ROBUST FACIAL FEATURE LOCALIZATION IN FRONTAL VIEWS

by

Oya Çeliktutan

B.S. in EE, Uludag University, 2005

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Electrical and Electronics Engineering

Boğaziçi University

2008

ACKNOWLEDGEMENTS

I would like to thank all the individuals who have helped me during my master thesis. First of all, I would like to express my sincere appreciation to my supervisor, Prof. Dr. Bülent Sankur, for his guidance and support throughout my graduate studies since 2005. I am grateful to my committee members, Prof. Dr. Lale Akarun and Assoc. Prof. Burak Acar, for their valuable ideas, suggestions and encouragement.

I would like to thank my colleagues in the BUSIM Lab for their patience, friendship, tolerance and support: Erinç Dikici, Helin Dutağacı, İpek Şen, Hatice Çınar Akakın, Lucía Teijeiro Mosquera, Arman Savran, Cem Demirkır, Sıddıka Parlak Polatkan, Sinan Yıldırım, Sergül Aydore, Ekin Şahin, Neslihan Gerek, Çağlayan Dicle and Assoc. Prof. Murat Saralar. I am also thankful to all my eNTERFACE'07 team mates and Niyazi Ölmez for their support and all volunteers for their generosity in constructing the Bosphorus Face Database.

I owe a special word of appreciation to Erdal Kayacan who have shared all the happiness, sorrow, excitement with me and always encouraged me when I lost my self-confidence. It would be so much harder without him.

Finally, my deepest gratitude is to my parents and my brother who have supported and motivated me with their never ending love, trust and understanding throughout my life.

ABSTRACT

ROBUST FACIAL FEATURE LOCALIZATION IN FRONTAL VIEWS

In this thesis, we focus on the reliable detection of facial fiducial points in frontal face images, such as eye, eyebrow and mouth corners. The proposed algorithm aims to improve automatic landmarking performance in challenging realistic face scenarios subject to high-valence facial expressions and occlusions.

In order to extract facial fiducial points, we explore the potential of several feature modalities, namely, Gabor Wavelet Transform (GWT), Independent Component Analysis (ICA), Non-negative Matrix Factorization (NMF) and Discrete Cosine Transform (DCT), both singly and jointly. The multitude of landmark candidates is associated via fusion techniques. We show that the selection of the highest scoring face patch as the corresponding landmark is not always the best, but that there is considerable room for improvement with the cooperation among several high scoring candidates and also using a graph-based post-processing method.

The developed methods are tested on Bosphorous, JAFFE (Japanese Females Facial Expression Database) and BioID face image databases. We also present comparative results with "Elastic Bunch Graph Matching" algorithm. The performance of each method and each conducted experiment is discussed separately.

ÖZET

ÖNYÜZ İMGELERİNDE NİRENGİ NOKTALARININ GÜRBÜZ OLARAK SAPTANMASI

Bu çalışma önyüz imgelerinden göz, kaş ve dudak kenarları gibi nirengi noktalarının otomatik ve güvenilir bir şekilde saptanması amacı ile yapılmıştır. Çok çeşitli ifade ve kapatılma içeren yüz imgelerinde otomatik nirengileme başarımını artırmak üzerine çalışılmıştır.

Öznitelikleri çıkarmak amacıyla dört farklı yerel öznitelik sezicisi kullanılmıştır: Ayrık Kosinüs Dönüşümü (AKD), Gabor Dalgacık Dönüşümü (GDD), Negatif Olmayan Matris Ayrıştırma (NOMA) ve Bağımsız Bileeenler Analizi (BBA). Önerilen yöntemlerin performanları bireysel ve tümleştirilerek karşılaştırılmıştır. Herhangi bir nirengi noktası için en yüksek skora sahip aday noktasını seçmek her zaman güvenilir olmayabilir; yüksek skora sahip noktaların işbirliği ve çizge modellerin kullanımı ile performans arttırılabilir.

Geliştirilen yöntem Boğaziçi, JAFFE (Japanese Females Facial Expression Database) ve BioID veri tabanlarında test edilmiştir. Önerilen her bir yötemin performansı ve yürütülen her bir deney ayrı ayrı incelenmiştir. Ayrıca "Elastic Bunch Graph" algoritması ile de karşılaştırmalı sonuçlar verilmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS/ABBREVIATIONS

a	Magnitude of a Gabor jet
A	Mixing matrix
B	Face Bunch Graph
C	Cost parameter
$C(.)$	Combination operator
$C(u, v)$	DCT coefficients matrix
$\overrightarrow{d}$	Displacement vector
d_{norm}	Normalization term; distance between two eye centers
E	Energy of a *k-landmark* configuration
$E\{\}$	Expected value
$g(.)$	Derivative of $G(.)$
$G(.)$	A non-quadratic function
G^{Bm}	A model graph
H	Encoding vectors
$H(.)$	Entropy
$I(x, y)$	A face image
J	A Gabor jet
$J(.)$	Negentropy
J_n^{Bm}	A bunch of Gabor jets
$\overrightarrow{k}_j$	Wave vector
$K(x_i, x_j)$	Kernel function
m_{error}	Error measure in terms of Euclidean distance
Q_{ij}	Positive semi-definite matrix
s	Independent component
$S_a(J, J')$	Magnitude similarity between two Gabor jets
$S_\phi(J, J')$	Phase similarity between two Gabor jets
$S_B(G^I, B)$	Graph similarity function
X	Data matrix
$(\tilde{x}, \tilde{y})$	Ground truth position of a landmark point

W_I	Whitening matrix
y	Separated output
γ	Kernel parameter
$\overline{\lambda}_{i,j}$	Mean length between the facial fiducial points
$\Lambda^2_{i,j}$	Variance of $\lambda_{i,j}$
ϕ	Phase of a Gabor jet
$\overline{\phi}_{i,j}$	Mean relative angle between the facial fiducial points
$\Phi^2_{i,j}$	Variance of $\phi_{i,j}$

AAM	Active Appearance Model
ASM	Active Shape Model
AU	Action Unit
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
ICA	Independent Component Analysis
IGF	Independent Gabor Features
IOD	Inter ocular distance
EBGM	Elastic Bunch Graph Matching
FACS	Facial Action Coding System
FBG	Face Bunch Graph
GWT	Gabor Wavelet Transform
ICA	Independent Component Analysis
IOD	Inter ocular distance
LIBSVM	Library for Support Vector Machines
LDA	Linear Discriminant Analysis
MLP	Multi-layer Perceptron
NMF	Non-negative Matrix Factorization
PCA	Principal Component Analysis
PGM	Probabilistic Graphical Model
RBF	Radial Basis Function
SDF	Synthetic Discriminant Function

SFFS	Sequential Floating Forward Search
SVM	Support Vector Machine

1. INTRODUCTION

In computer vision, face processing has grown significantly over the past decade. It has been studied by many researchers for different purposes, i.e. face recognition, face tracking and 3D face reconstruction. Facial feature localization forms a crucial step in such applications.

1.1. Scope of the Thesis

1.1.1. Definition of the Problem

Facial feature localization is the process of determination of facial features such as eyes, eyebrows, nose and lips, in the images that contain faces. In the context of face processing, the expression *facial feature* is used as equivalent to *facial component*, i.e eyes, nose, mouth, etc. and their fiducial points; tip of nose, the corners of the mouth, the corners of eyes and eyebrows refer to as landmark points. In this thesis, we will use both of the expressions, *facial feature localization* and *landmarking*, specifying each time the intended meaning.

Facial feature localization has long been a popular research area. Its wide range of applications and difficulty render facial feature localization an attractive problem for scientists. Despite many remarkable improvements, facial feature localization remains still a challenging computer-vision task due to the following reasons:

- Intrinsic variability: Facial expressions, pose, occlusions due to hair or hand movements or self-occlusion due to rotations impede successful feature detection. A unique facial feature localizer that will work well under all intrinsic variations of faces and that will deliver in a time efficient manner the target features has not yet been feasible.
- Acquisition conditions: Much as in the case of face recognition, acquisition conditions, such as illumination, resolution and pose greatly affect the localization

performance. For example, feature localizer trained in one database (say, ORL) can perform poorly in another database, (say, FRGC) where the acquisition conditions differ substantially from that of the former.

The facial features can be classified into two major types: first-order (primary) features and second-order (secondary) features. Discrete features (e.g. eyes, eyebrows, mouth, nose, chin etc.), their corners and edges defined between corners, which have been found to be fundamental in determining facial identity and expression and specified without reference to other facial features, are called as first-order features. Another configurable set of features which characterize the spatial relationships between the positions of the first-order features and information about the shape of the face are called as second-order features. These are called secondary in that they have more scarce low-level image evidence and typically found by graph structures determined by primary landmarks. Examples of important second-order features are nostrils, chin, nose bridge, cheek contours as many 62 points as given in [1].

1.1.2. Application Contexts

In general, facial feature localization is considered as a separate task in the literature. As a consequence, the face processing techniques are usually evaluated against manually landmarked points in the most of the related studies. Only recently researchers have become aware of the importance and necessity of automatic facial feature localization for building a fully automatic system. Correct localization of the facial features directly affects the overall performance of the face processing system and guarantees the robustness of the whole application. Therefore, facial landmarking is a fundamental step in many different applications such as facial expression analysis, face animation, 3D face reconstruction and it is instrumental in face recognition and face tracking. In the following, we briefly mention these applications.

1.1.2.1. Face Recognition. Automatic face recognition is relatively a new concept. A general definition of this problem is as follows: given a still image or video sequence

of a scene, identify or verify one or more persons in the scene using a stored database of faces. Developed in the 1960's, the first semi-automatic system for face recognition required an operator to locate the facial features manually on the images before it calculated distances and ratios to a common reference point, which were then compared to reference data. Today, state-of-the-art face recognition technology is being used to improve security and passenger processing in airports, support law enforcement, identify missing children, and investigate criminals.

Face recognition techniques can be classified according to several criteria. Here, we focus on two categories: holistic (global) approaches and feature-based (local) approaches. Holistic approach treats the raw face data as a whole and attempts to extract a suitable representation which can be used for classification. Examples of this approach are eigenfaces method, which is a PCA based technique [2], linear discriminant analysis [3] and template matching [4]. However, these methods are highly sensitive to translations, rotations and appearance variations and fail to capture intrinsic variations such as pose changes, size and illumination within the face class. In these circumstances feature-based methods offer a better representation of the face class and its typical variations.

Feature-based approach operates on local structures, namely facial features. Properties and geometrical relations of these features can then be used to construct a representation of the face. Examples of this approach are face recognition based on Support Vector Machines [5], eigenfeatures technique, which is a localized version of the eigenfaces [6] and Elastic Bunch Graph Matching (EBGM) [7].

1.1.2.2. Face Tracking. The term facial feature localization is generally used when static images are the concern, and tracking is used for referring the process of continuously localizing and tracking the features in video sequences. Facial feature tracking constitutes the basis of many applications such as human-computer interfaces, dynamic facial expression analysis systems, facial animation, driver fatigue detection systems, 3D face reconstruction, model-based coding of video sequences, etc. Tracking facial

features is also a very challenging problem due to uncertainties of poses and appearance of faces and environment conditions. Proposed methods for this problem can be classified into two categories: model-based methods [8], [9] and model-free methods [10], [11], [12]. In model-based methods, the shape of the facial features and geometrical relations between them are used in constructing a face model. Tracking is realized by updating the model shape parameters and suitably fitting the constructed model in each frame. Contrary to model-based approaches, there are no shape constraints and prior knowledge in model-free methods. These methods are basically based on motion estimation. Once the facial features are localized in the reference frame, the position of facial features in the subsequent frames can be found by a local search around the previously estimated locations.

1.1.2.3. Facial Expression Analysis. Facial expression is a visual way to communicate emotions and improve the understanding of spoken communication. However, the use of facial expression analysis has some limitations because of being human-observer dependent, labor intensive and difficult to standardize. The Facial Action Coding System (FACS) developed by Ekman and Friesen [13] provides an objective description of facial behavior. It decomposes facial expressions into action units (AUs) that roughly correspond to independent muscle movements of the face. The most promising approaches are based on recognizing a combination of these Action Units (AUs) instead of global expressions [14], [15], [16]. Here, facial feature localization plays an important role in extracting the information about the encountered facial expression in an automatic way.

1.1.2.4. 3D Face Reconstruction. In recent years, the use of 3D face models enables illumination-free and pose-free face recognition. They are also useful in many applications such as medicine, video compression/coding and computer graphics. 3D face models can be obtained by 3D scanner devices, e.g., laser range sensors, structured light, stereo cameras. However, 3D data acquisition process is often problematic. The final model is usually noisy because of the holes and spikes occurred due to head movements, inappropriate illumination conditions and head positions. A more advantageous

approach is to construct a 3D face model from multiple images [17], video sequences [18] or even a single image [19]. Automatic landmarking is necessary for developing such a system that is able to build face 3D models with minimal user interaction.

1.1.3. Project Objectives

In this thesis, we address the problem of the accurate localization of principal or primary facial fiducial points, which are the nose tip, chin tip, the two mouth corners, the four inner and outer eye corners, and similarly the four eyebrow corners, in total 12 points. The novelty of our facial landmarking algorithm is the use of a multi-feature framework. We model facial landmarks redundantly by four different feature categories, namely, Discrete Cosine Transform (DCT), Non-negative Matrix Factorization (NMF), Independent Component Analysis (ICA) and Gabor Wavelets. We run them in parallel in order to subsequently fuse their estimates or decisions so as to combat the effects of illumination and expression variations. We have also contrasted our proposed multi-attribute landmarking method with one of the most pioneering works, Elastic Bunch Graph Matching (EBGM) [7].

1.2. Thesis Outline

The outline of the thesis is organized as follows. Previous work is reviewed in Chapter 2. Automatic facial landmark extraction algorithm, including feature extraction to decision fusion stage, is given in Chapter 3. Graph-aided correction and Elastic Bunch Graph algorithm are introduced in Chapter 4. The database and experimental results are provided in Chapter 5. Finally, we discuss future work and draw our conclusions in Chapter 6.

2. OVERVIEW OF FACIAL FEATURE EXTRACTION

2.1. Previous Works on Facial Feature Extraction

There exist several methods for facial feature localization in the literature. These methods can be categorized into four main groups based on the type of the observed data and apriori information they exploit, namely:

2.1.1. Appearance-based Approaches

These approaches generally use texture (intensity) information only and learn the characteristics of the landmark points and their neighborhoods projected in a suitable subspace. The data is projected into the space in order to provide a better conditioned, more compact representation and tested for localization by using a suitable distance metric in the projection space. One pioneering example of the these approaches is the eigenfeatures method [6], [20], which is similar in nature to the eigenfaces approach with the main difference is that Principal Component Analysis (PCA) is performed for each of the features to be detected, instead of for the whole face. In [20], Ryu et al. used these features to train ensemble networks consisting of a series of independent Multi-layer Perceptrons (MLPs). Other examples of this approach are Gabor Wavelet Transform (GWT), Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), Linear Discriminant Analysis (LDA) and Independent Component Analysis (ICA), etc.

Among all existing methods, Gabor Wavelet Transform is the one that has been widely used in many face processing applications. Gabor wavelets are biologically motivated convolution kernels which has the shape of plane waves restricted by a Gaussian envelop function. Gabor wavelets of different orientations and scales respond differently to the features having different orientations and scales. This property of Gabor wavelets for collecting scale and orientation provides invariant metrics for matching. One approach is proposed by Smeraldi and Bigun [21]; SVM classifiers are trained by the responses of a bank of Gabor filters and applied to several image patches repre-

senting both positive and negative samples. Given a probe image, the trained SVMs are used to classify a randomly selected point as one of the features or discard it. The position of subsequent points are chosen with respect to the knowledge of that feature.

The proposed methods generally adopt a coarse-to-fine search strategy instead of directly attempting to locate facial features. For example, a two-level hierarchical wavelet network is presented by Krüger et al. in [22]. The first level wavelet network is used for face matching and yields a rough approximation of the feature locations. The second level wavelet networks, trained for each feature separately, are used to refine the initial feature locations. In another example, Akakın et al. first downsampled the face images and start searching on these low-resolution images for efficiency [23]. This step is followed by a refinement step in which we revert back to the original image and improve the coarsely estimated landmark locations. The authors used SVM classifiers with Gabor features in the coarse localization and DCT templates in the refinement stage in [24] or Gabor features and DCT templates, in both of the steps, separately as in [23]. Although the satisfactory performances of all these approaches, there is a limitation; for detecting a particular facial feature, it is necessary to process the whole face image, which may decrease the efficiency of the system. To overcome this problem, Zelek et al. [25] combined the Gabor wavelet coefficients with the entropy measure. The entropy of local intensities have a role in eliminating the irrelevant regions of the face image. Then, Gabor wavelet coefficients and local entropy are both used in comparing the candidate points with the training models.

Independent Component Analysis (ICA) is another outstanding technique in face processing applications. ICA can be applied directly to the facial feature extraction problem [26] or used for dimensionality reduction [23]. In [26], Thiran et al. used Independent Component Analysis to represent the face patches and perform the SVM classification in the ICA space. On the other hand, Salah et al. [23] employed a combination of Principal Component Analysis and Independent Component Analysis features to post-process the Gabor features and obtained so-called Independent Gabor Features (IGF) for localization.

Finally, the block-based DCT coefficients also produce good low-dimensional feature vectors with high localization performance. There are two types of information that can be used in detecting facial features: DC coefficients and AC coefficients. While DC coefficients represent the average color value of a $k \times k$ block, AC coefficients include information about the gray value changes inside the block. DC coefficient can be used for differentiate between color regions (i.e., eyes, mouth) and smooth regions (i.e., cheek, forehead) of the face; but AC coefficients are found to be more reliable and robust in precise detection of fiducial points. In [27], Zobel et al. used DCT features in a probabilistic structure in which the spatial dependencies between facial features are modeled. In a similar approach, Akakın et. al [28] used SVM classifiers, trained with DCT templates, to detect fiducial points and then the initial localization results are processed by a probabilistic graph-based framework.

2.1.2. Template-based Approaches

Template-based methods can be thought as similar to the appearance-based family; the essential differences are based on the matching procedure and the representation of facial features. Template-based methods can be divided into two categories: fixed and deformable.

In techniques based on fixed templates [4], [29], the appearance of each feature is represented by a template, used to locate that feature in a probe image. The matching procedure is carried out by means of cross-correlation technique, namely convolving the input image with the template. Good matches are interpreted as detected features. In [4], Brunelli and Poggio used a single template to compare with the image using a distance metric. The relevant face regions are obtained by integral projection. In a further study [30], the authors generalized the matched filter and introduced Synthetic Discriminant Function (SDF) in which a filter is built as a linear combination of matched filters for different patterns and thus, multi-class pattern recognition can be achieved. The SDFs has comparable performance in eye detection. Another approach for locating mouth and eye corners is presented by Zhang in [31]. This method used smaller templates for detecting corners of these features instead templates for whole

mouth and eye. After detecting corner locations, geometrical verification conditions are considered for irrelevant matches. However, these techniques are too rigid for facial feature extraction; both the feature scale and appearance are fixed when a template is defined. Thus, the major drawback of these techniques is scale and pose dependency. The pose or illumination variations can be accomplished by using multiple templates. For scale dependency, Poggio et al. proposed a multi-resolution method in [5]. After the whole image is scanned, each image block centered at the corresponding pixel is extracted in a multi-resolution way. Then, the differently scaled squares obtained from the image are given as input to SVMs trained to recognize a specific feature (eyes or mouth).

Deformable template approaches are proposed to cope with the limitations of classical template matching methods. The pioneering study using deformable templates was presented by Yuille et al. [32]. The deformable templates are specified by a set of parameters which uses a priori knowledge about the expected shape of the features to guide the contour deformation process. The templates are flexible enough to change their size and other parameter values so as to match themselves to data. An energy function is defined which contains terms attracting the template to salient features such as peaks and valleys in the intensity image and edges. In [32], the peak, valley and edges of the image are extracted by using morphological filters and smoothed by convolving with a Gaussian filter. The energy function is minimized by gradient descent algorithm. In a similar work [33], Herpers et al. obtained a saliency map by applying a bank of high pass filters where the features correspond to the maxima in this map. Thus the maxima is extracted and the corresponding portions are verified to be a feature by means of deformable templates; the optimization criterion is based on the amount of deformation of the template. In a recent study [34], Zhang and Ruan combined the fixed and deformable templates for facial feature extraction. First, fixed templates are used to locate a rectangle block isolating the facial features in the face image and, then deformable templates are used to extract the contour of them.

Active contours (snakes), which can be considered as generic deformable templates based on control points and parametric curves connecting them, form another

example of extracting the contours of facial features by deformable templates. This algorithm tries to adapt the initial model to the image data by moving the control points and by adjusting the parameters that regulate the curve shape. An active contour lip model is used in [35] to track the lips of a speaker in a biomodal speech recognition system. The algorithm is initialized by color segmentation applied for lips. Some recent approaches utilized Active Shape Models (ASM) or Active Appearance Models (AAM) for extracting facial feature points [36]. The ASM [37] proposed by Cootes et al. is a popular statistical approach to represent deformable objects, where shapes are represented by a set of feature points. The feature points are searched by local templates modelling the grey-level texture and principal component analysis is applied to analyze the shape variations so that the face shape can only deform in specific ways learned through a training session. Fitting to a large number of facial points reduce the effect of individual erroneous feature detections. The AAM [38] is subsequently proposed to combine constraints of both shape variation and texture variation. However, the main source problem when using these approaches is that they need a good starting configuration to avoid local minima.

2.1.3. Geometry-based Approaches

These methods are based on the geometric relationship (distances, angles etc.) between facial features. They have some success in detecting facial features; but they cannot individually handle large variations in face such as rotations, facial expressions and inaccurate landmarks. For example, Bhattacharya et al. introduced an anthropometrical face model in [39]. The algorithm is initialized by the detection of the eyes as all anthropometric measurements are described in terms of the distance between two eye centers. Based on the anthropometrical information, the corresponding region of each facial feature is located, and then further processed by a combination of different image processing techniques for precise detection of the fiducial points. In [40], Chuang and Shih proposed a double-threshold method in which the high-thresholded image is used for extracting head boundary and the low-thresholded image for face boundary. Then, facial features are obtained by applying x- and y-projections to the detected face region, and lastly the localization results are improved by a geometric face model.

The existing methods adopt an exhaustive search which is generally non-practical or introduce some heuristics to reduce the feature search area. In order to reduce the search area, the simplest approach based on the fact that gray levels corresponding to the features are darker than the skin ones. After being localized in the face region, the features are detected by projecting horizontally and vertically the gray levels, i.e., eyes correspond the rows with minima values [41]. In [42], Atalay et al. detected the smallest rectangles enclosing the facial features by using vertical and horizontal gray value projections of pixels. Analysis of such horizontal and vertical integral projections also fall into geometry-based category. However, such techniques are not very reliable individually; they can be combined with other techniques in order to attain robust facial feature localization. For example, in [43], Pantic et al. used gray level intensities and Gabor features in conjunction with boosted classifiers; but first the detected face is divided into relevant regions of interest, each of which is further examined by vertical and horizontal histograms. Such another approach is proposed by Lanzarotti et al. [44]. First, facial features are located based on deformable templates and characterized by different Gabor filter responses. Then, wrongly determined fiducial points are recovered on the basis of some rules derived from the anthropometrical relationship between the reliable ones.

2.1.4. Graph-based Approaches

These approaches generally process together the intensity and geometric information, and seem to be very effective in many cases. The idea is based on constructing a graph from anthropometrical information and deforming this graph according to a defined criterion. One of ground-breaking examples of this approach is Elastic Bunch Graph Matching (EBGM) proposed by Wiskott et al. [7]. EBGM algorithm exploits Gabor jets for registering facial feature points. Their method aims face recognition by using similarity measures on the elastic bunch graph and the Gabor wavelet responses of facial feature points at different scales and orientations. The nodes of the graph correspond to fiducial points, and at each node Gabor wavelet responses at different scales and orientations are stored. In the same vein, Ji et al. proposed to use Viola-Jones object detector, relying upon the Adaboost algorithm and a set of Haar Wavelet-like

features, to locate face and eyes and, then fit a trained mesh to the face based on the located eye positions [45]. Thus a rough position for each facial feature is estimated and refined by EBGM technique. The only difference is that the nearest neighbor searching approach is utilized instead of displacement estimation in computing the similarity of Gabor coefficients. In another work [46], Bloch et al. proposed inexact graph matching technique for feature localization. The face is segmented into relevant regions by using watershed algorithm; the face graph in which the nodes are attributed by average grey values and wavelet coefficients and the edges by distance vectors, is built from these regions and relationships between them. The global dissimilarity function, defined based on the comparison of the attributes of two graphs, is optimized by several stochastic algorithms such as randomized tree search, genetic algorithms etc.

Besides global approaches (i.e., EBGM, ASM), another approach is to individually locate the fiducial points, generally by machine learning techniques like Adaboost, SVMs, and then improve the results via a graphical model. In [28], first, SVM classifiers trained with DCT templates are used to detect candidate feature points. The irrelevant points are eliminated by a Probabilistic Graphical Model (PGM) in which the best configuration of a subset of candidate points is established considering the distances and angles between the fiducial points; namely n reliable points are selected out of k candidate points where n is smaller than k. Then, the remaining fiducial points are recovered by back projection method based on the spatial locations of the reliable ones [23]. Shape constrained Adaboost algorithm [47] adopted a similar approach; Cristinacce and Cootes adapted the Viola-Jones's face detection algorithm [48] to locate individual features within the face region. To overcome false positive responses, a shape model is fitted to a set of candidate points by least squares algorithm. An efficient search method is introduced to select the best candidate among many possible feature configurations and the positions of missing points are predicted by the shape model.

2.1.5. Other Approaches

There exists a huge number of facial feature localization methods in the literature; and it is impossible to categorize all these different methods into four groups. Hence, we introduced four more groups:

2.1.5.1. Color-based Approaches:. These approaches are capable of robustly detecting skin colors in the presence of complex background and different illuminations. Any non-skin color region within face is viewed as a candidate for eyes or mouth. The main limitation of these algorithms is being applicable only for color images. In [49], Jain et al. proposed a skin segmentation method based on a non-linear YCbCr color model. The possible face regions are detected by using an elliptical skin model in the transformed chrominance components (Cr and Cb), and then connected components analysis is applied for grouping and labelling skin pixels. In the face candidates, the eyes are automatically detected based on luminance and PCA edge direction information, and the remaining facial features by geometrical shape information.

2.1.5.2. Edge-based Approaches:. Facial features are detected without any intensity and motion information; but solely relying upon the edge information. In one example, presented by Phimoltares et al. [50], first the face detected in the input image by using Canny edge detector and by comparing the resulting edge image with a mean face template. The mean face template is rescaled to various sizes and rotated to different angles to cope with the possible variations of faces. In the second stage, to coarsely extract the facial features from the edge map of the detected face, the authors proposed a neural vision model (NVM) that is fundamentally based on multilayer perceptron (MLP) network. Then, to improve the results, irrelevant regions are removed by applying image dilation. The proposed algorithm exhibits an adequate performance under occlusion, intensity changes, facial expressions, orientation variations and noise. However, finding the optimal thresholds for the face detection task is a difficult problem. Another drawback of this approach is that edge appearance is sensitive to illumination, complex background and skin wrinkles.

2.1.5.3. Motion-based Approaches:. Facial features are detected from the image sequences. Using such methods, facial features cannot be detected using only a single still image from one view. In video sequences, head motion allow to determine roughly head and facial feature positions. Such techniques try to match a face model to the moving region; the model of vertices correspond to the features, and thus inform on their rough positions. For example, in [51], Ahlberg proposed to use a wire-frame model which has to be adapted to the face on the first frame. A correspondence between the vertices of the model and the related face image features is established by statistical template matching. At each successive frame, only the variations of the point positions are transmitted.

2.1.5.4. 3D Vision-based Approaches. Recent advances in 3D acquisition devices provide opportunities to use 3D face data. One advantage is that the 3D face measurements are independent from lighting and pose variations. However given the noisy nature of the 3D data, they require a preprocessing stage to reduce spikes and discontinuities. Recent 3D landmarking studies can be listed as [52], [53], [54], [55], etc. In addition, multi-modal approaches that combine 2D and 3D information, have shown promising performance [23], [56]. In this thesis, we focus on 2D facial feature extraction methods; therefore these approaches are out of scope of our work.

The Figure 2.1 summarizes the available automatic facial feature localization methods and their categories.

2.2. Our Approach

Our approach is a combination of appearance-based and graph-based methods. Basically, we run four appearance-based channels in parallel, fuse the outcomes and, then resort to a graphical model to correct gross localization errors or estimate the missing features.

The general framework of the proposed method is given in Figure 2.2. First, the

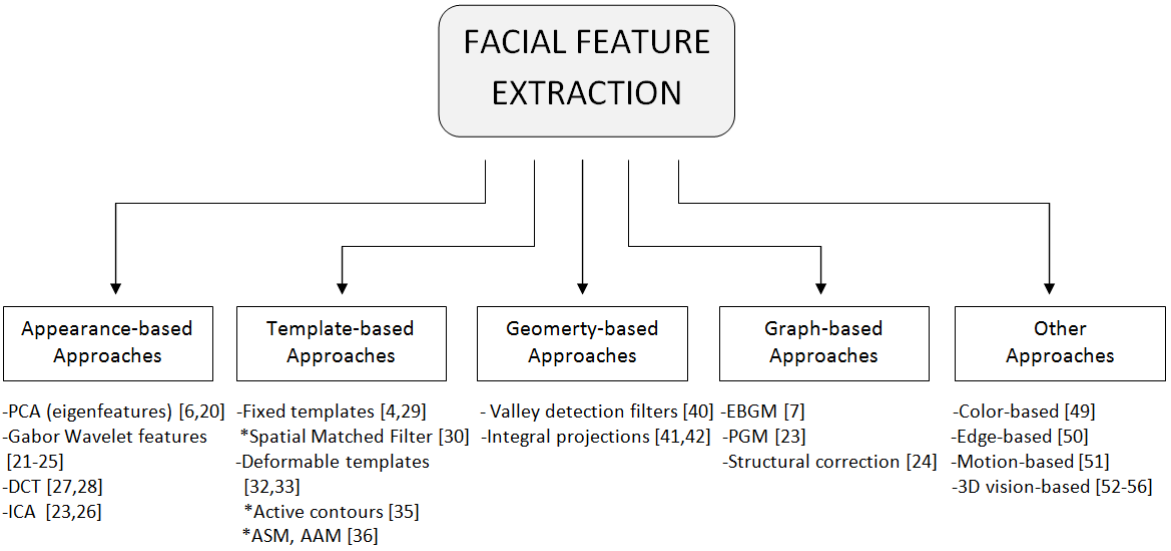


Figure 2.1. Methods for automatic landmark localization.

original 640×480 image is downsampled to 80×60 resolution for a two-tier search. In the first low-resolution tier, each feature channel identifies the candidate landmark points in the face image. For each landmark, we allow a number of candidates that will take role in a voting or fusion scheme as outputted by the SVM. These candidates are qualified with the goodness scores given by the corresponding SVM. Consequently, the candidates are accompanied by their goodness scores supplied by the SVM. The multitude of landmark candidates from feature channels, first have the SVM scores normalized and then they are fused into one landmark. When multi-feature approach is not able to locate landmark points at the coarse level, we declare a missing landmark and we try to recover these points via a graph-based back projection, that is, simply estimating their position using the face graph fitted to the detected landmarks as in [23]. In the second tier, the coarse-level landmarks are transferred to the full-resolution image, where they are refined by searching for a better fit around the coarsely estimated points. The further details of each step are given in the following chapters.

2.3. Performance Evaluation

The performance evaluation of a feature extraction method depends on the nature of the feature. Here, we only consider the methods that aim at localizing the position of the facial fiducial points. Given true positions of the searched features (by manual an-

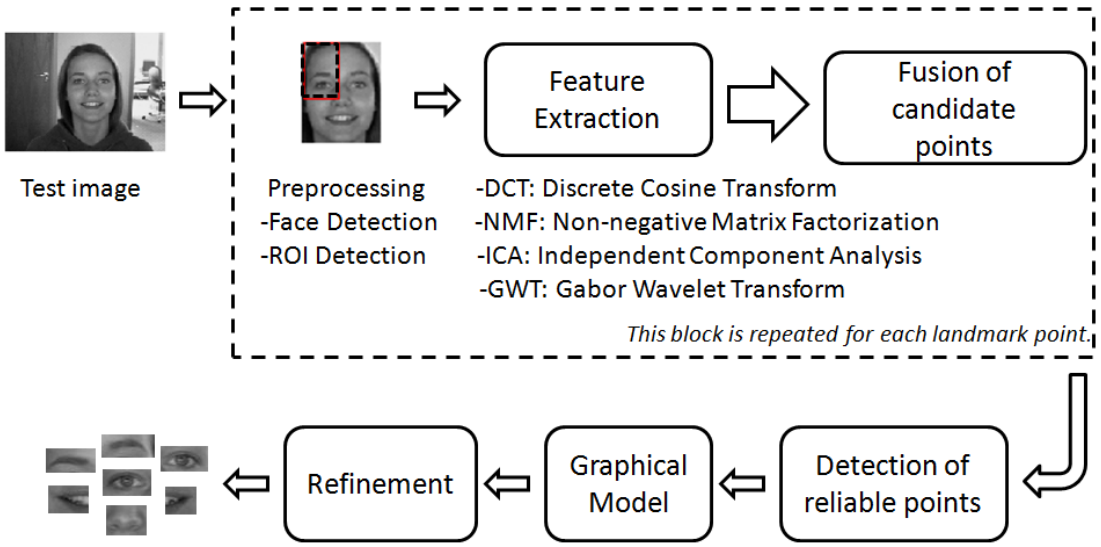


Figure 2.2. Block diagram of the proposed method.

notation), the localization accuracy is expressed as a statistics of the error distribution made over each feature (usually the mean), measured as the Euclidean pixel distance. In order to make these statistics meaningful, so that they can be used to compare the results on any dataset, it is necessary to standardize the error by normalizing it over the face scale.

One popular error measure, which has been adopted by many works on feature localization, can be defined as

$$m_{error} = \frac{d_{euclidean}\{(x, y), (\tilde{x}, \tilde{y})\}}{d_{norm}} \tag{2.1}$$

where (x, y) is the result of automatic localization, $(\tilde{x}, \tilde{y})$ is the ground truth position of the search point and d_{norm} is the normalization term, the inter ocular distance (IOD: the Euclidean distance between left and right eye centers). There is a general agreement that $m_{error} < 0.1$ is a good criterion to claim correct localization. In this thesis, we refer to this threshold as *acceptance threshold*. Then, the performance can be expressed as the percentage of the correctly detected points among all searched points.

2.4. Summary

Despite many sophisticated methods, automatic facial feature localization is still an open problem. The proposed solutions generally consider head and shoulder images, without addressing the problem of localizing the face in a clutter scene. This assumption greatly simplifies the problem, but it would be useful to put more emphasis on the face detection and face segmentation. Another open problem is the scale dependency: most of the works make an assumption of the face scale, and limit their generality.

Human face appearance has potentially very large intra-subject variations due to poses (yaw-pitch-roll rotations of the head), illumination conditions, facial expressions, facial hair, occlusions due to other objects or accessories (hand, hair, glasses, scarf etc.) and aging. In some cases, the inter-subject variations can be smaller than intra-subject variations due to the similarity of individual appearances. Another major drawback of the systems is the need for a large number of training images taken from different viewpoints and under different conditions to handle all these variations. Therefore, the proposed algorithms are generally incapable to deal with occlusions, extreme facial expressions and large pose variations. These lead to limit the applicability of existing methods to only well controlled environments.

3D landmarking approach proposes a solution to the limitations of 2D approaches. Because the human face is a 3D object, its 2D projection (image) is sensitive to relative changes while 3D face data provide a huge geometrical information and a significant advantage over 2D face images in the case of illumination and large pose changes. However, acquisition and preprocessing steps of 3D data are quite demanding due to its noisy nature.

3. INDIVIDUAL FACIAL FEATURE EXTRACTION METHODS

3.1. Description of Facial Features in Subspaces

The typical appearance of each facial feature can be characterized by different approaches. In this thesis, we investigate four different methods: DCT templates (Discrete Cosine Transform), Gabor Wavelet coefficients (Gabor Wavelet Transform), encoding vectors produced by NMF (Non-negative Matrix Factorization) and mixing coefficients of ICA (Independent Component Analysis). For this purpose, we first downsample the face images from their actual size to 80×60 resolution for a computationally efficient system. Several $k \times k$ patches (e.g., $k = 8$ for the coarse level) centered at facial fiducial points, i.e. eye corners, nose tip, mouth corners, plus non-fiducial points are cropped, vectorized and stored in the columns of a data matrix. Then, we extract relevant features from the data matrix separately and use them to train individual classifiers, i.e., Binary Support Vector Machines.

3.1.1. Discrete Cosine Transform

DCT templates represent the intensity changes and statistical shape variations in a given image block. They are proved to be quite discriminative in the previous studies [28], [23]. At a given point $I(x, y)$, the $C(u, v)$ matrix containing DCT coefficients is computed as follows:

$$C(u,v) = \alpha(u)\alpha(v) \sum_{x=0}^{K-1} \sum_{y=0}^{K-1} I(x,y) \cos\left[\frac{(2x+1)u\pi}{2K} \right] \cos\left[\frac{(2y+1)v\pi}{2K} \right] \tag{3.1}$$

for $u, v = 0, 1, 2 \ldots, K-1$, where $\alpha(u) = \sqrt{\frac{1}{K}}$ for $u = 0$ and $\alpha(u) = \sqrt{\frac{2}{K}}$ otherwise. The features are the low to bandpass coefficients; the coefficients are ordered according to a zigzag pattern, depending on the amount of information stored in them.

Figure 3.1. (a) Original image, (b) DCT template, (c) Reconstructed image, % 5 of the coefficients is saved.

The first coefficient (DC value) is removed as it represents the average intensity value of the block. From 8×8 blocks, two thirds of the DCT coefficients in the zigzag pattern are considered as the feature vector and fed into their respective SVMs. In Figure 3.1, we give an example of an eye patch reconstructed via DCT templates.

3.1.2. Independent Component Analysis

ICA aims to express a set of random variables as linear combination of statistically independent component variables. ICA can be thought as a generalization of PCA. The main difference is that PCA seeks directions in feature space that best represents the data in a sum-squared error sense, ICA instead seeks directions that are most independent from each other. ICA can be used for both of two purposes feature extraction and blind source separation.

In matrix notation, we can express the ICA problem as $x = As$. This describes how the observed data are generated by a process of mixing components. Here, A denotes the mixing matrix. We must estimate both A and s just by using observed data. In ideal conditions, once we estimate A, we can compute the inverse, $W = A^{-1}$ and obtain the independent components simply as $s = Wx$.

In our experiments, we used the Fast ICA algorithm [57]. However, Fast ICA algorithm have some restrictions. Since mixing matrix cannot be estimated for Gaussian

independent components, Gaussian variables are forbidden. In order to obtain independent components, we have to find a vector w that maximizes the non-gaussianity of $w^T x$. ICA achieves this goal by maximizing the cost function negentropy which is a measure of non-gaussianity. Negentropy can be defined as $J(y) = H(y_{gauss}) - H(y)$ where $H(.)$ denotes the entropy and y is the separated output (the learned ICA output). Negentropy is always non-negative; it only equals to zero when y has Gaussian distribution. The estimation of negentropy is difficult; so in our model we use an approximation of its value as $J(y) = (E\{G(y)\} - E\{G(v)\})^2$. Here, G is a non-quadratic function and v is a Gaussian random variable. We define G function as $G(u) = \frac{1}{a_1} \log \cosh a_1 u$ and its derivative as $g(u) = \tanh a_1 u$ for $(1 \leq a_1 \leq 2)$.

Another restriction of ICA model is that both mixture models and independent components are assumed to be having zero mean and unit variance without loss of generality. Consequently, ICA algorithm requires a preprocessing step (whitening process). Finally, the derived algorithm is given in Figure 3.2.

Once we obtain W, we can compute independent sources as $Y = W\tilde{X} = WW_I X$ where W_I is whitening matrix. Then, separating matrix $B = WW_I$ and mixing matrix $A = B^{-1}$.

There are two different approaches in the use of ICA for feature extraction [58]. In Architecture I, the face images are modeled as a linear mixture of statistically independent source images while in Architecture II, the representation coefficients are assumed to be statistically independent. In the first approach face images take place in the rows of the observation matrix X while in the second approach they constitute its columns. In Figure 3.3, we visually compare the representation of facial features obtained by two architectures. As seen, while Architecture I provides a sparse representation, Architecture II exhibits holistic basis images as in PCA-case.

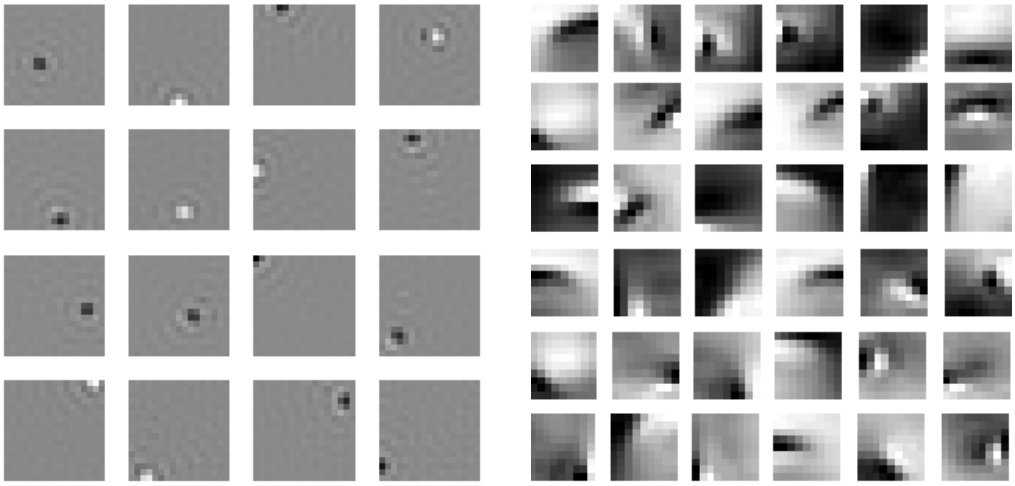
In our experiments, we adopted Architecture I for facial landmark detection problem as illustrated in Figure 3.4. The data matrix, X is first subjected to PCA so that the data is first projected on the eigenvector matrix, V, that is $R = XV$. The ICA

- ***Preprocessing:***
  - Centering: Set the mean of the vectors to zero.
    $x - m \rightarrow x$, $m = E\{x\}$.
  - Whitening: Sphere the vectors. Thus, the components of x are uncorrelated and their variances are equal to unity. In other words, covariance matrix $E\{\tilde{X}\tilde{X}^T\}$ is equal to identity matrix. This can be accomplished by Eigen Value Decompositon (EVD). $W_I x \rightarrow \tilde{x}$
- ***Fast ICA algorithm:***

  Goal: Maximize negentropy, $J(y)$ by fixed point iteration scheme.

  1. Initialize the weight vector w
  2. $w^+ = E\{xg(w^T\tilde{x})\} - E\{g'(w^T\tilde{x})\}w$
  3. $w = w^+ / \parallel w^+$
  4. To estimate several independent components, we need to run one-unit Fast ICA algorithm using several units with weight vectors $w_1, ..., w_n$. To prevent different vectors from converging the same maxima we must decorrelate the outputs $w_1^T\tilde{x}, ..., w_n^T\tilde{x}$ after every iteration. This can be accomplished by symmetric orthogonalization $W = (WW^T)^{1/2}W$.
  5. If not converged, go back to step 2. Here, convergence means that the old and new values of W point in the same direction and dot product is equal to 1.

Figure 3.2. Fast ICA algorithm [57].



(a) (b)

Figure 3.3. Representation of the facial data via two different ICA architectures: (a) ICs of the image set obtained by Architecture I, which provide a set of statistically independent basis images (rows of Y), (b) Basis images for the ICA-factorial representation (columns of $A = W^{-1}$) obtained with Architecture II.

analysis is performed on V^T, where eigenvectors form the rows of this matrix. Then, we obtain ICA basis images as $S = WV^T$. By using the PCA representation coefficients R and separating matrix W, the mixing matrix is calculated as $A = RW^{-1}$. In testing stage, a given test block, x_{test} is first projected onto feature subspace $r_{test} = x_{test}V$. Then, ICA feature vector is obtained by multiplying with the inverse of the separation matrix, $a_{test} = r_{test}W^{-1}$. The decision as to whether a given patch x_{test} corresponds to one of the fiducial landmarks is based on the comparison between the training feature vectors $\{a_1, a_2 \ldots a_k\}$ and the patch data appropriately projected onto principal components and then de-mixed, that is, a_{test}. Here, we use 52 factors for the coarse level.

3.1.3. Non-negative Matrix Factorization

Non-negative Matrix Factorization (NMF) is a matrix factorization technique that decomposes the data as a product of two matrices that are constrained by having non-negative elements. The original non-negative matrix factorization method was

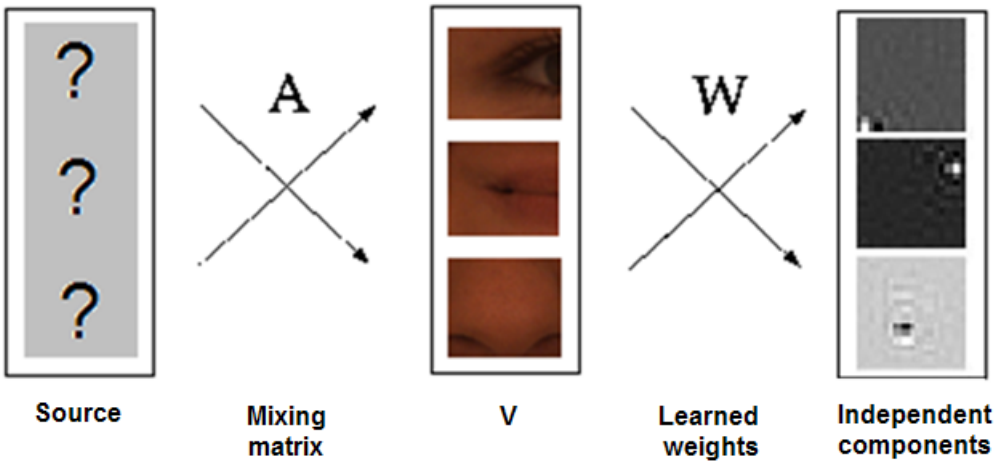


Figure 3.4. Feature extraction method via Architecture I. The feature components in X are considered to be a linear combination of statistically independent basis images, S, where A is an unknown mixing process. The basis images are estimated by the learned ICA output (independent components).

proposed by Lee and Seung [59]. A formal description of non-negative matrix decomposition as described in [59] follows:

Given an $n \times m$ data matrix V, NMF finds two non-negative matrices $W \in R^{nxr}$ and $H \in R^{rxm}$ such that $V \approx WH$. Here, each column of V represents an object, i.e. a human face or a small patch around a facial feature point. NMF approximates it by a linear combination of r basis vectors in columns of W and H contains the coefficients of the linear combination in its rows (also known as encoding vectors). The main difference of NMF from other factorization methods is the nonnegativity constraints both on W and H which allows only additive combinations. Due to the non-negativity constraints imposed in this method, NMF can be interpreted as a parts-based representation of the data, unlike other factorization methods, such as singular value decomposition (SVD) or independent component analysis (ICA-Architecture II), which are only capable of extracting holistic features from the data.

To find an approximate factorization $V \approx WH$, we first need to define cost functions that quantifies the quality of the approximation. Such a cost function can be constructed using some measure of distance between two non-negative matrices V and

WH. One useful measure is simply the square sum of the Euclidean distance between V and WH:

$$f(V, WH) = \|V - WH\|^2 = \sum_{i=1}^{n}\sum_{j=1}^{m}(V_{ij} - (WH)_{ij})^2 \tag{3.2}$$

Another useful measure is divergence of V from WH:

$$D(V, WH) = \sum_{i=1}^{n}\sum_{j=1}^{m}(V_{ij}\log\frac{V_{ij}}{WH_{ij}} - V_{ij} + WH_{ij}) \tag{3.3}$$

We now consider NMF as an optimization problem that is the minimization of the distance between V and WH with respect to W and H and subject to constraints $W, H \geq 0$. The most popular approach to solve Equation 3.2 or Equation 3.3 is the multiplicative update algorithm [59]. Basically, multiplicative update algorithm fixes one matrix and improves the other, namely, finds W^{k+1} such that $f(W^{k+1}, H^k) \leq f(W^k, H^k)$, and then H^{k+1} such that $f(W^{k+1}, H^{k+1}) \leq f(W^{k+1}, H^k)$. More explicitly, the algorithm using Euclidean distance can be derived as follows:

1. Initialize W and H with positive random numbers.
2. For $k = 1, 2, \ldots$

$$W_{ia}^{k+1} \longleftarrow W_{ia}^{k}\frac{(V(H^k)^T)_{ia}}{(W^k H^k (H^k)^T)_{ia}}, \forall i, a, \tag{3.4}$$

$$H_{bj}^{k+1} \longleftarrow H_{bj}^{k}\frac{((W^{k+1})^T V)_{bj}}{((W^{k+1})^T W^{k+1} H^k)_{bj}}, \forall b, j. \tag{3.5}$$

3. Repeat this process until convergence.

Since the function values $f(W^{k+1}, H^k)$ and $f(W^{k+1}, H^{k+1})$ are non-increasing after every update, the limit of the sequence $\{W^k, H^k\}_{k=1}^{\infty}$ is a stationary point [59]. However, this multiplicative update method still lacks optimization properties. For this reason, we employed projected gradient methods in our experiments [60]. Among existing methods, projected gradient method is simple and converges faster than the

multiplicative update approach.

In [60], Lin et al. introduced "sub-problem"; a collection of independent non-negative least squares problems, when one block of variables is fixed. Then, we refer Equation 3.6 or Equation 3.7 as sub-problem and the algorithm takes the form:

1. Initialize W and H with positive random numbers.
2. For $k = 1, 2, \ldots$

$$W^{k+1} = \arg\min_{W \geq 0} f(W, H^k) \tag{3.6}$$

$$H^{k+1} = \arg\min_{H \geq 0} f(W^{k+1}, H) \tag{3.7}$$

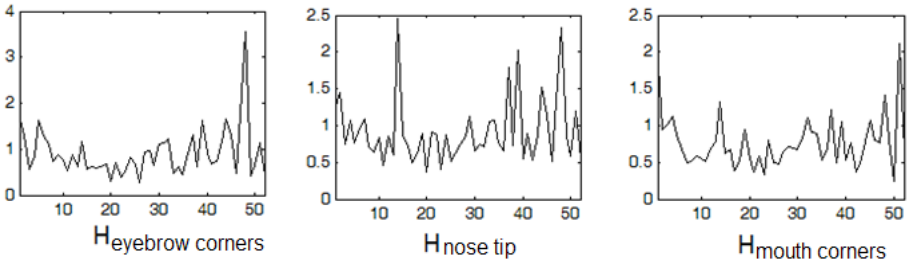
For example, Equation 3.7 consists of m independent non-negative least square problems:

$$h_j^{k+1} = \min_{h \geq 0} \|v - W^{k+1} h\|^2 \tag{3.8}$$

Solving sub-problems separately per iteration could be more expensive than the simple multiplicative update algorithm. Therefore, the proposed algorithm may be slower even though it decreases the function value better at each iteration. Efficient methods to solve sub-problems are thus essential. In [60], the authors make the algorithm fasten by a serial setting, namely treating non-negative least square problems together and employing projected gradient at each sub-iteration.

In Figure 3.5-(a), the idea behind NMF is illustrated as a facial feature extraction method. In the training stage, NMF approximately factorized the data matrix, V which is composed of different feature types, into matrices W and H. The rows of H, namely, encoding vectors construct the feature vectors of each fiducial point. Once we obtain W, the testing stage is straightforward. For any unknown image block, feature vector is obtained as $h_{test} = W^{\dagger} * v_{test}$. In Figure 3.5-(b), we give some examples of encoding vectors correspond to different type of features. As shown, various facial features result in different encoding vectors. In our experiments, as in ICA case, we use 52 factors at

(a)



(b)

Figure 3.5. Results of NMF algorithm applied to facial features: (a) Non-negative matrix factorization, (b) Encodings vectors corresponding to each feature type.

the coarse level.

3.1.4. Gabor Wavelet Transform

Gabor wavelets represent local characteristics and are robust against luminance variations in the image. They can be obtained as a convolution with a family of Gabor kernels

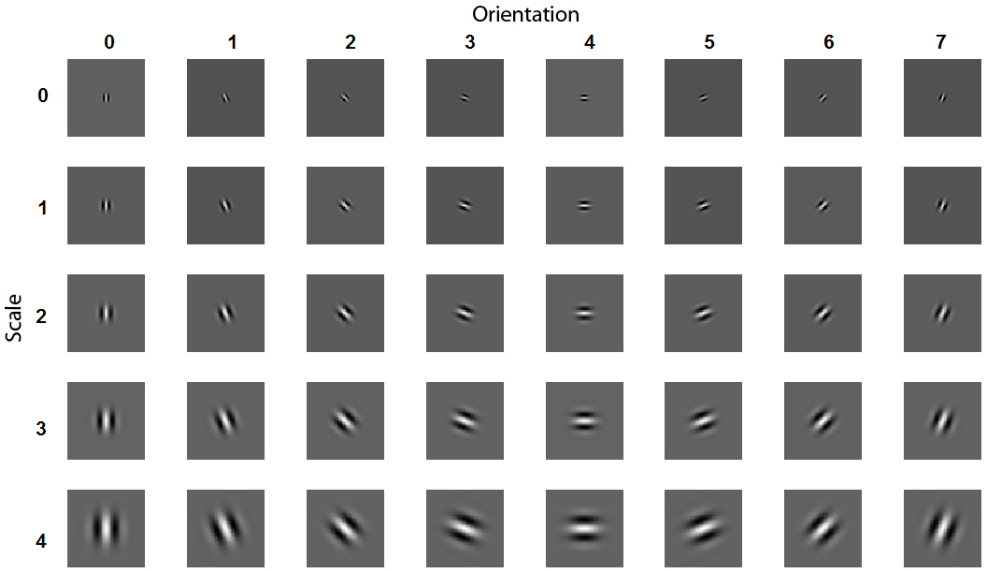
$$W_j(\overrightarrow{x}) = \frac{k_j^2}{\sigma^2} exp(-\frac{k_j^2 x^2}{2\sigma^2})[exp(i\overrightarrow{k_j}\overrightarrow{x}) - exp(-\frac{\sigma^2}{2})] \tag{3.9}$$

Here, the extent of the plane waves with wave vector $\overrightarrow{k_j}$ at $\overrightarrow{x} = (x, y)$ is restricted by a Gaussian envelope function

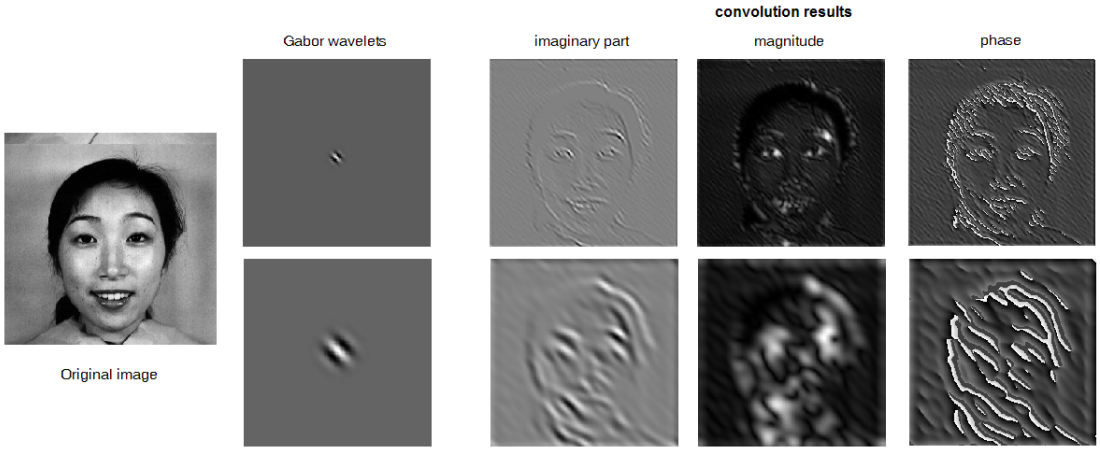
$$\overrightarrow{k_j} = \begin{pmatrix} k_{jx} \\ k_{jy} \end{pmatrix} = \begin{pmatrix} k_v cos\phi_\mu \\ k_v sin\phi_\mu \end{pmatrix} \tag{3.10}$$

where $k_v = 2^{-\frac{v+2}{2}}\pi, \phi_\mu = \mu\frac{\pi}{8}$ with $j = \mu + 8v$. The width of the Gaussian is controlled by the parameter $\sigma = 2\pi$. By changing the frequency v and orientation μ parameters, one can match and hence locate patterns having similar scales and orientations. Figure 3.6 shows the wavelet filters for different frequency and orientation parameters and a convolution example of an image with proposed filters.

In our experiments, we employed three frequencies $v = 0, 1, 2$ and four orientations $\mu = 0, 2, 4, 6$, resulting in 12 Gabor masks [23]. Gabor feature vector is obtained by cropping 8×8 patches around each search point and convolving these patches with proposed filters. Then, PCA is used to reduce this $8 \times 8 \times 12$-dimensional feature vector to 100-dimensional feature vector.



(a)



(b)

Figure 3.6. Feature extraction by Gabor wavelet decomposition: (a) The real part of the Gabor filter with 5 frequencies and 8 orientations [7], (b) Wavelet convolution example.

4. DECISION FUSION AND REFINEMENT

The extracted features from patches classified by a binary Support Vector Machine (SVM) classifier, trained for that specific landmark, to be a possible landmark point or not. This is repeated for each patch: for example, in the coarse search for eye corners we consider n patches in the upper left quarter of the face. We rank the positive outcomes from the SVM and then we can pick the highest ranking patch.

However, the highest ranking patch might be misleading under some face scenarios, such as extreme facial expressions, different illumination conditions. This causes some problems in SVM-based techniques. First, the detection result should be post-processed to obtain a localizing results, e.g., there might be lots of detected points as the candidate facial features and selecting the highest ranking patch sometimes will be erroneous or there are no detection results at all. Second, if the facial features support is large, the localization result could not be precise while if the support is small, there would be many false positive detection results and therefore it is not robust. Alternatively, we propose the following algorithm to overcome these difficulties:

4.1. Elimination of the Irrelevant Features

If there is one or more low-reliability landmarks, we propose to eliminate these irrelevant ones by fusing the outcomes of individual feature extraction methods. This can be accomplished in two ways: *Feature Fusion* and *Score Fusion*. It is also possible to determine a set of reliable landmark points based on the anthropometrical information as in Probabilistic Graphical Model (PGM) [28]. In this work, we extended PGM to cover 12 fiducial points in question.

4.1.1. Fusion of Facial Feature Extraction Methods

4.1.1.1. Feature Fusion. In this scheme, for each sample, the corresponding features obtained from each channel, namely DCT, GWT, ICA and NMF, are concatenated

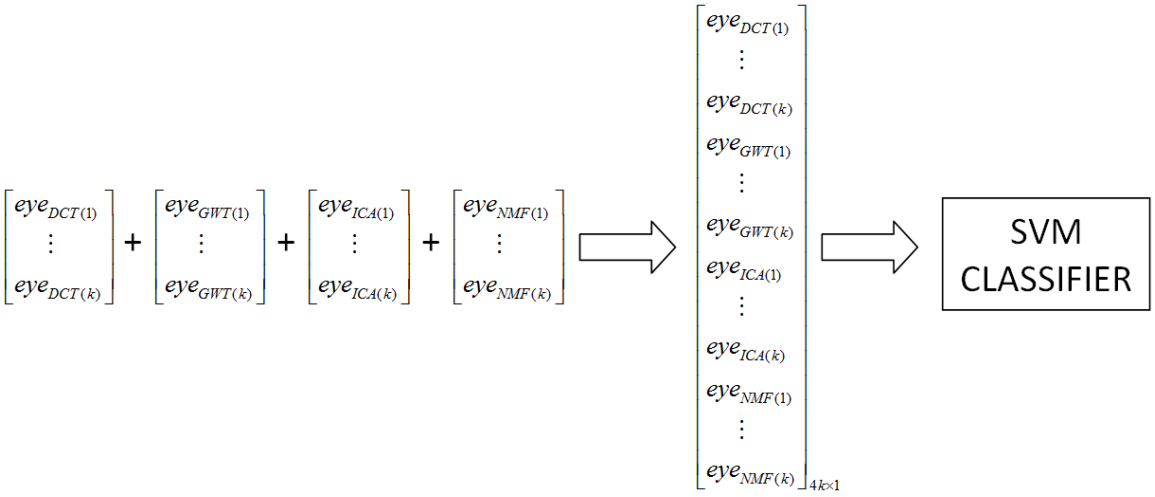


Figure 4.1. Illustration of feature fusion method for a single eye sample.

into a single vector. If we keep k coefficients for each feature type, the resulting feature vector will be $k \times 4$ dimensional. In our experiments, we take $k = 52$; thus the fused vector is 208 in length. After having normalized the coefficients, the SVM classifiers are trained by using these fused feature vectors. Given a probe image, the individual feature vectors are fused in the same way for each search point and, then used to test the classifiers. Final decision is achieved by simply selecting the highest scoring candidate. The feature fusion technique for only a single training sample is illustrated in Figure 4.1.

4.1.1.2. Fusion by Weighted Median. Instead of immediately deciding upon only one candidate, we can select the top L highest scoring patches, or the p percent of highest scoring patches. For this purpose, we collected several candidate points outputted from each feature channel with a normalized SVM score higher than a threshold predetermined during a training session. Then, we fuse these multiple candidates using a weighted median filter applied on their spatial locations. Before median filtering, we re-normalized the SVM scores of the candidates by min-max method. More explicitly, give an original score w, the normalized score $\hat{w}$ is computed by $\hat{w} = \frac{w-min}{max-min}$ where minimum and maximum scores are obtained from the data. These normalized SVM scores, $\hat{w} \in [0, 1]$, constitutes the weights in median filtering. Given the spatial locations of the points, (x_i, y_i), $i = 1, \ldots, N$ where N is the number of points, the weighted

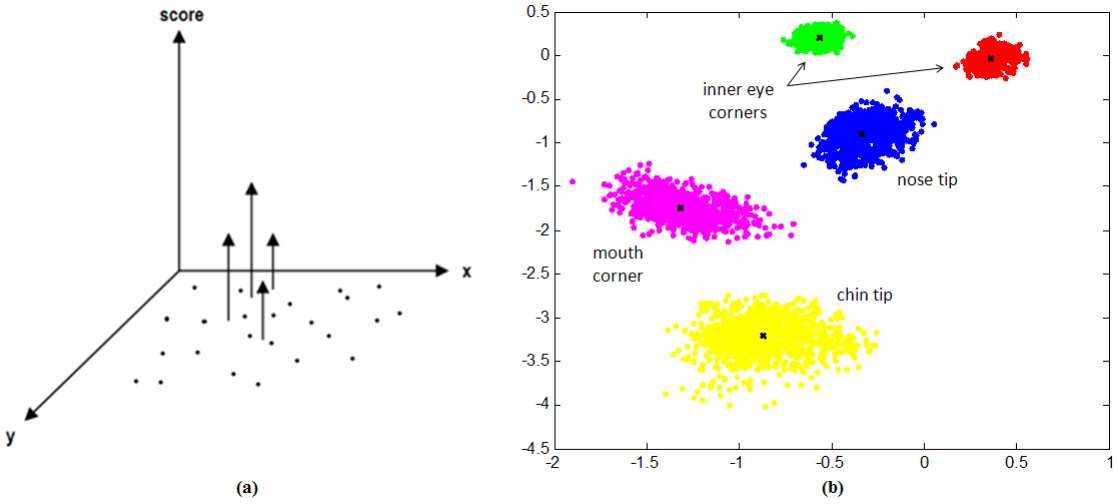


Figure 4.2. (a) Illustration of score fusion based on weighted median filter, (b) The spatial distribution of missing landmarks estimated with respect to the reliable landmarks.

median filter is applied as follows:

$$\begin{bmatrix} x_{median} \\ y_{median} \end{bmatrix} = \text{vector median}\left(\begin{bmatrix} x_1 & \dots & x_1 \\ y_1 & \dots & y_1 \end{bmatrix}_{2\times n_1} \dots \begin{bmatrix} x_N & \dots & x_N \\ y_N & \dots & y_N \end{bmatrix}_{2\times n_N}\right) \tag{4.1}$$

Here, n_i, representing the number of repetitions of each candidate point, is assigned with respect to $\hat{w}_i$. For example, if $\hat{w}_i > 0.9$, n_i is equal to 3. The score fusion method is illustrated in Figure 4.2-(a). If our search over the entire region does not yield any candidate points for a specific landmark we label this landmark as missing, and we leave its recovery to post-processing via the graph-based structural completion (see Section 4.2). An example of score fusion is given in Figure 4.3.

4.1.2. Probabilistic Graphical Model

Probabilistic Graphical Model (PGM) is another approach to attain the reliable landmark points [28]. Briefly, PGM is based on statistical inter-landmark distance and angle information. Such anthropometrical information is then used to tune a non-rigid lattice. The lattice nodes correspond to facial landmarks and the spring forces on the

(a) (b) (c)

Figure 4.3. The landmarking results of the proposed algorithm: (a) Candidates populated from each feature channel, (b) Reliable points selected by score fusion, (c) Graph completion.

cords and angles joining the vertices are assigned as Gaussian distributions, with mean and variance estimated based on a training set. Given the information of mean length $\overline{\lambda}_{i,j}$, mean relative angle $\overline{\phi}_{i,j}$ between the feature points and their respective variances $\Lambda^2_{i,j}$, $\Phi^2_{i,j}$, for a *k-landmark* configuration, the energy is calculated as:

$$E = \sum_{j=1}^{k} \frac{(\lambda_{i,j} - \overline{\lambda}_{i,j})^2}{\Lambda^2_{i,j}} + \sum_{l=1}^{k} \frac{(\phi_{i,l} - \overline{\phi}_{i,l})^2}{\Phi^2_{i,l}}; \quad i = 1, \ldots, \begin{pmatrix} 12 \\ k \end{pmatrix} \tag{4.2}$$

For each possible configuration, totally $combination(n; k)$ (i.e., $n = 12$) configurations in a *k-landmark* case, the energy is evaluated and the configuration having the minimum energy is assigned as the most reliable set of landmark points, see Figure 4.4. Accordingly, these reliably estimated landmarks are used to adapt a generic landmark graph to the actual face as in Section 4.2. The missing landmarks are simply read off from the adapted graph; but the lengths and angles of the cords of these landmarks to be estimated are set to their average value. In this work, we have extended this work to cover 12 fiducial points (corners of eye, eyebrows, mouth, nose and chin tip).

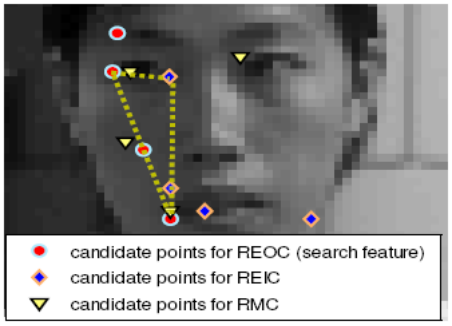


Figure 4.4. A plausible landmark triple of the right eye outer corner (REOC), right eye inner corner (REIC) and right mouth corner (RMC).

4.2. Structural Completion

The structural completion method serves the purposes of recovering the missing landmarks [23]. In this method, we define a graph whose nodes coincide with the 12 landmark points. The arcs between the nodes are modeled as Gaussian spring forces with means and variances learnt during training phase. Thus, for example, the left outer eye corner is tied tightly to left eye inner corner, but more loosely, to right eye corners, to mouth corners and to nose. In fact, the more the anthropometric variability in the training database, the larger the corresponding variance and the looser the bond.

In this thesis, we used the graph-aided structural completion method in the following manner. Once we obtain a subset of located landmarks, namely reliable landmarks via multi-attribute framework or probabilistic graphical model, we recover the remaining ones by structural completion method. Here, we refer to these reliable points as support set. If we have a support set size of k and totally n landmarks, the number of possible support sets is $C(n;k)$, where $C(.)$ is the combination operator. For example, if the size of a support set is four ($k = 4$), k-combination of $n = 12$ landmarks results in 495 combinations. In training stage, first each support set is normalized; the normalization process involves three steps:

- Translation: The centroid of the landmark points in the support set is moved to the origin.
- Scaling: The average distance of the landmarks points to the origin is set to $\sqrt{2}$.

- Rotation: The first landmark in the support set is aligned with respect to the y-axis.

Then, from a set of training samples, we learn the spatial distribution of the normalized landmarks for each possible support set, as each support set exhibits a different normalization. For each combination, the remaining landmark positions are normalized and their distribution is modeled by a mixture of Gaussians. In the testing stage, we find the missing landmarks by back-projection of the expected landmark location. First, we establish and normalize the support set, then the missing features are simply estimated using the means of the respective spatial distributions learned during the training stage. These center points in the normalized coordinates are then subjected to the inverse normalization process to obtain their actual coordinates.

In Fig. 4.2-(b), we give an example scatter plot of the estimated locations of the missing landmark points based on seven reliable landmark points (eyebrow corners, outer eye corners and left mouth corner). In our proposed scheme, unlike PGM, there is no constraint on the size of the support set; we construct a separate graph model for each different size of the support set in the training session.

4.2.1. Refinement

The structural completion step is followed by a refinement stage in which each coarse-level landmark is transferred to the full-resolution image (i.e., 640×480) and refined by searching for a better fit around the coarsely estimated points. In our experiments, for fine level, we centered 20×20 windows at the coarse localization points and trained SVM classifiers using only DCT-based feature descriptors. A similar search method as in the coarse level is adopted; but final decision is achieved by choosing the point with highest score as the fine location of the searched fiducial point.

4.3. Elastic Bunch Graph Matching

For comparing our proposed method with a global approach, we implemented Wiskott's Elastic Bunch Graph Matching (EBGM) algorithm in which faces are represented as graphs, with nodes positioned at fiducial points (eyes, nose...) and edges labeled with 2-D distance vectors [7]. Each node contains a set of 40 complex Gabor wavelet coefficients at different scales and orientations (phase, amplitude); called as a "jet". Face recognition is based on comparing labeled graphs. The details of the EBGM algorithm is as follows.

4.3.1. Preprocessing using Gabor Wavelets

The first step of the EBGM algorithm is convolving the face image with Gabor wavelet masks as stated in Section 3.1.4. The set of complex convolution coefficients for kernels of different orientations and frequencies at one image pixel is called as a jet. Then, a jet J can be described in polar coordinates

$$J = aexp(i\phi) \tag{4.3}$$

where a is the magnitude and ϕ is the phase angle of the coefficients. a and ϕ can be found by using the equations:

$$a = \sqrt{a_{real}^2 + a_{imag}^2} \tag{4.4}$$

$$\phi = \begin{cases} \arctan\{a_{imag}/a_{real}\} & \text{if } a_{real} > 0 \\ \pi + \arctan\{a_{imag}/a_{real}\} & \text{if } a_{real} < 0 \\ \pi/2 & \text{if } a_{real} = 0 \text{ and } a_{imag} \geq 0 \\ -\pi/2 & \text{if } a_{real} = 0 \text{ and } a_{imag} < 0 \end{cases} \tag{4.5}$$

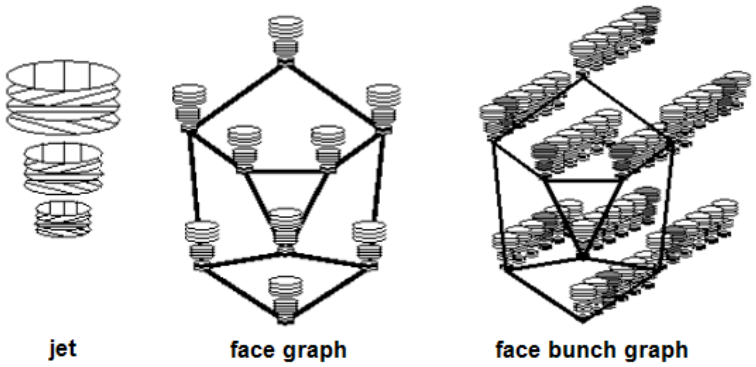


Figure 4.5. The set of n coefficients obtained for one image point is referred to as a jet. The collection of jets corresponding to different landmark locations constitutes a face graph. Finally, a face bunch graph is a stack-like structure that combines graphs of individual sample faces.

In [7], the face image is convolved with 40 Gabor wavelet filters (5 frequencies and 8 orientations) as shown in Figure 3.6-(a), and the set of 40 coefficients obtained for one image point form a jet. The jets are extracted from images with manually selected landmark locations, and then collected in a data structure called as a face graph. In Figure 4.5, an example of face graph is illustrated: it is composed of 9 nodes and in each node, 3 frequencies and 4 orientations are used.

Measuring the similarity for two jets is fundamental in landmark localization and face graph similarity evaluation. There are two similarity function defined in [7]: magnitude similarity and phase similarity. With a jet J taken at a fixed landmark position and $J' = J'(\overrightarrow{x})$ taken from a search point $\overrightarrow{x}$, the magnitude similarity S_a is based on the covariance of two magnitudes

$$S_a(J, J') = \frac{\sum_{j=1}^{N} a_j a'_j}{\sqrt{\sum_{j=1}^{N} a_j^2 \sum_{j=1}^{N} a_j'^2}} \tag{4.6}$$

where N is the number of wavelet coefficients in the jet. This measure only compute the similarity of the energy of the frequencies; the phase information is not used. Since the similarity is completely unaffected by phase, this measure is tolerant to small displacements; it can be easily confused and may respond to an incorrect spatial

feature.

Patterns with similar magnitudes can be discriminated by using the phase information and the accurate jet localization in an image can also be realized because phase varies so quickly with location. Assuming that two jets J and J' refer to locations with small relative distance $\overrightarrow{d}$, the phase similarity function S_ϕ is defined as

$$S_\phi(J, J') = \frac{\sum_{j=1}^{N} a_j a'_j cos(\phi_j - \phi'_j - \overrightarrow{d}\,\overrightarrow{k}_j)}{\sqrt{\sum_{j=1}^{N} a_j^2 \sum_{j=1}^{N} a_j'^2}}. \tag{4.7}$$

This measure is based on both the magnitude and phase components of the coefficients and can approximately compensate for the phase shifts by the term $\overrightarrow{d}\,\overrightarrow{k}_j$. Unfortunately the vector $\overrightarrow{d}$ is undefined. The displacement $\overrightarrow{d}$ can be estimated by disparity estimation method which is maximization of the similarity S_ϕ in its Taylor expansion:

$$S_\phi(J, J') = \frac{\sum_{j=1}^{N} a_j a'_j [1 - 0.5(\phi_j - \phi'_j - \overrightarrow{d}\,\overrightarrow{k}_j)^2]}{\sqrt{\sum_{j=1}^{N} a_j^2 \sum_{j=1}^{N} a_j'^2}} \tag{4.8}$$

By setting $\frac{\partial}{\partial d_x} S_\phi = \frac{\partial}{\partial d_y} S_\phi = 0$ and solving for $\overrightarrow{d}$ yield

$$\overrightarrow{d}\,(J, J') = \begin{bmatrix} d_x \\ d_y \end{bmatrix} = \begin{bmatrix} \Gamma_{xx} & \Gamma_{yx} \\ \Gamma_{xy} & \Gamma_{yy} \end{bmatrix}^{-1} \begin{bmatrix} \Phi_x \\ \Phi_y \end{bmatrix} \tag{4.9}$$

if the determinant $\Gamma_{xx}\Gamma_{yy} - \Gamma_{xy}\Gamma_{yx}$ does not vanish, with

$$\Phi_x = \sum_{j}^{N} a_j a'_j k_{jx} (\phi_j - \phi'_j) \tag{4.10}$$

$$\Gamma_{xy} = \sum_{j}^{N} a_j a'_j k_{jx} k_{jy} \tag{4.11}$$

and Φ_y, Γ_{xx}, Γ_{yx}, Γ_{yy} defined correspondingly. In this function, the phase differences may exist the range of $\pm\pi$, we need to correct it by $\pm 2\pi$. The displacement can be estimated between two jets when they close enough that their Gabor kernels are highly overlapping.

Moreover, the range for displacement estimation varies with the frequency of the kernel. The equation S_ϕ can determine displacements up to half the wavelength of the highest frequency kernel. The estimated range can be increased by using low frequency kernels only. The number of frequency levels used for the first displacement estimation is referred to as focus. A focus of 1 mean that only the lowest frequency level is used and the estimated range may be up to 8 pixels. In other word, a focus of 5 means that all five levels are consecutively used. For each higher level, the phases of the higher frequency coefficients have to be corrected by multiples of 2π to match as closely as possible the expected phase differences inferred from the displacement estimated on the lower frequency level. The accurate position of the jets can be found by this iterative refinement process.

4.3.2. Face Representation

To represent a single face, labeled graphs are introduced via a set fiducial points, e.g. the pupils, the eyebrows, the mouth corners, the nose tip, etc. A labeled graph G consists of N nodes at the positions of these fiducial points, $\overrightarrow{x}$, $n = 1, \ldots N$ and E edges between them. Each node is labeled with its corresponding jet J_n and the edges are labeled with distances $\Delta\overrightarrow{x}_e = \overrightarrow{x}_n - \overrightarrow{x}_{n'}$, $e = 1, \ldots, E$, where edge e connects node n with n'. This face graph is object-adapted, since the nodes are selected from face-specific points. Although all nodes refer to the same fiducial points, the distances may vary due to rotations in depth.

Automatic detection of fiducial points in a probe image needs a general representation rather than an individual face graph. The wide range of possible variations in the appearance of faces, like differently shaped facial components, facial hair, variations due to sex, age, eyeglasses, facial expressions (i.e., closed eye, open mouth) etc., should

be covered. Representing each feature with a separate graph will not be efficient, instead we utilize a Face Bunch Graph (FBG) that combines graphs of individual sample faces, see Figure 4.5.

While forming a FBG, it is crucial that all individual graphs have the same grid structure and the nodes refer the same fiducial points. A set of jets referring to the same fiducial point is called as a bunch. During the localization of each fiducial point in a probe face image, the best fitting jet, called as local expert, is selected from the bunch by a matching procedure (described in the next section). Assume for a particular pose that there are M model graphs G^{Bm} of identical structure, taken from different individual samples. The corresponding FBG, B is then a stack-like structure; its nodes are labeled with bunches of jets J_n^{Bm} and its edges are labeled with average distances $\Delta\overrightarrow{x}_e^B = \sum_m \Delta\overrightarrow{x}_e^{Bm}/M$.

The desired localization accuracy of fiducial points in the probe image can be achieved by increasing the number of model graphs in the FBG. However, it requires too much computational effort to match such a FBG, hence the model graphs should be as different as possible to reduce the redundancy and maximize variability. In Figure 4.5, an example of FBG is illustrated; each of the nine nodes is labeled with a bunch of six jets, resulting in $9 * 6! = 6480$ different faces. There are two types of FBGs defined in [7]: one is for normalization stage (face detection), referred to as coarse FBG, composed of 30 individual graphs and second one, fine FBG is for final graph extraction (refinement of the positions of the fiducial points), composed of 70 individuals. These sizes are seemed to be sufficient for a compromise between reliable localization of landmark points and computational burden.

4.3.3. Generating Face Representations by Graph Matching

In this section, we will discuss the procedure for generating a face image graph. To form the FBG, model graphs are constructed with manually marked points. Then, FBG is used to automatically extract the face graph of a test image by Elastic Bunch Graph Matching algorithm.

Manual Definition of Graphs: The manual definition of the graphs is composed of three steps: (1) Mark a set of fiducial points for a given image; (2) Draw the edges between fiducial points and compute edge labels as the differences between node positions; (3) Use Gabor wavelet transform to compute the jets at the nodes. The set of fiducial points should cover the face symmetrically. For face detection, we need to use more nodes on the outline of the face. In refinement stage, we place more nodes at the center of the face as the central features are more important for face recognition.

The Graph Similarity Function: The graph similarity between an image graph and the FBG is based on the jet similarity and the distortion of the image grid relative to the FBG grid. The similarity function is

$$S_B(G^I, B) = \frac{1}{N}\sum_n \max_m(S_\phi(J_n^I, J_n^{Bm})) - \frac{\lambda}{E}\sum_e \frac{(\Delta\overrightarrow{x}_e^I - \Delta\overrightarrow{x}_e^B)^2}{(\Delta\overrightarrow{x}_e^B)^2} \tag{4.12}$$

where G^I is an image graph with node $n = 1, \ldots, N$, edges $e = 1, \ldots, E$, FBG B with M model graphs $m = 1, \ldots, M$ and λ determines the relative importance of jets and metric structure. Here, the first term is for feature (jet) comparison and the second term for metric comparison (distortions).

Matching Procedure: The matching procedure is based on the maximizing the similarity with the FBG defined in Equation 4.12. The proposed algorithm is as follows:

1. Find approximate face position: Condense the FBG into an average graph by taking the average magnitudes of the jets in each bunch (or select one arbitrary graph as a representative). Fit this condensed FBG by evaluating similarity at each location of a square lattice with a spacing 4 pixels. The similarity is computed by using a similarity function based on magnitude of jets (the phase of the jets and the distortion of the image grid relative to the FBG grid are not considered). Repeat the scanning process until finding the the best fitting position with a spacing of 1 pixel.
2. Refine position and size: Use FBG without averaging, varying it in position and

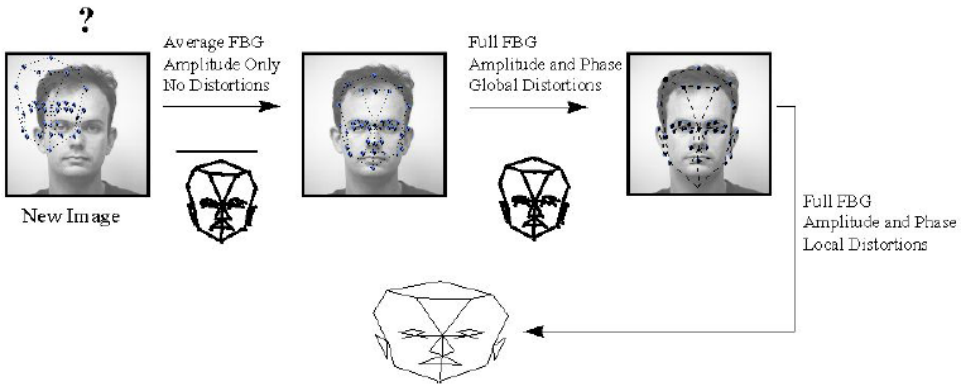


Figure 4.6. EBGM procedure.

size: four different positions $(\pm 3, \pm 3)$ pixels displaced from the position previously found and at each position check two different sizes which same center position, a factor of 1.18 smaller or larger than the FBG average size. This is done with a focus of 1, the displacements may be of a magnitude up to eight pixels. The grids are then rescaled and repositioned to minimize the square sum over the displacements. Keep best of the eight variations as the starting point for the next step.

3. Refine size and find aspect ratio: A relaxation process is applied to x and y-dimensional independently. The focus is increased from 1 to 5.
4. Local distortion: In a pseudo-random sequence the position of each individual image node is varied to further increase the similarity to the FBG. In this step, the distortion of grids are also considered and only positions for which the estimated displacement vector is small ($d < 1$). For this local distortion the focus again increases from 1 to 5.

The matching algorithm is illustrated in Figure 4.6. The resulting graph is called as the image graph and stored as a representation of the probe image. In this thesis, we adapted the EBGM algorithm for 12 fiducial points (eye brow and eye corners, nose, mouth corners and chin); the locations of the nodes are considered as estimated landmark points.

5. EXPERIMENTAL RESULTS

In this chapter, the performance of the proposed facial feature extraction schemes are presented. Firstly, the utilized images databases and preprocessing steps are explained. Secondly, four different feature extraction algorithms (DCT, GWT, ICA and NMF), fusion schemes and graph-aided method are tested individually, and their performances are compared relative to different landmark points and changes in facial expression, lighting and different databases. Then, the results observed for each of the schemes are discussed separately.

5.1. Experimental Setup

5.1.1. Utilized Databases

In this study, we performed our experiments on three different databases. In these data sets, we only consider frontal views with different facial expressions, illumination conditions, facial hair, eye glasses or instinctive rotations.

5.1.1.1. Bosphorus Database. This challenging 2D and 3D facial image database includes various poses, expressions and occlusion conditions [61]. There are 60 men and 45 women, totally 105 subjects, and most of the subjects are Caucasian. The database has two sessions. The first session includes 34 subjects with 10 expressions, 13 poses, four occlusions and four neutral poses, resulting in a total of 31 scans per subjects. The second session is more comprehensive, including 71 subjects and per each subject 54 scans which are 34 expressions, 13 poses, four occlusions and one or two neutral poses. There a total of 4652 face scans. In our experiments, we consider a subset of 31 different facial expressions, slight head rotations (smaller than 10^o), occlusions (eye glasses, hand and hair) and neutral poses common to the 81 subjects. Some example images are shown in Figure 5.1. The data are split into two disjoint parts including non-overlapping subjects; training set (1048 samples) and test set (1186 samples). If

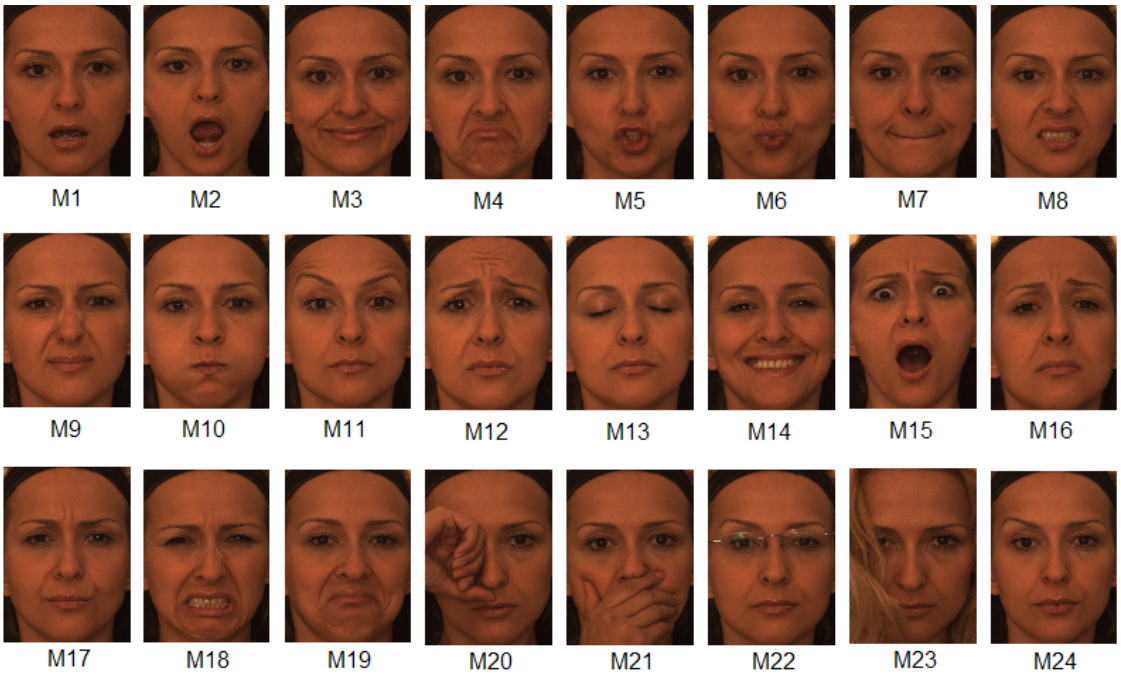


Figure 5.1. Poses and expressions used in the experiments. Facial expressions; (M1) Jaw drop, (M2) Mouth stretch, (M3) Lip corner puller, (M4) Chin raiser, (M5) Lip funneler, (M6) Lip puckerer, (M7) Lip suck, (M8) Upper lip raiser, (M9) Nose wrinkler, (M10) Cheek puff, (M11) Outer brow raiser, (M12) Inner brow raiser, (M13) Eyes closed, (M14) Happiness, (M15) Fear, (M16) Sadness, (M17) Anger, (M18) Disgust, (M19) Jaw drop + Low intensity lip corner puller; Occlusion: (M20) Eye occlusion, (M21) Mouth occlusion, (M22) Eye glasses, (M23) Hair and (M24) Neutral pose.

needed, the half of the training set is used as validation set.

5.1.1.2. BioID Database. BioID database consists of 1521 gray level images with a resolution of 384x286 pixel [62]. Each one shows the frontal view of a face of one out of 23 different test persons and was recorded during several sessions in uncontrolled conditions using a web camera within an office environment. Compared to the other databases, this data set features a larger variety of illumination conditions, backgrounds and face size. Some examples are given in Figure 5.2. We have manually picked 14 subjects (847 samples) as our training set and 9 subjects (674 samples) for testing.

Figure 5.2. Sample images from BioID Database: The images are acquired under less controlled conditions, lower resolution and often with illumination effects.

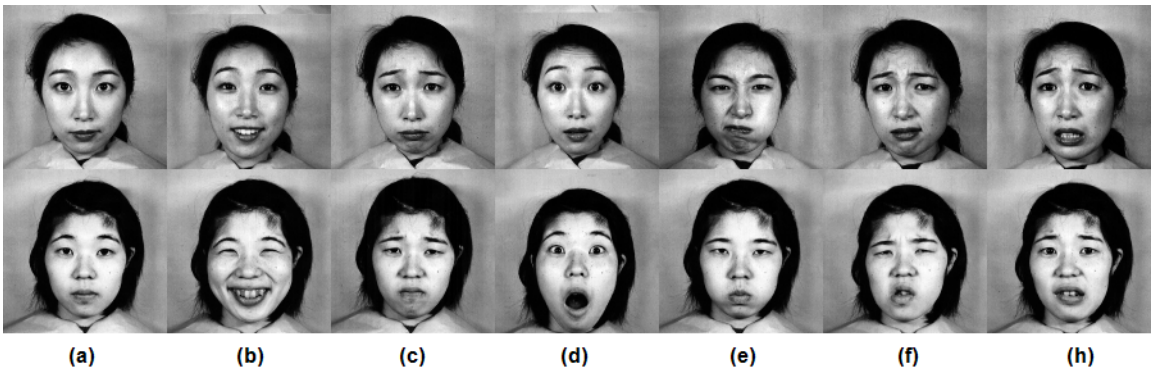


Figure 5.3. Sample images from the JAFFE database: (a) Neutral, (b) Happy, (c) Sad, (d) Surprise, (e)Angry, (f) Disgust, (h) Fear.

5.1.1.3. JAFFE Database. The Japanese Female Facial Expression (JAFFE) database is composed of facial expression images [63]. The database contains 213 images of 7 facial expressions (six basic facial expressions plus one neutral) posed by 10 Japanese female models. During acquisition, lights were positioned to create homogeneous illumination on the face. Sample images are given in Figure 5.3.

We summarized content of the databases, the number of subjects and samples images per subject used in the experiments in Table 5.1.

5.1.2. Preprocessing

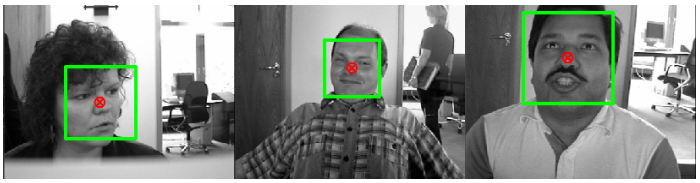
5.1.2.1. Face and Region of Interest Detection. Facial feature localization mostly starts with face detection. Face detection is a problem on its own on which significant re-

Table 5.1. The properties of sample images used in the experiments.

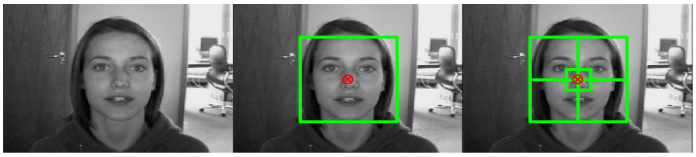
	Bosphorus	BioID	JAFFE
Number of subjects	81	23	10
Samples per subject	15-17	varying	20-23
# train samples	1048	847	148
# test samples	1186	674	65
Total	2234	1521	213
Expression	31 expressions (25 action units & 6 emotions)	Unspecified	6 emotions
Pose	Spontaneous	Uncontrolled	Spontaneous
Occlusion	3 occlusions (eye glasses, hand & hair)	Uncontrolled (eye glasses & hand)	NA

search has been conducted. Face detection is out of scope of this thesis; we employed a MATLAB version of the ground-breaking Viola-Jones face detector algorithm [48]. This algorithm is available in open-source format as part of the openCV library and an implemented MATLAB version is available in [64]. In Figure 5.4-(a), we present some results of the face detector algorithm. It works very well even if the background is complex or the head is rotated slightly.

To do a more cost-effective subsequent landmarking, we limit the face region within which a fiducial point is expected to be found. Since the face detector yields a square enclosing the face area in the image, we segment the face into six regions (NE,NW,SE,SW,S and centrum) with respect to the center of the square as illustrated in Figure 5.4-(b). For example, the left eye and left eyebrow corners are searched in the upper left quarter of the square (NE); nose tip is searched in a $k \times k$ (i.e., 20×20) window defined at the center of the square. This method can work only on frontal or slightly rotated faces. There are also more sophisticated methods for this problem, i.e., using color information, orientation histograms or component based classifiers as in [65]. Since we are working only on frontal views, we used this method in order to avoid computational burden.



(a)



(b)

Figure 5.4. (a) Results of face detection algorithm, (b) An example of face segmentation.

5.1.3. Training Stage

In the training stage, we have trained separate Support Vector Machine (SVM) classifiers for each facial feature type; one for outer eye corners, one for inner eye corners, one for mouth corners, one for the tip of the nose, etc. To construct the training set, we first crop image patches around manually marked fiducial points to obtain feature vectors (positive samples) and randomly selected points that do not contain fiducial points (negative samples). For any facial feature, negative samples also include all the remaining facial fiducial points, i.e., if we are training a SVM classifier for the outer eye corner, the negative samples are composed of randomly selected points plus eyebrow corners, inner eye corners, nose tip, etc. To increase the number of the training samples and reduce the computational complexity, right eyebrow, eye and mouth-corners are mirrored to generate their left corner images and the SVM classifiers are trained only for the left side of the face. This process is repeated for each feature category (DCT, ICA, GF and NMF). We have also used bootstrapping method for improved learning. In bootstrapping, the SVM classifiers are re-trained on a validation set with false positives (FPs) in order to pick meaningful negative samples. In Table 5.2, we give the number of samples needed for building a local feature detector.

Table 5.2. The number of training samples used in the experiments.

	Training stage		Boosting stage	
	Positive samples	Negative samples	Positive samples	Negative samples
Outer eyebrow corners	6417	37191	8631	53954
Inner eyebrow corners	8020	35588	10480	47850
Outer eye corners	8018	35590	10478	49830
Inner eye corners	8020	35588	10480	50698
Mouth corners	8020	14227	10480	27123
Nose tip	4010	25163	5240	33489
Chin tip	1094	21153	1531	32670

5.1.4. Support Vector Machines

We used as classifier Binary Support Vector Machines (SVM) of RBF kernel variety. In our experiments, we have used LIBSVM (library for support vector machines) package [66]. The formulation is as follows.

Given two classes of data, for observed feature vectors $x_i \in R^n$, i.e., $n = 52$, $i = 1, \ldots, l$, and their corresponding class labels $y \in R^l$ such that $y_i \in \{-1, 1\}$, the goal is to solve the following primal problem:

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i \\ \text{subject to} \quad & y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, \quad i = 1, \ldots, l \end{aligned} \tag{5.1}$$

Its dual is

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2} \alpha^T Q \alpha - e^T \alpha \\ \text{subject to} \quad & y^T \alpha = 0, \\ & 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, l \end{aligned} \tag{5.2}$$

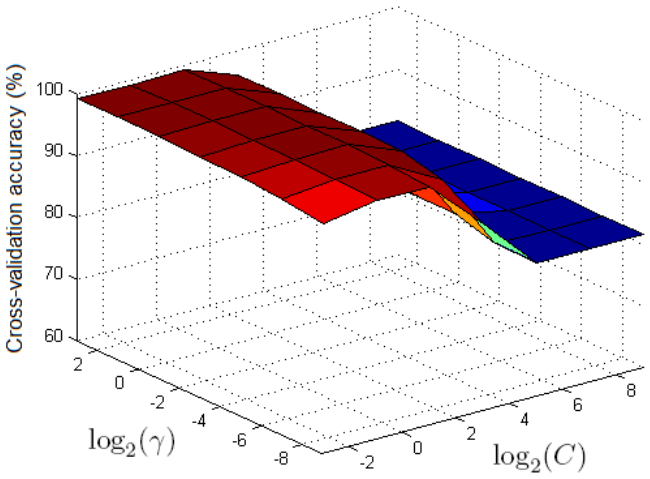


Figure 5.5. Cross-validation results for determining the parameters of SVM classification.

where e is the vector of all ones, $C > 0$ is the upper bound, Q is an $l \times l$ positive semidefinite matrix, $Q_{ij} \equiv y_i y_j K(x_i, x_j)$, and $K(x_i, x_j) \equiv \phi(x_i)^T \phi(x_j)$ is the RBF kernel. Here feature vectors are mapped into a higher dimensional space by the function ϕ. The decision function is

$$sgn\left(\sum_{i=1}^{l} y_i \alpha_i K(x_i, x) + b \right) \tag{5.3}$$

In order to set the SVM parameters we evaluated the performance over the validation set for different kernel parameters γ and cost parameters C by scanning over the ranges $\gamma = [2^3, 2^2, 2^1, \ldots, 2^{-15}]$ and $C = [2^{15}, 2^{14}, 2^{13}, \ldots, 2^{-5}]$. Figure 5.5 shows an example of the cross-validation results in determining the two parameters. In the final analysis, we have selected a separate parameter pair for each fiducial point and feature type.

5.2. Evaluation Experiments

Given a test image, we consecutively extract the proposed features from overlapping blocks in each region and run binary SVM classifiers over these regions. For each candidate point, SVM classifier produces a score which can then be used in final decision or fusion schemes.

The performance of the feature localizers is evaluated by computing Euclidean distance (in terms of pixels) between the estimated point and its manually marked reference point. In our experiments, an estimated point is considered as correctly detected if its distance from the reference point is less than 10 % of the inter-ocular distance (acceptance threshold = $0.1 \times$ IOD).

5.2.1. Comparison of Individual Feature Extraction Methods

In this part of experiments, we extract feature vectors for each method (DCT, GWT, ICA and NMF), sort candidate points relative to their respective SVM scores, and we select the highest ranking candidate point as our detected landmark. In Table 5.3, the coarse level performance (image resolution = 80×60, block size = 8×8) of the proposed methods is given with respect to each landmark point. The SVM classifiers are trained with Bosphorous database and then they are tested on unseen portion of the Bosphorous database. The accuracy of the feature localizer is expressed as the percentage number of correctly detected points.

Table 5.3. Performance comparison of individual feature types (%). Feature size = 52. Acceptance threshold = 0.1.

	DCT	ICA	NMF	GWT
Outer eyebrow corners	**79.8**	78.9	77.2	76.7
Inner eyebrow corners	**91**	90.9	88.9	83.6
Outer eye corners	**96.3**	85.9	79.8	89.4
Inner eye corners	**96.3**	87.7	89.2	91.2
Nose tip	79	81.7	**82.3**	76.9
Mouth corners	**85.8**	79	81.1	78.5
Chin tip	**51.4**	51.33	37.6	50.2

The four feature types exhibit almost equivalent performance; DCT features provide more accurate results on average (3-6 % higher performance, see Figure 5.6). The inner eyebrow corners, the inner and outer eye corners can be successfully localized in

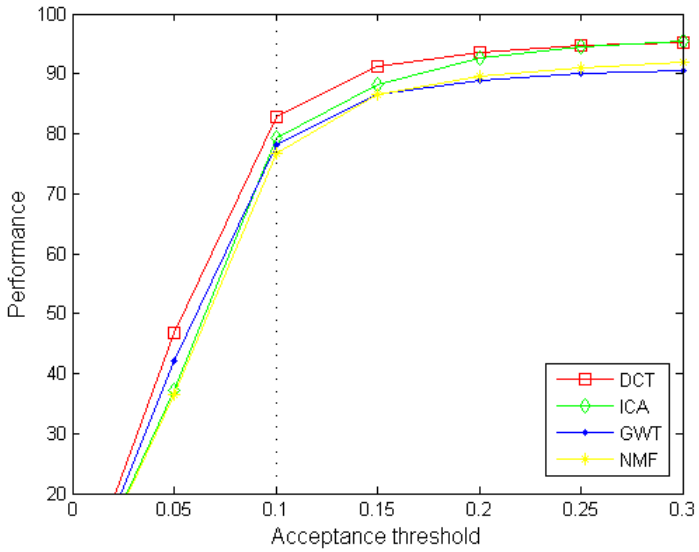


Figure 5.6. Comparison of average performance of individual feature extraction methods.

all cases. However, all feature channels drastically fail in the localization of the chin tip. This problem can be handled by using larger search windows, i.e., 16×16; but the face images in the Bosphorus database are cropped just below the chin which does not allow block-based methods to do a proper search. Inversely, Figure 5.7 represents the normalized error (m_{error}/IOD) histograms of DCT features corresponding to different landmark points. While the error values of eye corners and mouth corners peak at approximately 0.05, the nose and chin error have a much broader distribution. We performed the tests with different number of coefficients (i.e., 35, 44, 52); as the number coefficient increases we obtain gradually diminishing improvements in accuracy. For example, the average performance of DCT increases 4.8% with 52 coefficients as compared to 44 coefficients.

5.2.2. Performance of Fusion Schemes

As mentioned before, subspace methods are unable to handle extreme facial expressions and occlusions individually. Some landmarking examples are shown in Figure 5.8-(a). To overcome these difficulties, we proposed to eliminate irrelevant estimates via fusion schemes and then, to complete the missing ones via graph aided structural completion method. The results are as follows.

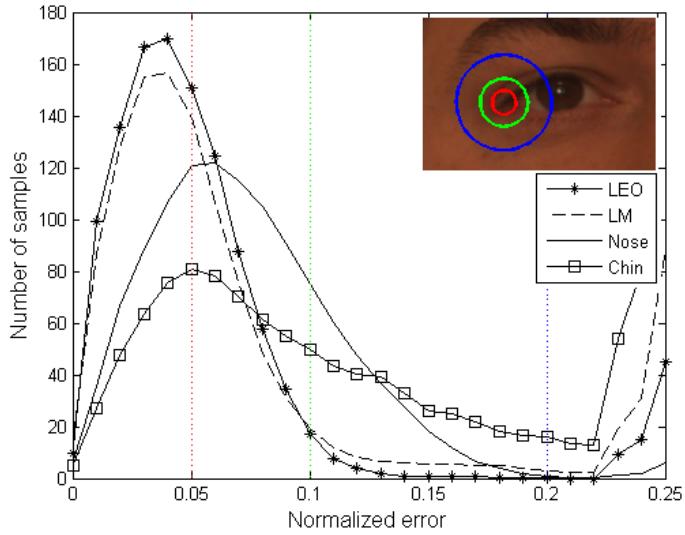
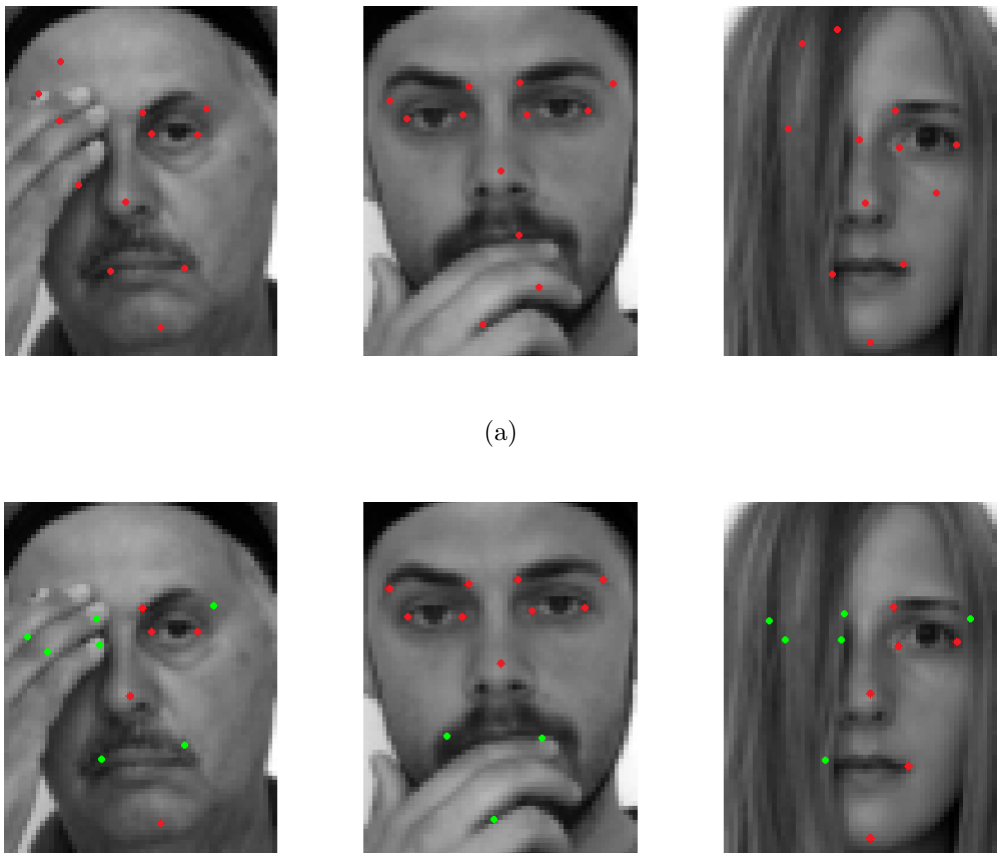


Figure 5.7. Histogram of normalized error. Red circle includes the points having error value smaller than 0.05; similarly, green and blue circles correspond to error values of 0.1 and 0.2, respectively. (LEO = left eye corner, LM = left mouth corner)

5.2.2.1. Feature Fusion:. In training stage, we concatenate the features obtained from each channel. We keep 52 coefficients for each method and the resulting feature vector is $52 \times 4 = 208$ dimensional. Then, SVM classifiers are trained by using these joined feature vectors. In the same way, given a test image, the individual feature vectors are fused for each search point and then used to test SVM classifiers. The average performance of feature fusion technique is given in Figure 5.9 vis-á-vis the individual performances. Despite all those efforts, we could not observe a significant improvement in performance; the average performance increases 0.4% as compared with DCT features (acceptance threshold = 0.1). For dimensionality reduction, PCA or LDA can be helpful to obtain more sparse representations.

5.2.2.2. Fusion by weighted median:. In this scheme, the feature vectors of different types are processed individually to generate landmark estimates. The candidate points having SVM scores above a threshold are collected from each channel and sorted relative to their spatial position. Then, we simply assign the median of the list as the reliable point for a specific landmark by weighting the positions with SVM scores. For each sample, the number of reliably detected landmark points depends on the difficulty of the pose, i.e., while this number can be up to 10-12 in neutral poses, in occlusions it



(a)

(b)

Figure 5.8. Landmarking examples of the proposed methods under occlusions: (a) Detected points by using only DCT features, (b) Red points represent the reliably detected points (outputted by score fusion); green points are replaced by structural completion method.

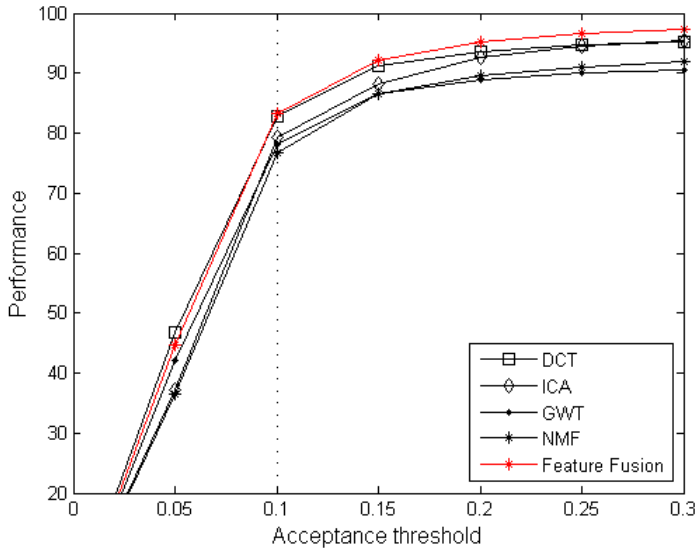


Figure 5.9. Average landmarking performance after feature fusion scheme (Train set = Bosphorus, test set = Bosphorus).

decreases down to 3-5. As explained in Section 4.2, the eliminated points, landmarks for which no candidates survive, can be recovered by structural correction method. Finally, we achieve the results as shown in Figure 5.8-(a). We also compare the performance of individual feature localizers and the proposed method in Figure 5.10; this time, an increase of 1.7% is observed in the average performance with acceptance threshold = 0.1).

The Bosphorus database is composed of high quality images and homogeneous illumination conditions. Consequently, the individual feature channels exhibit all similar performance and work very well if the SVM classifiers are trained properly against facial expressions. However, when the texture information has been lost (i.e., in hand or hair occlusion, in the presence of facial hair, eye glasses), the score fusion method followed by structural completion performs better. The detailed experimental results are given in Section 5.2.3.1. Similarly, JAFFE database is quite similar to Bosphorous database in content (uniform illumination and extreme facial expressions); the gain is 1.8% by the proposed method as shown in Figure 5.11-(a). On the other hand, in BioID database, the low quality images and adverse illumination conditions result in significantly lower performance. In this case, it does pay to fuse the scores of individual feature channels followed by graph-completion resulting in a net improvement 5.8%,

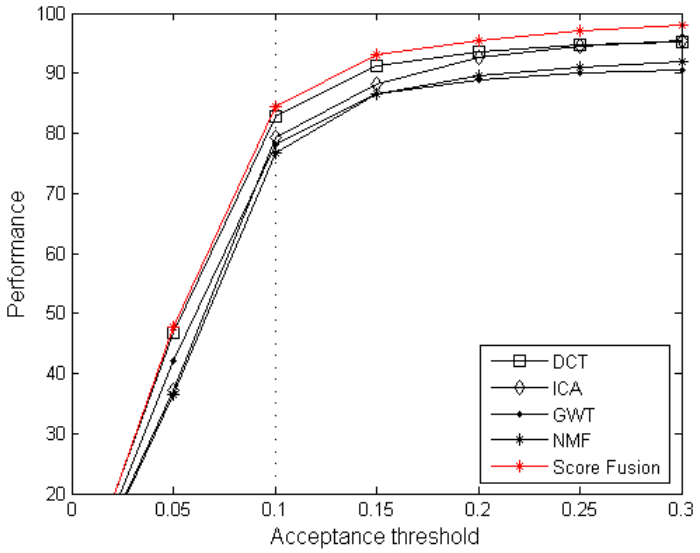
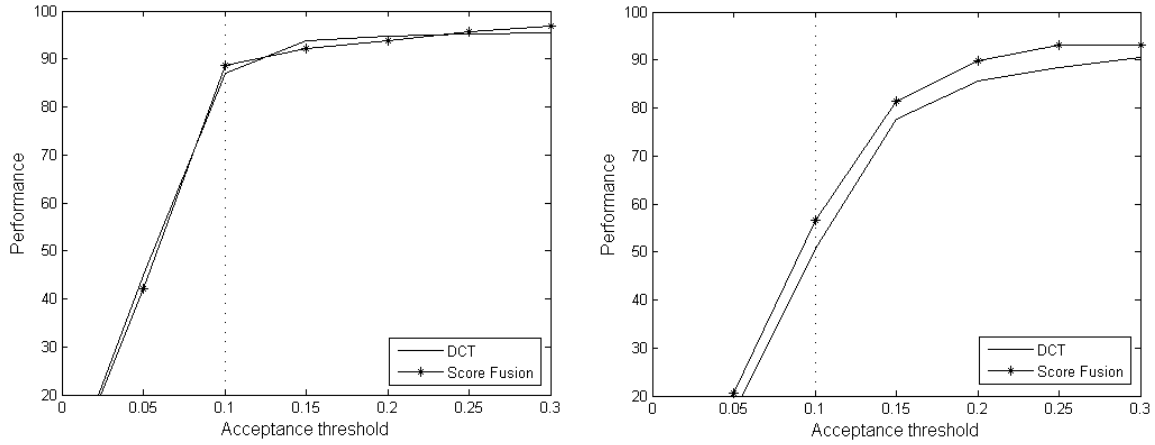


Figure 5.10. Average landmarking performance after score fusion scheme (Train set = Bosphorus, test set = Bosphorus).

see Figure 5.11-(b). The missing landmarks are estimated via graph-completion algorithm, which is itself based on the more accurately detected landmarks, hence yielding overall more reliable results.

Furthermore, we separately analyze the score fusion and structural completion methods for their individual contribution to the performance. Table 5.4 presents the relationship between landmarking errors or deviations for each feature channel. We compute covariance of error values relative to different landmark points where large values indicate that there is a considerable amount of overlapping between errors of two different feature channels; thus nothing is practically gained by any fusion. In Bosphorus database, we invoke the structural completion method for 82% of the test samples, for which the average localization performance is 82.7% while 81.5% with only DCT features. For the remaining ones, namely without graph-aided completion, the average performance of DCT features and fused features are 88.4% and 88.8%, respectively. To measure the contribution of the score fusion scheme, we also repeat the same experiment by using only DCT features plus structural completion; the average performance fall down to 79.3% while the average performance is 84.5% with score fusion followed by graph-aided completion.



(a) (b)

Figure 5.11. Average landmarking performance after score fusion scheme: (a) Train set = JAFFE, test set = JAFFE, (b) Train set = BioID, test set = BioID.

Table 5.4. Error covariance matrix relative to different landmark points.

	DCT	ICA	NMF
ICA	0.83		
NMF	0.85	0.86	
GWT	0.86	0.82	0.85

(a) Left Eyebrow Outer Corner

	DCT	ICA	NMF
ICA	0.90		
NMF	0.85	0.82	
GWT	0.91	0.88	0.81

(b) Left Eye Outer Corner

	DCT	ICA	NMF
ICA	0.92		
NMF	0.90	0.91	
GWT	0.92	0.89	0.90

(c) Left Mouth Corner

	DCT	ICA	NMF
ICA	0.33		
NMF	0.40	0.26	
GWT	0.31	0.22	0.36

(d) Nose Tip

5.2.3. Challenging experiments

In this section, we compare the localization performance of the proposed algorithm with respect to different facial expressions and occlusions common to the Bosphorus database. In a second experiment, we analyze the cross-database effects, that is, training the landmarker in one database and testing on a diverse database. The results are as follows:

5.2.3.1. Performance variation under facial expressions. In Figure 5.12, we separately investigate the localization performance for each type of landmark. One can observe that the most successfully detected landmarks are inner and outer eye corners which have an accuracy of 97.9% and 97.5%, respectively. The inner eyebrow corners (92.7%) can also be localized with high accuracy. The mouth corners and outer eyebrow corners are the most affected landmark points by facial expressions. All but one of the landmark points can be detected fairly successfully (80-96% accuracy), while chin tip fails most of the time (53.1%). Moreover, we present the performance fluctuations with respect to different facial expressions in Figure 5.13. As expected, the highest performance is obtained in neutral poses. The inner eye corners are more robust than mouth corners as we observe more variation in mouth localization performance. For example, in images including extreme facial action units, such as lip suck (71.9%), cheek puff (79.2%), nose wrinkler (78.2%) etc., the performance decreases rapidly while it is 92.6% in neutral poses. Finally, we examine the contribution of the proposed method for each face scenario, see Figure 5.14. Especially, the most amount of contribution is provided in occluded poses. However, one disadvantage of structural completion is that this method sometimes fails to replace the missing fiducial points in extreme facial action units even if the action unit does not include those points. For example, in facial action units including eyes or eyebrows such as brow lowerer, eyes closed, if the fusion scheme can not detect any reliable points on mouth corners, the structural-completion method cannot estimate precisely the location of mouth corners based on the detected ones.

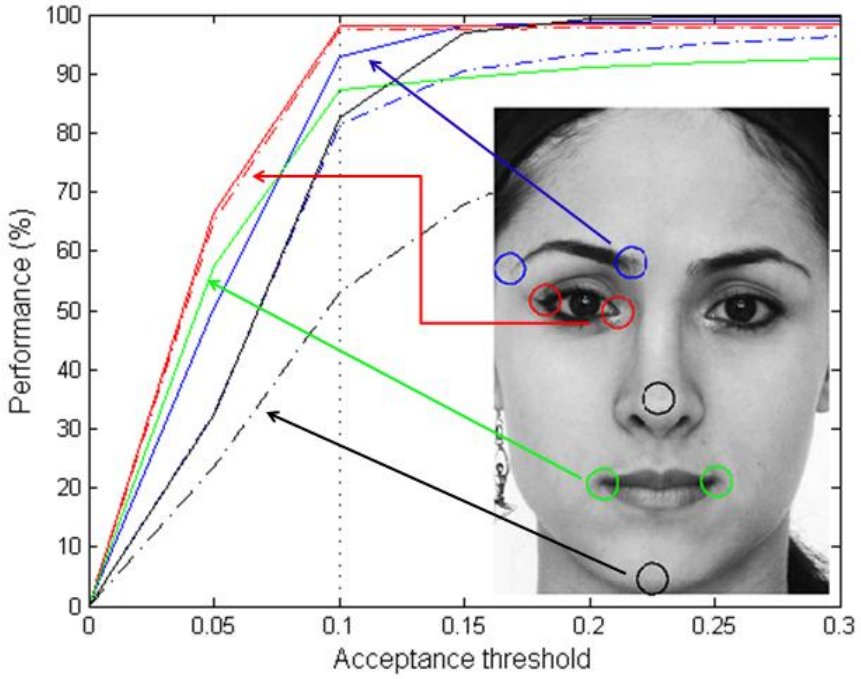


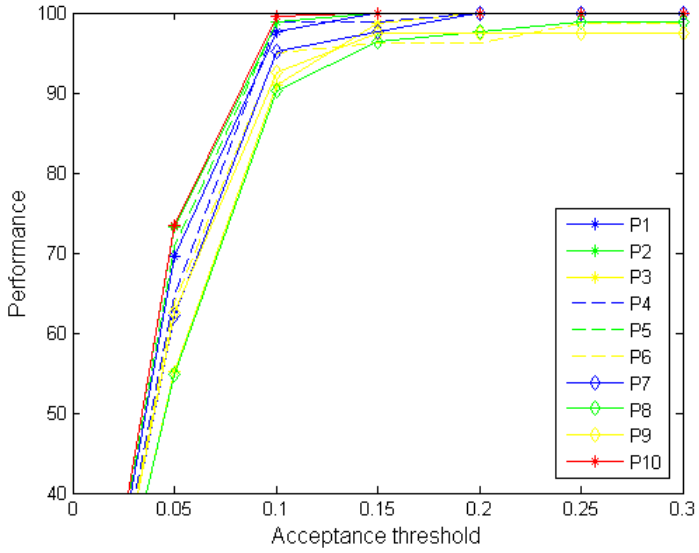
Figure 5.12. Comparison of localization performance corresponding different landmark types. Circles over the landmarks have a radius equivalent to the $0.1 \times IOD$.

5.2.3.2. Testing on different databases. In this part, three different experimental setups are conducted. First, the feature detectors trained on one database (say Bosphorus) are tested on a different database (BioID or JAFFE). Second, a data set including samples from two different databases, e.g. BioID and Bosphorus, is used for training and testing. Finally, the detectors are trained using all three databases.

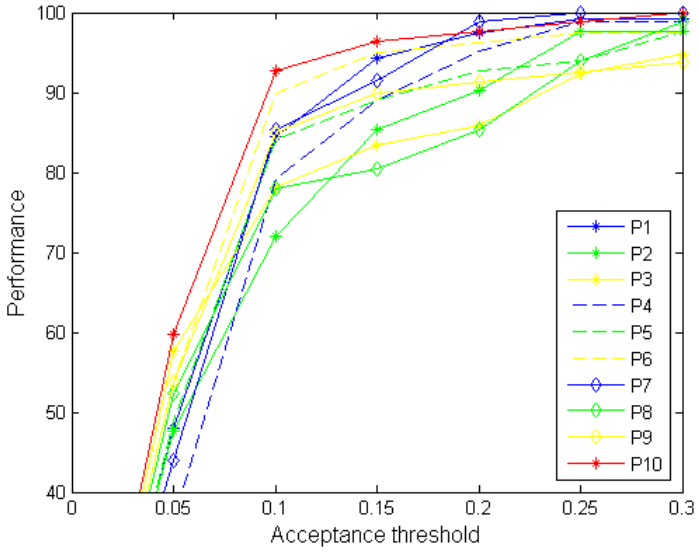
The highest performance is obtained when we perform both training and test on the same database, as evident in Figure 5.15. On the other hand, if we train over BioID (JAFFE) and test on the other one, that is Bosphorus, the landmarking algorithm deteriorates rapidly. The cross-database experiments are summarized in Table 5.5.

5.2.4. Comparison of different approaches

In this section, we compare the performance of our proposed method with two different approaches [7], [28]:

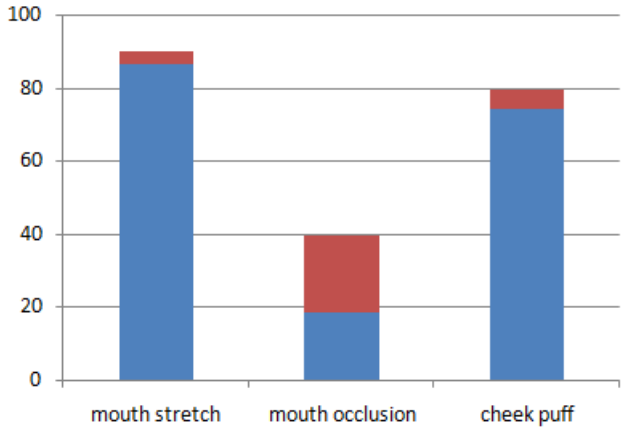


(a) Inner eye corners

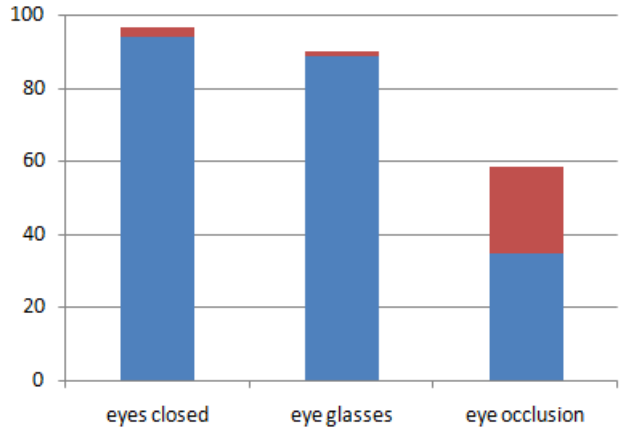


(b) Mouth corners

Figure 5.13. Variation in the localization performance with respect to different facial expressions (P1 = mouth stretch, P2 = lip suck, P3 = nose wrinkler, P4 = cheek puff, P5 = outer brow raiser, P6 = brow lowerer, P7 = eyes closed, P8 = happiness, P9 = eye glasses, P10 = neutral).



(a) Mouth corners



(b) Outer eye corners

Figure 5.14. Contribution of the proposed method. Blue bars represent the performance of DCT features; the increase with score fusion plus structural completion method is in red.

Table 5.5. Average performances over varying training datasets (%). Acceptance threshold = 0.1

	Training set			
Test set	Bosphorus	BioID	JAFFE	All
Bosphorus	84.5	64	51.8	59.2
BioID	52.6	54.5	36.2	51.7
JAFFE	68.4	49.1	88.7	66.7

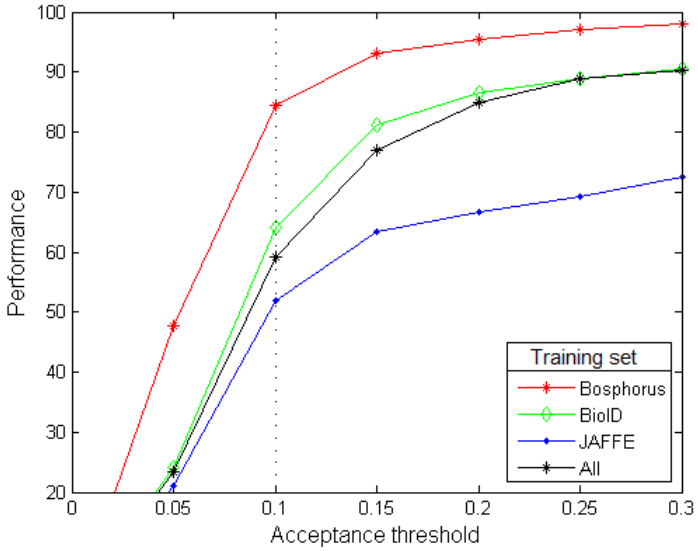


Figure 5.15. Cross-database effects on Bosphorus database.

5.2.4.1. PGM vs. Proposed method. There are two different approaches in [28]: PGM-I and PGM-II. In our experiments, we adopted Probabilistic Graph Model (PGM-I) which is used for eliminating the false alarms and replacing these points with their estimate locations. The main difference of PGM is that the reliable points are determined by anthropometrical information (angles and distances between landmark points). Once the support set is defined, the missing landmark points are replaced by structural completion method. Both of the methods exhibit equivalent performance as presented in Table 5.6. However, as the number of reliably detected points are fixed, PGM cannot handle extreme facial expressions properly. Especially, the most problematic landmark points are mouth corners and outer eyebrow corners; the structural completion is generally insufficient to cover their variations. On the other hand, in occlusions, PGM is a better method to overcome false positive responses of the SVM classifier.

5.2.4.2. EBGM vs. Proposed method. The results of Elastic Bunch Graph algorithm are given in Table 5.7. However, we have some problems in the implementation of this algorithm; it works very well in the first step (face localization) where we only use the magnitude similarity. In the second and third steps, the estimation of the displacement sometimes results in irrelevant points; thus, we cannot obtain reliable points with phase similarity. To overcome this problem, we used a grid search method

Table 5.6. Performance of PGM algorithm (%). Acceptance threshold = 0.1.

	Score fusion	PGM
Outer eyebrow corners	81.4	76.7
Inner eyebrow corners	92.7	87.7
Outer eye corners	97.9	95.2
Inner eye corners	97.5	95.7
Nose tip	82.5	80.6
Mouth corners	87.1	79.5
Chin tip	53.1	61.8

where we search a $n \times n$ pixel grid around the estimated location at the previous step instead of estimating the displacement by Equation 4.9. The displacement measure that produces the best similarity can be considered as our estimation for the novel jet. The main disadvantage of the grid search method is its computational complexity. We tried the same idea by using only magnitude similarity; the localization performance is better than using phase similarity. To handle the difficulties with EBGM algorithm, different methods can be used for displacement estimation. For example, Solari et. al [67] introduced a new disparity measurement technique which compute the phase difference directly in the complex plane.

Table 5.7. Performance of EBGM algorithm (%). Acceptance threshold = 0.1.

	Using only magnitude similarity	Using magnitude similarity and phase similarity
Outer eyebrow corners	72.3	63.1
Inner eyebrow corners	75.8	68.2
Outer eye corners	77.6	69.4
Inner eye corners	84.2	67.3
Nose tip	70.4	59.1
Mouth corners	69.6	54.9
Chin tip	58.7	41.5

6. CONCLUSIONS

6.1. Summary

This thesis provides a comprehensive review and a reasonable comparison of existing methods for facial feature extraction. The novelty consists in the multi-feature approach for the determination of fiducial points on faces and in a fusion scheme based on weighted median filter. The following conclusion can be drawn:

- Feature detectors being robust against facial hair, eye glasses, facial expressions, can result from subspace methods when the SVM classifiers are trained properly.
- Score fusion seems to contribute somewhat especially when faces are captured under uncontrolled conditions (BioID database). Otherwise, it only help to combat against the effects of extreme facial expressions($\sim 1-5\%$) and occlusions ($\sim 20\%$) with the help of structural completion method.
- If we train over a database (BioID) and test on the other one, i.e. Bosphorus, there is a considerable amount of decrease in the performance of the landmarking algorithm. However, some of the lost performance can be recuperated via fusion scheme ($\sim 5\%$).
- Whether one uses model-driven features such as DCT or NMF coefficients or one uses data-driven features such as Adaboost features as in the Viola-Jones algorithm, we have come to the determination that the face landmarking problem must be attacked by a face pose, followed by pose-specialized landmarker.

6.2. Future Work

This study can be advanced along several avenues: 1) The feature search area can be reduced by defining an ellipsoid around the facial features. For a specific landmark point, i.e. eye centers, this ellipsoid can be defined as enclosing 90% of the eye centers within the detected face region of our training images. 2) All tested subspace methods exhibit almost equivalent performance; it will be useful determine

the most discriminative feature set. This can be achieved by using a feature selection algorithm such as Sequential Floating Forward Search (SFFS). 3) We plan to extend this work to detect the orientation and pose ($\pm 30^o$, $\pm 60^o$, $\pm 90^o$) of the face by using multiple graphs corresponding to different poses. 4) Another goal is to investigate alternate features to fuse with DCT features. Matched spatial filter will be a good candidate for this purpose [30]. We can utilize a two-step search algorithm. In the first step, the facial components such as eyes, nose and mouths can be detected by template matching method and then, the DCT features can be used to determine their fiducial points.

For further study, we intend to apply the multi-class object detection method introduced by Torralba et. al in [68] to facial feature detection. Their approach, called as joint boosting, is a variation on boosting that allows for sharing features across objects, automatically selecting the best sharing pattern. More explicitly, the detectors for each class trained jointly, rather than independently. The main advantage of shared features is learning from fever examples with fewer features.

REFERENCES

1. González-Jiménez, D., and J. L. Alba-Castro, "Towards Pose Invariant 2D Face Recognition Through Point Distribution Models and Facial Symmetry", *IEEE TIFS*, vol. 2, pp 413-429, Sep. 2007.

2. Turker, M. and A. Petland, "Face Recognition using Eigenfaces", *Journal of cognitive neuroscience*, 3(1), 1991.

3. Martinez, A. M. and A. C. Kak, "PCA versus LDA", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):228-233, 2001.

4. Brunelli, R. and T. Poggio, "Face Recognition: Features versus Templates", *IEEE Transactions, PAMI*, pp. 1042-1052, 15(10), 1993.

5. Heisele, B., P. Ho, and T. Poggio, "Face Recognition with Support Vector Machines: Global versus Component-based Approaches", *Proceedings IEEE International Conference on Computer Vision (ICCV2001)*, pp. 688-694, 2001.

6. Pentland, A., B. Moghaddam, and T. Starner, "View-based and Modular Eigenspaces for Face Recognition", *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 84-91, 1994.

7. Wiskott, L., J. M. Fellous, N. Kruger, and C. von der Malsburg, "Face Recognition by Elastic Bunch Graph Matching", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, no:7, pp. 775-779, 1997.

8. Dornaika, F. and F. Davoine, "Online Appearance-based Face and Facial Feature Tracking", *In Proceedings of the 17th International Conference on Pattern Recognition*, pp. 814-817, 2004.

9. Tong, Y. and Qiang Ji, "Multiview Facial Feature Tracking with a Multi-modal

Probabilistic Model", *Proceedings of the 18th International Conference on Pattern Recognition (ICPR)*, pp. 307-310, Washington, DC, USA, 2006.

10. Cohn, J., A. Zlochower, J.-J. James Lien, and T. Kanade, "Feature-point Tracking by Optical Flow Discriminates Subtle Differences in Facial Expression", *In Proceedings of the 3rd IEEE International Conference on Automatic Face and Gesture Recognition (FGR)*, pp. 396-401, April 1998.

11. Chen, J. and B. Tiddeman, "Robust Facial Feature Tracking under Various Illuminations", *In IEEE International Conference on Image Processing*, pp. 2829-2832, 2006.

12. Wieghardt, J., R. P. Würtz, and C. von der Malsburg, "Gabor-based Feature Point Tracking with Automatically Learned Constraints", *In Proceedings ECCV*, Copenhagen, 2002.

13. Ekman, P. and W. V. Friesen, "Facial Action Coding System (FACS)", *Consulting Psychologists Press*, 1978.

14. Cohn Y., J. Tian, and T. Kanade, "Recognizing Action Units for Facial Expression Analysis", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):97114, 2001.

15. Pantic, M. and L.J.M. Rothkrantz, "Automatic Analysis of Facial Expressions: The State of the Art", *Pattern Analysis and Machine Intelligence*, 22(12):14241445, 2000.

16. Braathen, B., M. S. Bartlett, G. Littlewort, E. Smith, and J. Movellan, "An Approach to Automatic Recognition of Spontaneous Facial Actions", *International Conference on Automatic Face and Gesture Recognition (FGR)*, 2002.

17. Park I. K., H. Zhang, V. Vezhnevets, and H. K. Choh, "Image-based Photorealistic 3-D Face Modeling", *IEEE International Conference on Automatic Face and*

Gesture Recognition (FGR), 2004.

18. Xin, L., Q. Wang, J. Tao, X. Tang, T. Tan, and H. Shum, "Automatic 3D Face Modeling from Video", *IEEE International Conference on Computer Vision (ICCV)*, 2005.

19. Blanz, V. and T. Vetter, "A Morphable Model for the Synthesis of 3D Faces", *SIGGRAPH Conference Proceedings*, pp. 187-194, 1999.

20. Ryu, Y. S. and S. Y. Oh, "Automatic Extraction of Eye and Mouth Fields from a Face Image Using Eigenfeatures and Ensemble Networks", *Applied Intelligence*, vol. 17, pp. 171-185, 2002.

21. Smeraldi, F. and J. Bigun, "Retinal Vision Applied to Facial Features Detection and Face Authentication", *Pattern recognition letters*, 23:463475, 2002.

22. Feris, R. S., J. Gemmell, K. Toyama, and V. Krüger, "Hierarchical Wavelet Networks for Facial Feature Localization", *International Conference on Automatic Face and Gesture Recognition*, Washington D.C., USA, 2002.

23. Akakın, H. Ç., A. A. Salah, L. Akarun, and B. Sankur, "2D/3D Facial Feature Extraction", *SPIE Conf. on Electronic Imaging*, 2006.

24. Salah, A. A., H. Çınar, L. Akarun, and B. Sankur, "Robust Facial Landmarking for Registration", *Annals of Telecommunications*, December, 2006.

25. Ersi, E. F. and J. S. Zelek, "Rotation-Invariant Facial Feature Detection Using Gabor Wavelet and Entropy", *International Conference on Image Analysis and Recognition (ICIAR)*, Toronto, Canada, 2005.

26. Antonini, G., V. Popovici and J. P. Thiran, "Independent Component Analysis and Support Vector Machine for Face Feature Extraction", *Int. Conf. on Audio and Vide-based Biometric Person Authentication*, pp. 111-118, Guildford, UK, 2003.

27. Zobel M., A. Gebhard, D. Paulus, J. Denzler, and H. Niemann, "Robust Facial Feature Localization by Coupled Features", *IEEE International Conference on Automatic Face and Gesture Recognition (FGR)*, Grenoble, France, 2000.

28. Akakın, H. Ç. and B. Sankur, "Robust 2D/3D Face Landmarking", *3DTV Con.*, 2007.

29. Lin, C. H. and J.L. Wu, "Automatic Facial Feature Extraction by Genetic Algorithms", *IEEE Transactions on Image processing*, 8(6):-, 1999.

30. Brunelli, R. and T. Poggio, "Template Matching: Matched Spatial Filters and Beyond", *MIT Technical Report*, AIM-1549, 1995.

31. Zhang, L., "Estimation of the Eye and Mouth Corner Point Positions in a Knowledge-based Coding System",*Digital Compression Technologies and Systems for Video Communications*, vol. 2952, pp. 21-28, 1996.

32. Yuille, A. L., D.S. Cohen, and P.W. Hallinan, "Feature Extraction from Faces using Deformable Templates", *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pp. 104-109, San Diego, CA, USA, 1989.

33. Herpers, R. and G. Sommer, "An Attentive Processing Strategy for the Analysis of Facial Features", *Face recognition*, pp. 457468, Springer, London, 1998.

34. Zhang, B. and Q. Ruan, "Facial Feature Extraction using Improved Deformable Templates", *IEEE Int. Conf. on Signal Processing*, 2006.

35. Chan, M. T., Y. Zhang, and T. S. Huanz, "Real-time Lip Tracking and Biomodal Continuous Speech Recognition", *Electronic Proceedings of Workshop on Multimedia Signal Processing*, LA, California, USA, IEEE Signal Processing Society, 1998.

36. Cristinacce, D., T. Cootes, and I. Scott, "A Multi-Stage Approach to Facial Feature Detection", *Proc.* 15^{th} *British Machine Vision Conference*, pp. 277-286, 2004.

37. Cootes, T., C. J. Taylor, D. H. Cooper, and J. Graham "Active Shape Models Their Training and Application", *Computer Vision and Image Understanding*, vol. 61, pp. 38-59, 1995.

38. Cootes, T., G. Edwards, and C. Taylor, "Active Appearance Models", *Proc. Int. Conf. on Computer Vision*, 2:484-498, 1998.

39. Sohail, A.S.M. and P. Bhattacharya, "Detection of Facial Feature Points Using Anthropometric Face Model", *IEEE International Conference on Signal-Image Technology and Internet-Based Systems*, pp. 656-665, Tunisia, December, 2006.

40. Shih, F. Y. and C. F. Chuang, "Automatic Extraction of Head and Face Boundaries and Facial Features", *Special issue on Informatics and computer science intelligent systems applications*, pp. 117-130, 2004.

41. Tsekeridou, S. and I. Pitas, "Facial Feature Extraction in Frontal Views using Biometric Analogies", *Proc. of the IX European Signal Processing Conference*, I:315318, 1998.

42. Başkan, S., M. M. Bulut, and V. Atalay, "Projection based Method for Segmentation of Human Faces and its Evaluation", *Pattern Recognition Letters*, vol. 23(14), pp. 1623-1629, 2002.

43. Vukadinovic, D. and M. Pantic, "Fully Automatic Facial Feature Point Detection using Gabor Feature Based Boosted Classifiers", *IEEE Int. Conf. on Systems, Man and Cybernetics*, Hawaii, October 2005.

44. Arca, S., P. Campadelli and R. Lanzarotti, "An Efficient Method to Detect Facial Fiducial Points for Face Recognition", *IEEE Conf. on Pattern Recognition (ICPR)*, 2004.

45. Zhu, Z., and Q. Ji, "Robust Pose Invariant Facial Feature Detection and Tracking in Real-Time", *Int. Conf. on Pattern Recognition*, vol. 1, pp. 1092-1095, 2006.

46. Cesar, R., E. Bengoetxea and I. Bloch, "Inexact Graph Matching using Stochastic Optimization Techniques for Facial Feature Recognition", *Int. Conf. on Pattern Recognition (ICPR)*, vol. 2, p. 20465, 2002.

47. Cristinacce, D. and T.F. Cootes, "Facial Feature Detection using Adaboost with Shape Constraints", *Proc. BMVC2003*, vol.1, pp.231-240, 2003.

48. Viola, P. and M. Jones, "Robust real-time object detection", *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137154, 2001.

49. Fang, L. Z., Y. Z. Sheng, A. K. Jain and W. Y. Qiong, "Face Detection and Facial Feature Extraction in Color Image", *International Conference on Computational Intelligence and Multimedia Applications (ICCIMA)*, Xi'an, China, 2003.

50. Phimoltares, S., C. Lursinsap, and K. Chamnongthai, "Face Detection and Facial Feauture Localization without Considering the Appearance of Image Context", *Image and Vision Computing*, vol. 25, pp. 741-753, 2007.

51. Ahlberg, J., "A System for Face Localization and Facial Feature Extraction", *Technical Report LiTH-ISY-R-2172*, 1999.

52. Colbry, D., G. Stockman and A. Jain, "Detection of Anchor Points for 3D Face Verification", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005.

53. Conde, C., L. J. Rodríguez-Aragón, and Enrique Cabello, "Automatic 3D Face Feature Points Extraction with Spin Images", *Image Analysis and Recognition*, LNCS, vol. 4142, pp. 317-328, Springer-Verlag, 2006.

54. Gökberk, B., M. O. İrfanoğlu, and L. Akarun, "3D Shape-based Face Representation and Feature Extraction for Face Recognition", *Image and Vision Computing*, vol. 24(8), pp. 857-869, Aug. 2006.

55. Akagündüz, E. and İ. Ulusoy, "Extraction of 3D Transform and Scale Invariant

Patches from Range Scans", *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2007.

56. Boehnen, C. and T. Russ, "A Fast Multi-Modal Approach to Facial Feature Detection", *IEEE Workshop on Applications of Computer Vision*, pp. 135-142, Breckenridge, USA 2005.

57. Hyvärinen, A. and E. Oja, "A Fast Fixed-Point Algorithm for Independent Component Analysis", *Neural Computation*, vol. 9(7), pp. 1483-1492, 1997.

58. Bartlett, M.S., J. R. Movellan, and T. J. Sejonowski, "Face Recognition by Independent Component Analysis", *IEEE Trans. on Neural Networks*, vol. 13, no. 6, 2002.

59. Lee, D. D. and H. S. Seung, "Algorithms for Nonnegative Matrix Factorization", *Advances in Neural and Information Processing Systems 13*, pp. 556-562, 2001.

60. Lin, C. J. , "Projected Gradient Methods for Non-negative Matrix Factorization", *Neural Computation*, 2007.

61. Savran, A., N. Alyüz, H. Dibeklioğlu, O. Çeliktutan, B. Gökberk, B. Sankur, and L. Akarun, "Bosphorus Database for 3D Face Analysis", *The First COST 2101 Workshop on Biometrics and Identity Management (BIOID)*, Roskilde University, Denmark, 7-9 May 2008.

62. A reference for The BioID Face Database (http://www.bioid.com/downloads/facedb/index.php).

63. Lyons, M. J., J. Budynek, and S. Akamatsu, "Automatic Classification of Single Facial Images", *IEEE Transactions on Pattern analysis and Machine Intelligence*, vol. 21, no. 12, pp. 1357-1362, 1999. As a reference for the JAFFE database (http://www.kasrl.org/jaffe.html).

64. A reference for MATLAB Face Detector

(http://staff.science.uva.nl/ anoulas/MatlabFaceDetector.htm).

65. Heisele, B., T. Poggio, and M. Pontil, "Face Detection in Still Gray Images", *AI Memo 1687*, Center for Biological and Computational Learning, MIT, Cambridge, MA, 2000.

66. Chang, C.C. and C.J. Lin, "LIBSVM: A Library for Support Vector Machines", 2001. Software available at http://www.csie.ntu.edu.tw/cjlin/libsvm.

67. Solari, F., S.P. Sabatini, and G.M. Bisio, "Fast Technique for Phase-based Disparity Estimation with no Explicit Calculation of Phase", *Electron. Lett.*, vol. 73, pp. 1382-1383, 2001.

68. Torralba, A., K. P. Murphy and W. T. Freeman, "Sharing Visual Features for Multiclass and Multiview Object Detection", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, No. 5, pp. 854-869, 2007.