision Conference*. Springer Verlag, 1991, pp. 233–239.
- [33] R. Niese, A. Al-Hamadi, and B. Michaelis, “A stereo and color-based method for face pose estimation and facial feature extraction,” in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 1, 0-0 2006, pp. 299–302.
- [34] V. Vezhnevets, “Face and facial feature tracking for natural human-computer interface,” 2002.
- [35] M. Y. Tatsuo Kozakaya, Tomoyuki Shibata and O. Yamaguchi, “Facial feature localization using weighted vector concentration approach,” in *IEEE Automatic Face and Gesture Recognition*, 2008.
- [36] H. Demirel, T. J. Clarke, and P. Y. K. Cheung, “Adaptive automatic facial feature segmentation,” in *FG '96: Proceedings of the 2nd International Conference on Automatic Face and Gesture Recognition (FG '96)*. Washington, DC, USA: IEEE Computer Society, 1996, p. 277.

- [37] J. Lttin, N. A. Thacker, and S. W. Beet, “Speaker identification by lipreading,” in *In : Internat. Conf. Spoken Language Processing*, 1996, pp. 62–65.
- [38] F. Bourel, C. Chibelushi, and A. Low, “Robust facial feature tracking,” in *Proc. 11th British Machine Vision Conference*, Bristol, England, 2000, pp. 232–241.
- [39] K. Kumatani, H. K. Ekenel, H. Gao, R. Stiefelhagen, and A. Ercil, “Multi-stream gaussian mixture model based facial feature localization,” in *Signal Processing, Communication and Applications Conference, 2008. SIU 2008. IEEE 16th*, April 2008, pp. 1–4.
- [40] Z. Riaz, C. Mayer, M. Beetz, and B. Radig, “Model based analysis of face images for facial feature extraction,” in *CAIP*. Springer, 2009.
- [41] B. Schölkopf, A. Smola, and K.-R. Müller, “Nonlinear component analysis as a kernel eigenvalue problem,” *Neural Comput.*, vol. 10, no. 5, pp. 1299–1319, 1998.
- [42] Z. Yankun and L. Chongqing, “Face recognition using kernel principal component analysis and genetic algorithms,” in *Neural Networks for Signal Processing, 2002. Proceedings of the 2002 12th IEEE Workshop on*, 2002, pp. 337–343.
- [43] H. Shum, K. Ikeuchi, and R. Reddy, “Principal component analysis with missing data and its application to polyhedral object modeling,” pp. 3–39, 2001.
- [44] Z. Gui and C. Zhang, “Robust active shape model construction and fitting for facial feature localization,” in *AVBPA*, 2005, pp. 1029–1038.
- [45] H. E. Çetingül, E. Erzin, Y. Yemez, and A. M. Tekalp, “Multimodal speaker/speech recognition using lip motion, lip texture and audio,” *Signal Process.*, vol. 86, no. 12, pp. 3549–3558, 2006.
- [46] M. C. Serhan Cosar and A. Ercil, “Graphical model based facial feature point tracking in a vehicle environment,” in *Biennial on DSP for in-Vehicle and Mobile Systems*, Istanbul, Turkey, June 2007.
- [47] K. Wang, W. Zuo, and D. Zhang, “Post-processing on lda’s discriminant vectors for facial feature extraction,” in *AVBPA*, 2005, pp. 346–354.

- [48] Z. Jiao, W. Zhang, and R. Tong, “A method for accurate localization of facial features,” in *Education Technology and Computer Science, 2009. ETCS '09. First International Workshop on*, vol. 3, March 2009, pp. 261–264.
- [49] P.-H. Lee, Y.-W. Wang, J. Hsu, M.-H. Yang, and Y.-P. Hung, “Robust facial feature extraction using embedded hidden markov model for face recognition under large pose variation,” in *MVA, 2007*, pp. 392–395.
- [50] N. Gourier, D. Hall, and J. Crowley, “Facial features detection robust to pose, illumination and identity,” in *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, vol. 1, 0-0 2004, pp. 617–622 vol.1.
- [51] B. Takacs and H. Wechsler, “Locating facial features using sofm,” in *Proceedings of the 12th IAPR International Conference on Pattern Recognition (Cat. No.94CH3440-5)*, vol. 2, Inst. for Comput. Sci., George Mason Univ., Fairfax, VA, USA. Los Alamitos, CA, USA: IEEE Comput. Soc. Press, 1994, pp. 55–60.
- [52] D. Lowe, “Object recognition from local scale-invariant features,” 1999, pp. 1150–1157.
- [53] Y. Meng and D. B. Tiddeman, “Implementing the scale invariant feature transform(sift) method,” 2008.
- [54] M. Lievin and F. Luthon, “Unsupervised lip segmentation under natural conditions,” in *Acoustics, Speech, and Signal Processing, 1999. ICASSP '99. Proceedings., 1999 IEEE International Conference on*, vol. 6, Mar 1999, pp. 3065–3068 vol.6.
- [55] N. Eveno, A. Caplier, and P. Coulon, “New color transformation for lips segmentation,” in *Multimedia Signal Processing, 2001 IEEE Fourth Workshop on*, 2001, pp. 3–8.
- [56] ———, “A parametric model for realistic lip segmentation,” in *Control, Automation, Robotics and Vision, 2002. ICARCV 2002. 7th International Conference on*, vol. 3, Dec. 2002, pp. 1426–1431.

- [57] ———, “Jumping snakes and parametric model for lip segmentation,” in *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, vol. 2, Sept. 2003.
- [58] ———, “Accurate and quasi-automatic lip tracking,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 14, no. 5, pp. 706–715, May 2004.
- [59] ———, “Key points based segmentation of lips,” in *Multimedia and Expo, 2002. ICME '02. Proceedings. 2002 IEEE International Conference on*, vol. 2, 2002, pp. 125–128.
- [60] C. Bouvier, P.-Y. Coulon, and X. Maldague, “Unsupervised lips segmentation based on roi optimisation and parametric model,” in *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, vol. 4, 16 2007-Oct. 19 2007, pp. IV –301–IV –304.
- [61] S. Stillittano and A. Caplier, “Inner lip segmentation by combining active contours and parametric models,” in *VISAPP (1)*, 2008, pp. 297–304.
- [62] P. Delmas, N. Eveno, and M. Lievin, “Towards robust lip tracking,” in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, vol. 2, 2002, pp. 528–531 vol.2.
- [63] S. Werda, W. Mahdi, A. B. Hamadou, S. Werda, W. Mahdi, and A. B. Hamadou, “Lip localization and viseme classification for visual speech recognition,” *International Journal of Computing and Information Sciences*, 2008.
- [64] S.-H. Leung, S.-L. Wang, and W.-H. Lau, “Lip image segmentation using fuzzy clustering incorporating an elliptic shape function,” *Image Processing, IEEE Transactions on*, vol. 13, no. 1, pp. 51–62, Jan. 2004.
- [65] S. Wang, W. Lau, S. Leung, and A. Liew, “Lip segmentation with the presence of beards,” in *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, vol. 3, May 2004, pp. iii–529–32 vol.3.

- [66] P. Gacon, P.-Y. Coulon, and G. Bailly, “Non-linear active model for mouth inner and outer contours detection,” in *EUSIPCO*, 2005.
- [67] Z. Hammal, N. Eveno, A. Caplier, and P. Coulon, “Parametric models for facial features segmentation,” *Signal Process.*, vol. 86, no. 2, pp. 399–413, 2006.
- [68] A. Khan, W. Christmas, and J. Kittler, “Lip contour segmentation using kernel methods and level sets,” 2007, pp. II: 86–95.
- [69] S.-L. Wang, W.-H. Lau, A. W.-C. Liew, and S.-H. Leung, “Robust lip region segmentation for lip images with complex background,” *Pattern Recogn.*, vol. 40, no. 12, pp. 3481–3491, 2007.
- [70] Y. Guan, “Automatic extraction of lips based on multi-scale wavelet edge detection,” *Computer Vision, IET*, vol. 2, no. 1, pp. 23–33, March 2008.
- [71] J. S. Chang, E. Y. Kim, and S. H. Park, “Lip contour extraction using level set curve evolution with shape constraint,” in *HCI (3)*, 2007, pp. 583–588.
- [72] E. Ozgur, M. Unel, H. Erdogan, and A. Ercil, “Evolving implicit polynomial interfaces,” in *BMVC 2008*, 2008.
- [73] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, 2001, pp. I-511–I-518 vol.1.
- [74] U. Meier, R. Stiefelhagen, J. Yang, and A. Waibel, “Towards unrestricted lip reading,” *International Journal of Pattern Recognition and Artificial Intelligence*, 1999.
- [75] M. B. Yilmaz, H. Erdogan, and M. Unel, “Statistical facial feature extraction using joint distribution of location and texture information,” in *Signal Processing, Communication and Applications Conference, 2009. SIU 2009.*, 2009.
- [76] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the em algorithm,” *Journal of the Royal Statistical Society*, vol. 39, pp. 1–38, 1977.

- [77] E. Malis, “Improving vision-based control using efficient second-order minimization techniques,” in *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, vol. 2, 26-May 1, 2004, pp. 1843–1848 Vol.2.
- [78] M2VTS Database. [Online]. Available: <http://www.tele.ucl.ac.be/PROJECTS/M2VTS/m2fdb.html>
- [79] M. B. Stegmann. IMM Face Database. [Online]. Available: <http://www2.imm.dtu.dk/aam/datasets/datasets.html>
- [80] The AAM-API. [Online]. Available: <http://www2.imm.dtu.dk/aam/aamapi/>
- [81] C. Neti, G. Potamianos, J. Luetin, I. Matthews, H. Glotin, and D. Vergyri, “Large-vocabulary audio-visual speech recognition: A summary of the johns hopkins summer 2000 workshop,” in *Proc. Works. Multimedia Signal Process. (MMSP)*, Cannes, France, 2001, pp. 619–624.
- [82] G. Potamianos, C. Neti, G. Gravier, A. Garg, and A. Senior, “Recent advances in the automatic recognition of audio-visual speech,” in *Proceedings of the IEEE*, vol. 91, no. 9, 2003, pp. 1306–1326.
- [83] E. Ozgur, M. B. Yilmaz, H. Karabalkan, H. Erdogan, and M. Unel, “Lip segmentation using adaptive color space training,” in *International Conference on Auditory-Visual Speech Processing 2008*, Moreton Island, Australia, September 2008.
- [84] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, “Speaker verification using adapted gaussian mixture models,” in *Digital Signal Processing*, vol. 10, January 2000.
- [85] J. A. Sethian, *Level set methods: Evolving interfaces in geometry, fluid mechanics, computer vision, and materials science*, ser. Cambridge monographs on applied and computational mathematics. Cambridge, U.K.: Cambridge University Press, 1996, no. 3, 218 pages.
- [86] S. Dambreville, Y. Rathi, and A. Tannenbaum, “A shape-based approach to robust image segmentation,” in *CVPR*, 2006.

- [87] J. Kim, M. Çetin, and A. S. Willsky, “Nonparametric shape priors for active contour-based image segmentation,” *Signal Process.*, vol. 87, no. 12, pp. 3021–3044, 2007.