HUMAN FACE DETECTION AND EYE LOCALIZATION IN VIDEO USING WAVELETS

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND

ELECTRONICS ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Mehmet Türkan

December, 2006

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

Prof. Dr. Ahmet Enis Çetin(Supervisor)

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

Prof. Dr. Özgür Ulusoy

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

Assoc. Prof. Dr. Uğur Güdükbay

Approved for the Institute of Engineering and Science:

Prof. Dr. Mehmet B. Baray
Director of the Institute Engineering and Science

ABSTRACT

HUMAN FACE DETECTION AND EYE LOCALIZATION IN VIDEO USING WAVELETS

Mehmet Türkan
M.Sc. in Electrical and Electronics Engineering
Supervisor: Prof. Dr. Ahmet Enis Çetin
December, 2006

Human face detection and eye localization problems have received significant attention during the past several years because of wide range of commercial and law enforcement applications. In this thesis, wavelet domain based human face detection and eye localization algorithms are developed. After determining all possible face candidate regions using color information in a given still image or video frame, each region is filtered by a high-pass filter of a wavelet transform. In this way, edge-highlighted caricature-like representations of candidate regions are obtained. Horizontal, vertical and filter-like edge projections of the candidate regions are used as feature signals for classification with dynamic programming (DP) and support vector machines (SVMs). It turns out that the proposed feature extraction method provides good detection rates with SVM based classifiers. Furthermore, the positions of eyes can be localized successfully using horizontal projections and profiles of horizontal- and vertical-crop edge image regions. After an approximate horizontal level detection, each eye is first localized horizontally using horizontal projections of associated edge regions. Horizontal edge profiles are then calculated on the estimated horizontal levels. After determining eye candidate points by pairing up the local maximum point locations in the horizontal profiles with the associated horizontal levels, the verification is also carried out by an SVM based classifier. The localization results show that the proposed algorithm is not affected by both illumination and scale changes.

Keywords: Human face detection, eye detection, eye localization, wavelet transform, edge projections, classification, support vector machines, dynamic programming.

ÖZET

DALGACIK DÖNÜŞÜMÜ KULLANARAK VİDEODA İNSAN YÜZÜ VE GÖZLERİN YERLERİNİN TESPİTİ

Mehmet Türkan
Elektrik ve Elektronik Mühendisliği, Yüksek Lisans
Tez Yöneticisi: Prof. Dr. Ahmet Enis Çetin
Aralık, 2006

Video ve/veya imgelerde insan yüzü bulma ve gözlerin yerlerinin tespiti geçtiğimiz yıllarda üzerlerinde önemle durulan konular olmuşlardır. Bu ilginin nedenleri arasında, geliştirilen sistemlerin ve algoritmaların ticari ve kanun uygulamalarında çok geniş alanlarda kullanılmaları gösterilebilir. Bu tez çalışmasında, dalgacık dönüşümü kullanılarak insan yüzü bulma ve gözlerin yerlerinin tespiti algoritmaları geliştirilmiştir. Verilen bir imge veya video çerçevesinde, renk bilgisi kullanılarak, insan yüzü olabilecek tüm alanlar tespit edildikten sonra bu alanlar dalgacık dönüşümünde kullanılan yüksek-geçiren süzgeçten geçirilir. Böylece bu alanların kenarları vurgulanarak karikatür benzeri gösterimleri elde edilir. Yatay, dikey ve filtre-benzeri kenar izdüşümleri öznitelik sinyali olarak dinamik programlama ve destekçi vektör makinaları kullanılarak sınıflandırılır. Deneysel sonuçlar, kullanılan öznitelik sinyallerinin destekçi vektör makinaları ile daha iyi sınıflandırıldığını göstermektedir. Benzer bir yöntemle, insan yüzü olarak sınıflandırılmış bir alanın yatay- ve dikey-kesim kenar bölgelerinin yatay izdüşümleri ve profilleri kullanılarak gözlerin yerleri tespit edilebilir. İlgili kenar bölgelerinin yatay izdüşümleri kullanılarak gözlerin yatay seviyeleri tespit edildikten sonra bu yatay seviyeler üzerinde her göz için ayrı ayrı yatay kenar profilleri hesaplanır. Göz olabilecek tüm noktalar yatay kenar profilleri ve yatay seviyelerin yardımı ile tespit edilir, ve destekçi vektör makinaları ile sınıflandırılır. Deneysel sonuçlar, geliştirilen algoritmanın aydınlanma ve ölçek değişimlerinden etkilenmediğini göstermektedir.

Anahtar sözcükler: İnsan yüzü tespiti, gözlerin tespiti, gözlerin yerlerinin tespiti, dalgacık dönüşümü, kenar izdüşümleri, sınıflandırma, destekçi vektör makinaları, dinamik programlama.

Acknowledgment

I would like to express my deep gratitude to my supervisor Prof. Dr. Ahmet Enis Çetin for his instructive comments and constant support throughout this study.

I would like to express my special thanks to Prof. Dr. Özgür Ulusoy and Assoc. Prof. Dr. Uğur Güdükbay for showing keen interest to the subject matter and accepting to read and review the thesis.

I would like to thank to Assoc. Prof. Dr. Montse Pardàs for her help and instructive comments in some part of this thesis.

I would also like to thank to my family for their support throughout my life.

This study is fully supported by The Scientific and Technical Research Council of Turkey (TÜBİTAK [http://www.tubitak.gov.tr/]) and, in part, by European Commission Multimedia Understanding through Semantics, Computation and Learning Network of Excellence (MUSCLE-NoE [http://www.muscle-noe.org/] with grant no. FP6-507752), and European Commission Integrated Three-Dimensional Television - Capture, Transmission, and Display Network of Excellence (3DTV-NoE [https://www.3dtv-research.org/] with grant no. FP6-511568) projects.

Dedicated to my family...

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Digital image and video processing algorithms [1] have found wide range of application areas with the recent development of fast and reliable computer technologies. Real-time video analysis is extensively used in video surveillance, quality control in industry and interactive multimedia communication systems. In this thesis, we focused on a surveillance and/or human-computer interaction application and propose algorithms, i.e., human face detection and eye localization, based on image and video processing techniques using wavelet analysis.

Wavelet transformation methods for signal analysis and representation have become very popular on the area of image and video processing. Wavelet theory was developed by independent researchers working in distinct communities such as engineering, physics and mathematics [2]. Advances in wavelet theory let researchers use this theory in wide range of applications because wavelet domain methods lead to computationally efficient solutions in signal analysis and synthesis problems in both time and frequency domains.

The specific application that we considered is the detection of human faces and location of eyes in a given still image or video. Following sections explain the motivations behind this research, summarize the previous work and conclude with an outline of the organization of this thesis.

1.1 Related Work in Literature

1.1.1 Face Detection

Human face detection problem has received significant attention during the past several years because of wide range of commercial and law enforcement applications. In recent years, many heuristic and pattern recognition based methods have been proposed to detect human faces in still images and video on gray-scale or color. Human face detection techniques based on neural networks (NNs) [3, 4], support vector machines (SVMs) [5, 6], hidden Markov models (HMMs) [7, 8], Fisherspace/subspace linear discriminant analysis (LDA) [9], principle component analysis (PCA) [10], and Bayesian or maximum-likelihood (ML) classification methods [4] have been described in the literature ranging from very simple algorithms to composite high-level approaches. Rowley et al. [3] presented a neural network-based frontal upright face detection system using a bootstrap algorithm in gray-scale images. Small windows (20x20 pixel region) of an image are examined by a retinally connected neural network which decides whether each window contains a face. They improved the performance over a single network by arbitrating the system among multiple networks. Sung and Poggio [4] described a distribution-based modeling of face and non-face patterns using a multilayer perceptron (MLP) classifier. They developed a successful example-based learning system for detecting vertical frontal views of human faces in complex scenes by computing a difference feature vector between the local image pattern and the distribution-based model at each image location. Osuna et al. [5] demonstrated a decomposition algorithm that guarantees global optimality to train support vector machines for frontal human face detection in images over large data sets. Their system divides the original image into overlapping subimages (19x19 pixel region) while exhaustively scanning for face or face-like patterns at many possible scales. The classification is carried out by SVMs to determine the appropriate class, i.e., face or non-face. Guo et al. [6] proposed a binary tree structure for recognizing human faces after extracting features and learning the discrimination functions via SVMs. Nefian and Hayes [7] described a hidden Markov model (HMM)-based framework using the projection coefficients of the

Karhunen-Loeve Transform (KLT) for detection and recognition of human faces. Although the proposed method results slight improvements on the recognition rate, it reduces significantly the computational complexity compared to previous HMM-based face recognition systems, e.g., Samaria [8] which uses strips of raw pixels. Belhumeur et al. [9] developed a successful face recognition system using Fisher's linear discriminant which produces well-separated classes in a low-dimensional subspace, insensitive to large variation in lighting direction and facial expressions. Turk and Pentland [10] presented an eigenface-based frontal view upright orientation human face detection and identification technique using principal component analysis. Face images are projected onto a feature space (face space which is defined by the eigenfaces) that the variations in known face images are encoded very well. They developed a system which tracks the head of a subject and then the person is recognized by comparing characteristics of the face to those of known individuals with a nearest-neighbor classifier. Their framework is also capable of learning to recognize new faces in an unsupervised manner. Conceptually detailed literature surveys on human face detection and recognition are conducted by Hjelmas and Low [11] and Zhao et al. [12], respectively.

Recently, wavelet domain [13, 14] face detection methods have been developed and become very popular. The main reason is that a complete framework has recently been built in particular for the construction of wavelet bases, and efficient algorithms for the wavelet transform computation. Wavelet packets allow more flexibility in signal decomposition and dimensionality reduction as the computational complexity is an important subject for face detection systems. Garcia and Tziritas [15] proposed a wavelet packet decomposition method on the intensity plane of the candidate face regions in color images under non-constrained scene conditions, such as complex background and uncontrolled illumination. After obtaining the skin color filtered (SCF) image using the color information of the original image, they extracted feature vectors from a set of wavelet packet coefficients in each region. The face candidate region is then classified into either face or non-face class by evaluating and thresholding the Bhattacharyya distance between the candidate region feature vector and a prototype feature vector. Zhu et al. [16] described a discriminant subspace approach to capture local discriminative features in the space-frequency domain for fast face detection based on orthonormal wavelet packet analysis. They demonstrated detail (high frequency sub-band) information within local facial areas contain information about eyes, nose and mouth, which exhibit noticeable discrimination ability for face detection problem of frontal view faces in complex backgrounds. The algorithm leads to a set of wavelet features with maximum class discrimination and dimensionality reduction. The classification is carried out by a likelihood test. Their frontal view face detection system greatly reduces the computational cost. Viola and Jones [17] presented a machine learning approach for visual object detection which has an ability of processing images extremely rapidly and achieving high detection rates. They represented images in a new structure called the *integral image* which allows the (Haar-like) features used in the detector to be computed very quickly. An adaptive boosting algorithm (AdaBoost) [18] which selects a number of critical features from a larger set and yields extremely efficient classifiers is used by combining successively more complex classifiers in a *cascade* structure. In the domain of face detection, their system yields promising detection rates. Uzunov et al. [19] described an adequate feature extraction method in a frontal upright face detection system. The optimal atomic decompositions are selected from various dictionaries of anisotropic wavelet packets using the AdaBoost algorithm with a Bayesian classifier as a weak-learner. Their method demonstrates a fast learning process with high detection accuracy.

1.1.2 Eye Localization

The problem of human eye detection, localization and tracking has also received significant attention during the past several years because of wide range of human-computer interaction (HCI) and surveillance applications. As eyes are one of the most important salient features of a human face, detecting and localizing them helps researchers working on face detection, face recognition, iris recognition, facial expression analysis [20], etc.

In recent years, many heuristic and pattern recognition based methods have been proposed to detect and localize eyes in still images and video. Most of these methods described in the literature ranging from very simple algorithms to composite high-level approaches are highly associated with face detection and face recognition. Traditional image-based eye detection methods assume that the eyes appear different from the rest of the face both in shape and intensity. Dark pupil, white sclera, circular iris, eye corners, eye shape, etc. are specific properties of an eye to distinguish it from other objects [21]. Morimoto and Mimica [22] reviewed the state of the art of eye gaze trackers by comparing the strengths and weaknesses of the alternatives available today. They presented a detailed description of the pupil-corneal reflection technique, and also improved the usability of several remote eye gaze tracking techniques. Zhou and Geng [23] defined a method for detecting eyes with projection functions. After localizing the rough eye positions using Wu and Zhou method [24], they expand a rectangular area near each rough position. Special cases of generalized projection function (GPF), i.e., integral projection function (IPF), variance projection function (VPF) and hybrid projection function (HPF), are used to localize the cental positions of eyes in eye windows. Their experimental results on face databases show that all the special cases of GPF are effective in eye detection. Huang and Mariani [25] described a model-based approach to represent eye images using wavelets. Their algorithm uses a structural model to characterize the geometric pattern of facial components, i.e., eyes, nose and mouth, using multiscale filters. The normalized horizontal and vertical projections of the image of eye areas, and the directions detected along the eyes are used as features in combination with structural model. Their system detects the center and radius of the eyeballs at subpixellic accuracy with a precise eyes location algorithm using contour and region information. Cristinacce et al. [26] developed a multi-stage approach to detect facial features on a human face, including the eyes. Their method is coarse-to-fine. After applying a face detector to find the approximate scale and location of the face in the image, they extract features with individual feature detectors and combine them using Pairwise Reinforcement of Feature Responses (PRFR) algorithm. The estimated feature points are then refined using a version of the Active Appearance Model (AMM) search which is based on edge and corner features. The output of this three-stage algorithm is shown to give much better results than any other combination of methods. Jesorsky et al. [27] proposed an edge-based, fast and accurate technique that is robust to changes in illumination and background in gray-scale still images. First, the face is initially localized using the *Hausdorff distance* [28] between a general face model and possible instances of the object within the image. A refinement phase is then performed in the estimated area of the face. Finally, exact pupil locations are determined using a multilayer perceptron classifier which is trained with pupil centered images of eyes. Asteriadis et al. [29] presented a new method for eye localization based on geometric information. They applied first a face detector in order to find the location of a face in the image. After extracting the edge map in the estimated face region, they assigned a vector to every pixel which is pointing to the closest edge pixel location. Eyes are detected and localized consequently using length and slope information of these vectors. Their experimental results are better at some cases than those of previous methods in [23, 26, 27].

1.2 Organization of the Thesis

In Chapter 2, a frontal pose and upright orientation human face detection algorithm is developed in images and video on both gray-scale and color. The proposed method is then compared to the dynamic programming with support vector machine based classifiers and state-of-the-art face detection methods.

In Chapter 3, a human eye localization system is presented in images and video with the assumption that a human face region in a given still image or video frame is already detected by means of a face detector. The experimental results of the proposed algorithm are presented and the detection performance is compared with currently available eye localization methods.

In the last chapter, conclusions and summary of the results are given.

Chapter 2

Face Detection System

In this chapter, we provide a human face detection system in images and video on both gray-scale and color. Our method is based on the idea that a typical human face can be recognized from its edges. In fact, a caricaturist draws a face image in a few strokes by drawing the major edges of the face. Most wavelet domain image classification methods are also based on this fact because significant wavelet coefficients are closely related with edges [13, 15, 30].

After determining all possible face candidate regions using color information in a given still image or video frame, a single-stage 2-D rectangular wavelet transform of each region is computed. In this way, wavelet domain edge-highlighted sub-images are obtained. The low-high and high-low sub-images contain horizontal and vertical edges of the region, respectively. The high-high sub-image may contain almost all the edges, if the face candidate region is sharp enough. It is clear that the detail (high frequency) information within local facial areas, e.g., edges due to eyes, nose, and mouth, show noticeable discrimination ability for face detection problem of frontal view faces. We take advantage of this fact by characterizing these wavelet domain sub-images using their projections and obtain 1-D projection feature vectors corresponding to edge images of face or face-like regions. The advantage of 1-D projections is that they can be easily normalized to a fixed size and this provides robustness against scale changes. Horizontal and vertical projections are simply computed by summing pixel values of the sum of the absolute values of the three high-band sub-images in a row and column, respectively. Furthermore, Haar filter-like projections are computed as in Viola and Jones [17] approach as additional feature vectors, which are obtained from differences of two sub-regions in the candidate region. The final feature vector for a face candidate region is obtained by concatenating all the horizontal, vertical and filter-like projections. These feature vectors are then classified using dynamic programming (DP) and support vector machine (SVM) based classifiers into face or non-face classes.

In the following sections, we specify a general block diagram of our face detection system where each block is briefly described for the techniques used in the implementation. We then compare the detection performance of the dynamic programming with support vector machines and currently available face detection methods after giving detailed information about dynamic programming and support vector machine based classifiers. Finally, a brief discussion is given at the end of the chapter.

2.1 The Algorithm

In this section, we present a human face detection scheme for frontal pose and upright orientation as shown in Fig. 2.1. After determining all possible face candidate regions in a given still image or video frame, each region is decomposed into its wavelet domain sub-images as shown in Fig. 2.3. Face candidate regions can be estimated based on color information in video as described in Section 2.1.1. All the detail (high frequency) information within local facial areas, e.g., edges due to eyes, nose and mouth, is obtained in high-high sub-image of the face pattern. This sub-image is similar to a hand-drawn face image, and in a given region, face patterns can be discriminated using this high-pass filtered sub-image. Other high-band sub-images can be also used to enhance the high-high sub-image. The wavelet domain processing is presented in Section 2.1.2. For a face candidate region, a feature vector is generated from wavelet domain sub-images using projections.

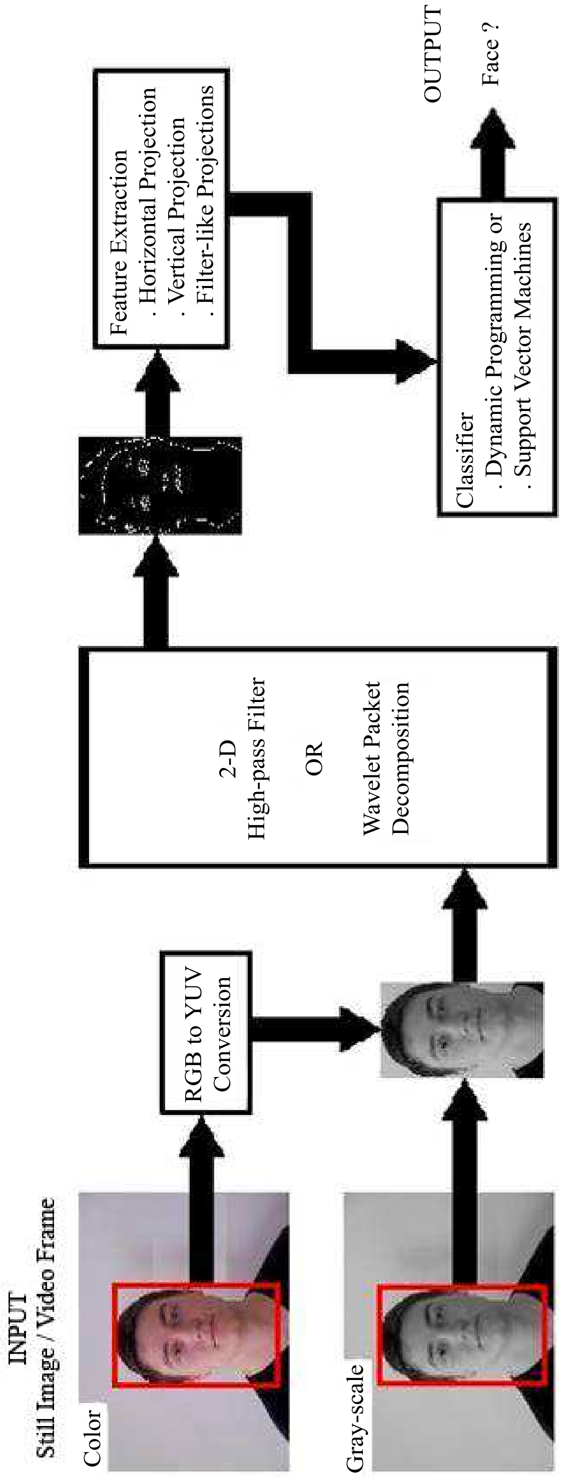


Figure 2.1: Block diagram of the face detection system.

Firstly, the generated feature vectors are classified using dynamic programming, which is an extensively studied and widely used tool in operations research for solving sequential decision problems in finite vocabulary speech recognition [31] and various communication theory applications, e.g., the Viterbi algorithm. Dynamic programming (DP) algorithm is generally used for computing the best possible alignment warp between a probe feature vector and a prototype feature vector, and the associated distortion between them. This property provides significant freedom on detecting faces which are slightly oriented in the horizontal and vertical directions. Two concepts are to be dealt with when using dynamic programming; the first one is the feature vector that the whole information of the pattern has to be represented in some manner, and the second one is the distance metric to be used in order to obtain a match path. The distance measure between a probe feature vector and a prototype feature vector is calculated using the *Euclidean distance* metric in this study. After evaluating the minimum alignment Euclidean distance between the probe feature vector and prototype feature vector in dynamic programming, a threshold value is used for classification of the probe feature vector. Dynamic programming based classifiers are reviewed in Section 2.1.4.1.

The second classification approach that we studied is support vector machines (SVMs), which are a brand new and powerful machine learning technique based on structural risk minimization for both regression and classification problems, although the subject can be said to have started in the late seventies by Vapnik [32, 33, 34]. SVMs have also been used by Osuna et al. [5] for detecting human faces in still images. While training SVMs, support vectors which define the boundary between two or more classes, are extracted. In our face detection case, the extracted support vectors define a boundary between two classes, namely face, labeled as $+1$, and non-face, labeled as -1, classes. The main idea behind the technique is to separate the classes with a surface that maximizes the margin between them [5]. It is obvious that, the main use of SVMs is in the classification block of our system, and it contributes the most critical part of this work. Section 2.1.4.2 presents a detailed description of support vector machine based classifiers.

2.1.1 Detection of Face Candidate Regions

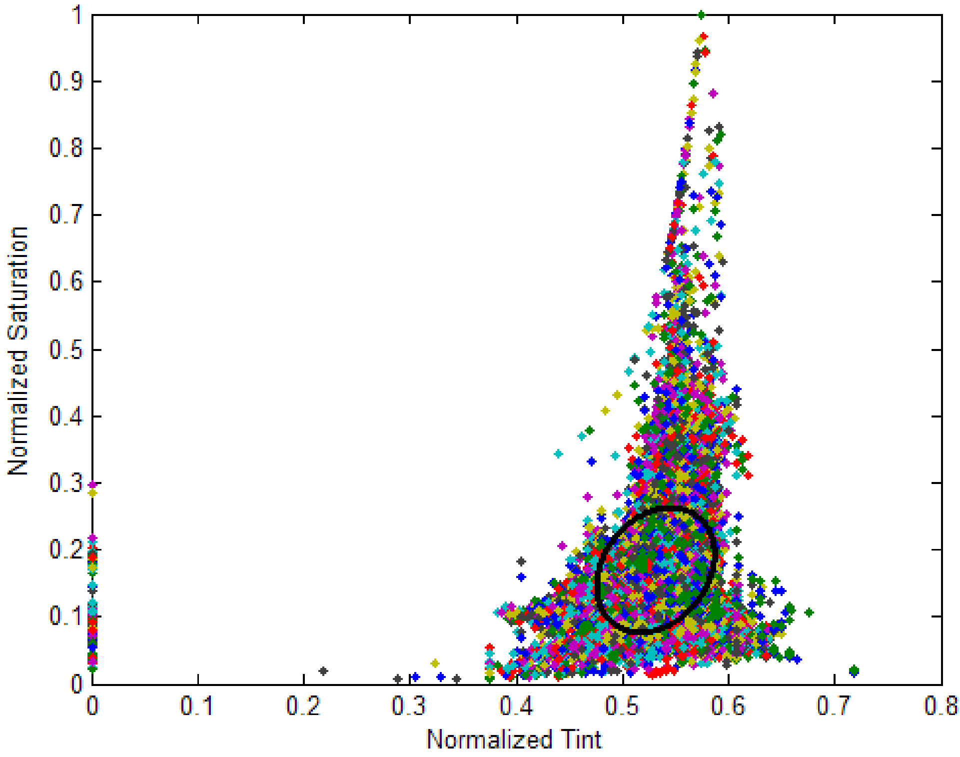
Human skin has a characteristic color which is shown to be a powerful fundamental cue for detecting human faces in images or video. It is useful for fast processing and also robust to geometric variations of the face patterns. Most existing human skin color modeling and segmentation techniques contain pixel-based skin detection methods which classify each individual pixel into skin and non-skin categories, independently from its neighbors. Kovac et al. [35] defined a very rapid skin classifier for clustering an individual pixel into skin or non-skin category through a number of rules in RGB (red, green, blue) color-space. Although it is a simple and rapid classifier, this method is very much subject to the illumination conditions because of the RGB color-space. Statistical skin chrominance models [36, 37, 38, 39] based on (Gaussian) mixture densities and histograms are also studied on different color-spaces for pixel-based skin color modeling and segmentation. On the other hand, region-based skin detection methods [40, 41, 42] try to take the spatial arrangements of skin pixels into account during the detection stage to enhance the methods performance [43].

In this study, especially for real-time implementation, the illumination effect is prevented using TSL (tint, saturation, luminance) color-space, and a pixel-based skin detection method is chosen for fast processing. A normalized chrominance-luminance TSL space is a transformation of the normalized RGB into more intuitive values, close to hue and saturation in their meaning [43].

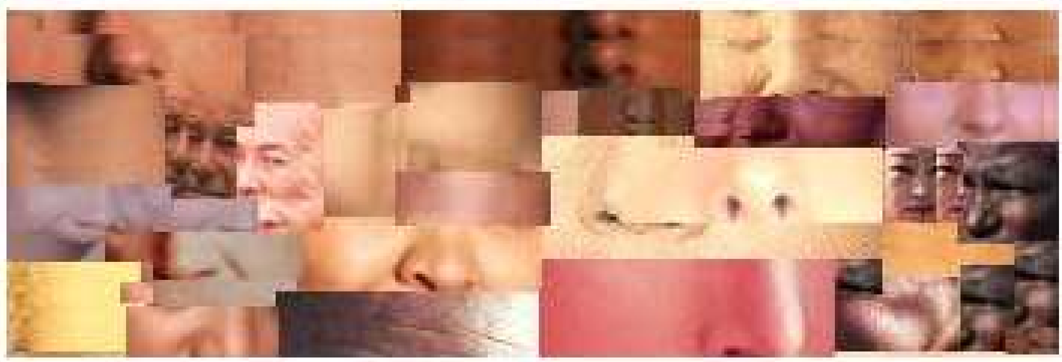
$$S = \sqrt{9/5\ (r'^2 + g'^2)} \tag{2.1}$$

$$T = \begin{cases} \arctan(r'/g')/2\pi + 1/4\,, & g' > 0 \\ \arctan(r'/g')/2\pi + 3/4\,, & g' < 0 \\ 0\,, & g' = 0 \end{cases} \tag{2.2}$$

$$L = 0.299R + 0.587G + 0.114B \tag{2.3}$$



(a)



(b)

Figure 2.2: (a) Single elliptical 2-D Gaussian classifier for (b) the training skin color samples' normalized T and S values.

where $r' = r - 1/3$, $g' = g - 1/3$ and r, g are normalized components of RGB color-space.

Given a color video frame or still image, each pixel is labeled as skin or non-skin using the pre-determined single elliptical Gaussian model as shown in Fig. 2.2-a with the training samples as shown in Fig. 2.2-b. Then morphological operations are performed on skin labeled pixels in order to have connected face candidate regions. The candidate regions' intensity images are then fed into a 2-D high-pass filter or a single stage 2-D rectangular wavelet decomposition block.

2.1.2 Wavelet Decomposition of Face Patterns

Possible face candidate regions are processed using a two-dimensional (2-D) filterbank. The regions are first processed row-wise using a 1-D filterbank with a low-pass and high-pass filter pair, $h[\,.\,]$ and $g[\,.\,]$ respectively. Resulting two image signals are processed column-wise once again using the same filterbank. The high-band sub-images that are obtained using a high-pass filter contain edge information, e.g., the low-high and high-low sub-images in Fig. 2.3 contain horizontal and vertical edges of the input image, respectively. Therefore, absolute values of low-high, high-low and high-high sub-images can be summed up to have an image having significant edges of the candidate region. Lagrange filterbank [44] consisting of the low-pass filter $h\,[n] = \{0.25,\, 0.5,\, 0.25\}$, and the high-pass filter $g\,[n] = \{-0.25,\, 0.5, -0.25\}$ is used in this study.

A second approach is to use a 2-D low-pass filter and subtract the low-pass filtered image from the original image. The resulting image also contains the edge information of the original image and it is equivalent to the sum of the undecimated low-high, high-low and high-high sub-images, which we call the *detail image*.

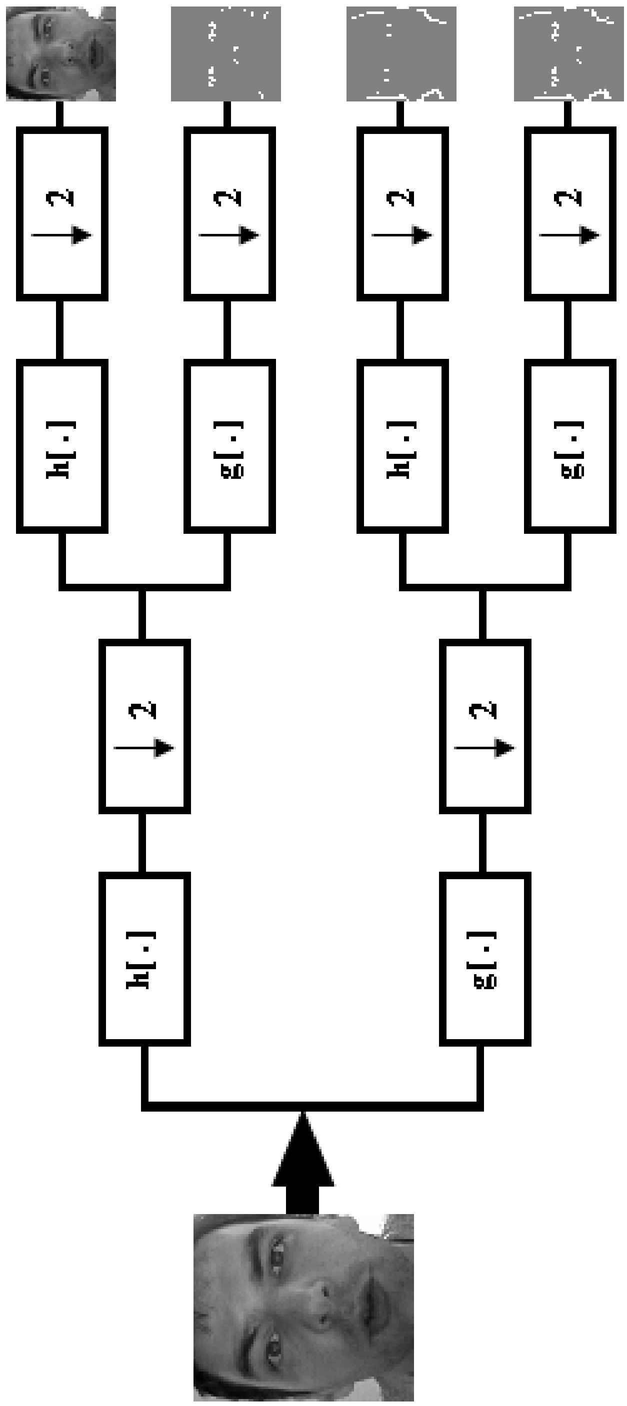


Figure 2.3: Two-dimensional (2-D) rectangular wavelet decomposition of a face pattern; low-low, low-high, high-low and high-high sub-images. $h[\,.\,]$ and $g[\,.\,]$ represent 1-D low-pass and high-pass filters, respectively.

2.1.3 Feature Extraction

In this thesis, edge projections of the candidate image regions are used as features. Edge information of the original image is available obtained using the wavelet analysis. The components of the feature vector are the horizontal, vertical and filter-like projections of the detail image. The advantage of the 1-D projection signals is that they can be easily normalized to a fixed size and this provides robustness against scale changes.

Horizontal projection $H[\,.\,]$ and vertical projection $V[\,.\,]$ are simply computed by summing the absolute pixel values, $d[\,.\,,\,.\,]$, of the detail image in a row and column, respectively as follows:

$$H[\,y\,] = \sum_{x} |d[\,x\,,\,y\,]| \tag{2.4}$$

$$V[\,x\,] = \sum_{y} |d[\,x\,,\,y\,]| \tag{2.5}$$

In this way, we take advantage of the detail (high frequency) information within local facial areas, e.g., edges due to eyes, nose and mouth, in both horizontal and vertical directions. These two edge projections actually provide significant discrimination ability for classification. Smoothed horizontal and vertical edge projection vectors of detail image of a typical human face are shown in Fig. 2.4, and two examples of non-face skin-color regions' detail images with their smoothed horizontal and vertical edge projections are shown in Fig. 2.5.

Furthermore, filter-like projections, $F_i[\,.\,]$, are computed similar to Viola and Jones [17] approach as additional feature vectors. We divide the detail image into two regions, R_1 and R_2 (i.e., vertical-cut filter-like feature regions), as shown in Fig. 2.6-a, and compute projections in these sub-regions. We subtract the horizontal projections in R_1 and R_2, and obtain a new horizontal projection vector $F_1[\,y\,]$. In this way, the symmetry property of a typical human face is considered as a feature also.

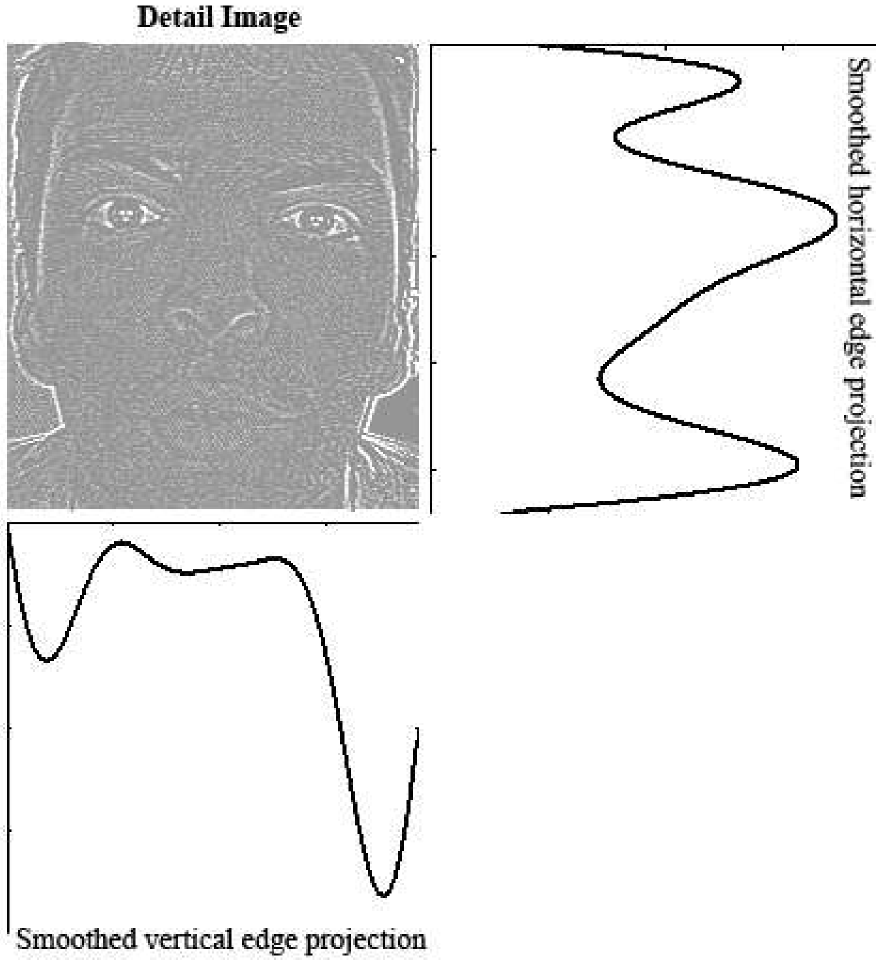


Figure 2.4: Smoothed horizontal and vertical edge projections of detail image of a typical human face.

$$F_1[\,y\,] = \left| \sum_{x \in R_1} |d[\,x\,,\,y\,]| - \sum_{x \in R_2} |d[\,x\,,\,y\,]| \right| \tag{2.6}$$

Because of the symmetry property of a face pattern, especially vertical-cut filter-like projections are very close to zero. Similarly, a new vertical projection vector $F_2[\,x\,]$ is computed as follows:

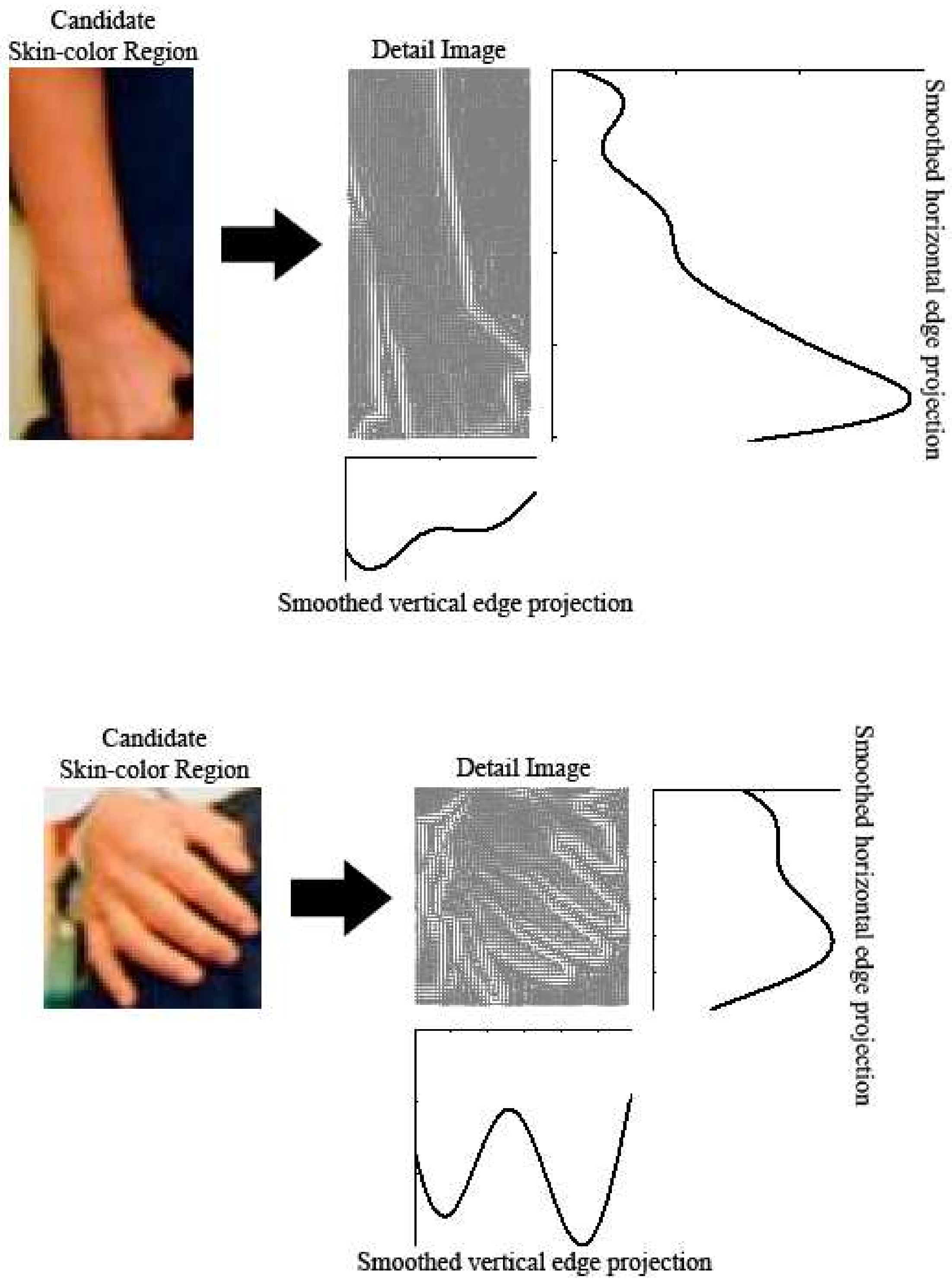


Figure 2.5: Two examples of non-face skin-color regions' detail images with their smoothed horizontal and vertical edge projections.

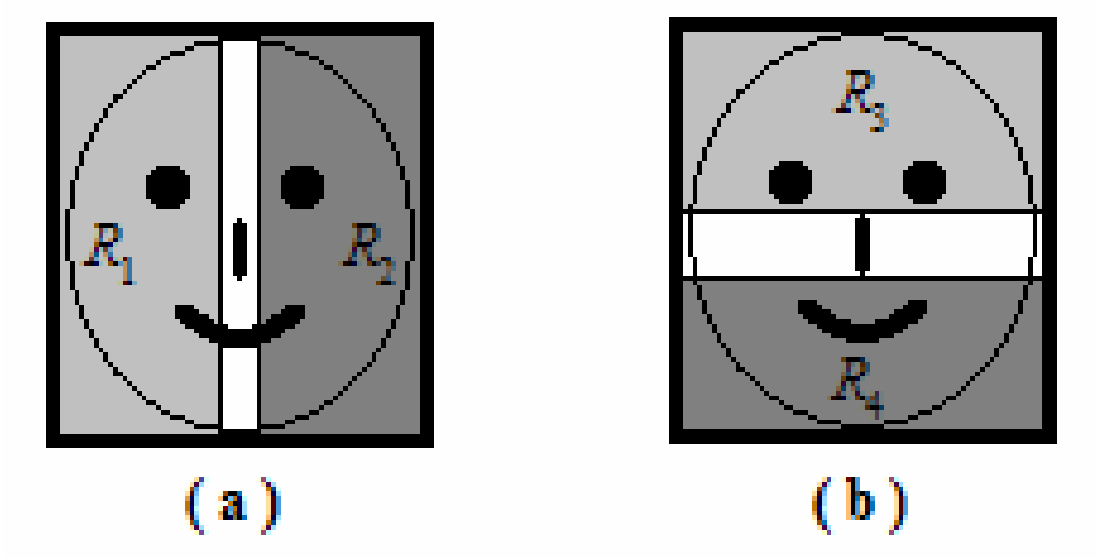


Figure 2.6: Two-rectangle Haar filter-like feature regions. White area between the feature regions is a 'dead-zone' in which no summation is carried out, (a) vertical-, (b) horizontal-cut of face candidate regions.

$$F_2[\,x\,] = \left| \sum_{y \in R_1} |d[\,x\,,\,y\,]| - \sum_{y \in R_2} |d[\,x\,,\,y\,]| \right| \tag{2.7}$$

We also repeat this process for regions R_3 and R_4 (i.e., horizontal-cut filter-like feature regions) shown in Fig. 2.6-b and obtain additional feature vectors.

Projection vectors are finally concatenated to obtain a composite feature vector. A composite feature vector consisting of the projections $H[\,.\,]$, $V[\,.\,]$ and $F_i[\,.\,]$ are used to represent a face candidate image region.

2.1.4 Classification Methods

After generating feature vectors, the face detection problem is reduced to a classification problem. The extracted feature vectors are classified using dynamic programming and support vector machine based classifiers into face or non-face classes. The first method that we used as a classifier is dynamic programming, which is used for measuring a distance metric between a probe feature vector and a typical prototype feature vector. The classification is applied by thresholding the resulting distance. The main reason that we use dynamic programming is that it produces better results than neural networks (NNs) and hidden Markov models (HMMs) in finite vocabulary speech recognition. The second method used is a support vector machine (SVM) based classifier. An important benefit of the support vector machine approach is that the complexity of the resulting classifier is characterized by the number of support vectors rather than the dimensionality of the transformed space [45]. Thus, SVMs compensate the problems of overfitting unlike some other classification methods.

2.1.4.1 Dynamic Programming

Dynamic programming (DP) is an extensively studied and widely used tool in operations research for solving sequential decision problems in finite vocabulary speech recognition [31] and various communication theory applications, e.g., the Viterbi algorithm. This technique is generally used in dynamic time-warping (DTW) algorithm for computing the best possible match path between a probe feature vector and a prototype feature vector, and the associated distortion between them. After determining a prototype (template) feature vector for a typical human face, similarity between template and probe feature vectors is measured by aligning them with distortion as shown in Fig. 2.7-a. The decision rule then classifies the probe feature vector with smallest alignment distortion. Similarity measure based on the Euclidean distance metric is as follows:

$$D_{1_m,2_n}\left(v_1^{(m)},\, v_2^{(n)}\right) \;=\; \sqrt{\left(v_1^{(m)} \,-\, v_2^{(n)}\right)^2} \tag{2.8}$$

where v_1 and v_2 are 1-D vectors not necessarily to be equal size.

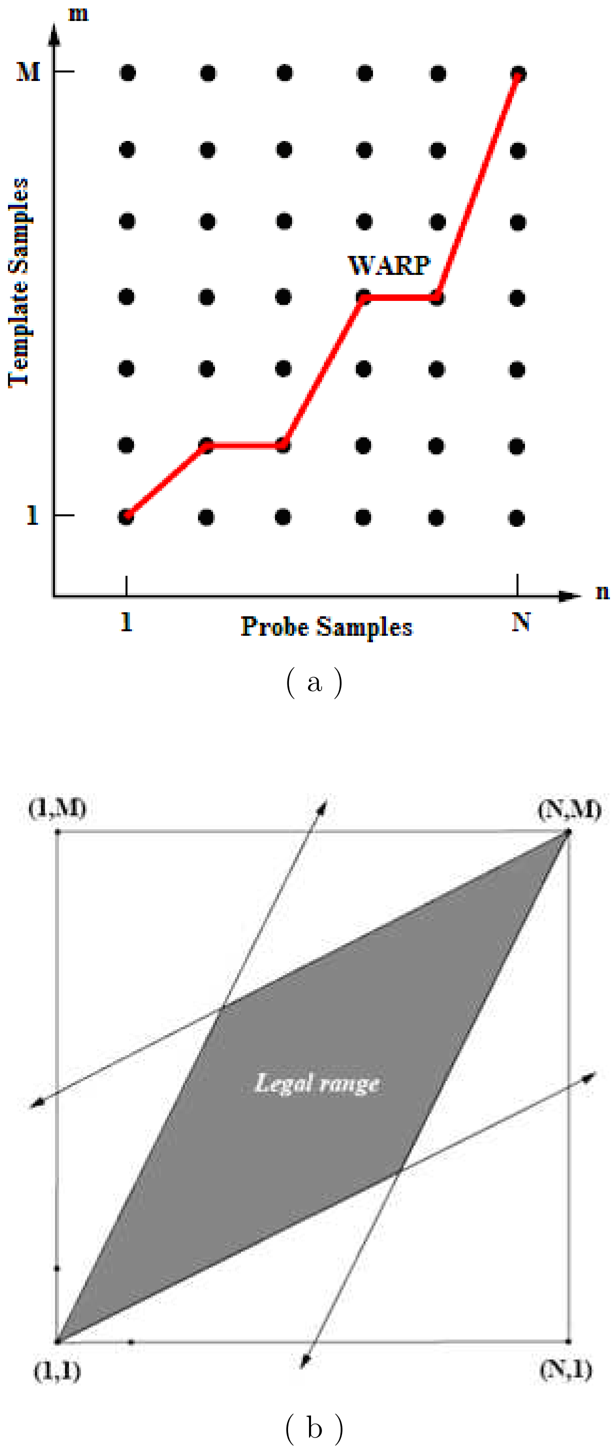


Figure 2.7: (a) Dynamic programming alignment example, (b) global constraints, range of allowable area for dynamic programming implementation.

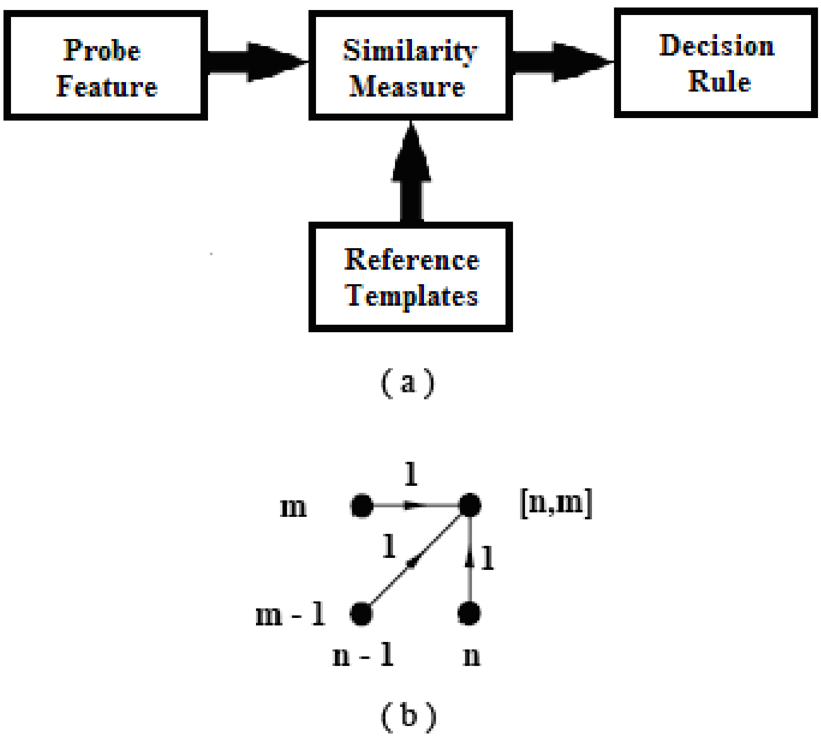


Figure 2.8: (a) Dynamic programming block diagram, (b) local constraints and slope weights used in our simulations.

There are two types of constraints on distance measure of dynamic programming: global and local. Global constraints are based on the overall computational difference between an entire signal and another signal of possibly different length. Global constraints exclude portions of search space as shown in Fig. 2.7-b. However, local constraints are based on the computational difference between a feature of one signal and a feature of the other. Local constraints determine the alignment flexibility (Fig. 2.8-b).

After evaluating the minimum alignment Euclidean distance between the probe feature vector and prototype feature vector in dynamic programming, a threshold value is used as a decision rule for classification of the probe feature vector. The threshold value used in our simulations is determined experimentally. A simple block diagram of dynamic programming technique is shown in Fig. 2.8-a.

There are several weaknesses of the dynamic programming algorithm. It has a high computational cost, i.e., it is not particularly fast. A distance metric must be defined, which may be difficult with different channels with distinct characteristics. Creation of the template vectors from data is non-trivial and typically is accomplished by pair-wise warping of training instances. Alternatively, all observed instances are stored as templates, but this is incredibly slow.

2.1.4.2 Support Vector Machines

Support vector machines (SVMs) seek to define a linear boundary between classes such that the margin of separation between samples from different classes that lie next to each other is maximized as shown in Fig. 2.9. Classification by SVMs is concerned only with data from each class near the decision boundary, called *support vectors*. Support vectors lie on the margin and carry all the relevant information about the classification problem. They are informally the hardest patterns to classify, and the most informative ones for designing the classifier [45]. This approach can be generalized to non-linear case by mapping the original feature space into some other space using a mapping function and performing optimal hyperplane algorithm in this dimensionally increased space. In the original feature space, the hyperplane corresponds to a non-linear decision function whose form is determined by the mapping kernel. Mapping kernels have been developed to compute the boundary as a polynomial, sigmoid and radial basis function.

While training the classifier, n-dimensional feature vectors are collected and their labels are manually determined to construct instance-label pairs $(x_i,\, y_i)$ where $x_i \in R^n$ and $y_i \in \{+1,\, -1\}$. An SVM classifier tries to determine an optimal solution of data separation by mapping the training data x_i to a higher dimension by a kernel function ϕ up to a penalty parameter C of the error term. In model training, the kernel function used is the radial basis function (RBF) in this study. RBF kernels are computed as follows:

$$K\,(x,\, y) \,=\, \exp\left(-\,\gamma\,||\,x_i \,-\, x_j\,||^2\right), \quad \gamma > 0 \tag{2.9}$$

It is very important to find the right parameters for C and γ in RBF. Therefore, five-fold cross validation with a grid search of $(C,\, \gamma)$ on the training feature set is applied to find the best parameters achieving the highest classification accuracy.

In this research, we used a library for support vector machines called LIBSVM [46] which is available online in either C++ or Java, with interfaces for MATLAB, Perl and Python. Our simulations are carried out in C++ with interface for Python using radial basis function as kernel. LIBSVM package provides the necessary quadratic programming routines to carry out classification. It also normalizes each feature by linearly scaling it to the range $[\,-1,\,+1\,]$, and performs cross validation on the feature set. This package also contains a multi-class probability estimation algorithm proposed by Wu et al. [47].

Support vector machines are used for isolated handwritten digit detection, object recognition, and also face detection by several researchers, e.g., Osuna et al. [5].

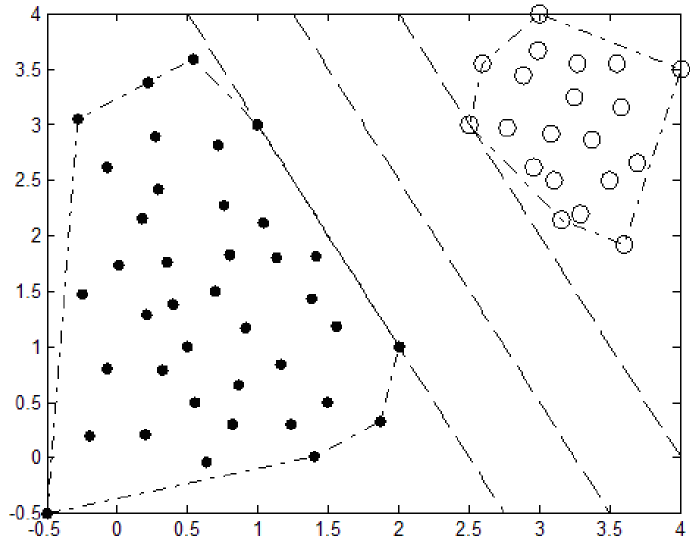


Figure 2.9: Decision boundary determination by SVMs using a linear kernel: Black dots indicate feature vectors of the first class and white circles indicate feature vectors of the second class, respectively. Lines (or hyperplanes in higher dimensions) separate the decision regions of the first and second class.

2.1.5 Experimental Results

The proposed human face detection algorithm is evaluated on several face image databases including the Computer Vision Laboratory (CVL)[1] Face Database. The database contains 797 color images of 114 persons. Each person has 7 different images of size 640x480 pixels: far left side view, 45^o angle side view, serious expression frontal view, 135^o angle side view, far right side view, smile -showing no teeth- frontal view and smile -showing teeth- frontal view. We extracted 335 frontal view face dataset from this database by cropping faces of variable size using color information. Furthermore, 100 non-face samples are extracted from color images downloaded randomly from the World Wide Web. These non-face samples include skin colored objects and typical human skin regions such as hands, arms and legs. The success rate achieved using dynamic programming is 95.80% over whole face and non-face datasets with concatenation of horizontal,

[1] More information is available on `http://www.lrv.fri.uni-lj.si/`

vertical and vertical-cut filter-like projections as feature vectors. After generating 11 face template vectors, each probe vector is compared to these face templates, and an experimentally determined threshold value is used in the decision block of dynamic programming. The classification is then carried out using majority voting technique. SVMs also applied to these datasets with same feature vectors and a success rate of 99.60% is achieved. While training SVMs, 100 face samples and 50 non-face samples used, and then these are also included in the test set.

The second experimental setup consists of a real-time human face dataset for real-time implementation. This dataset is collected in our laboratory. The dataset currently contains the video of 12 different people with 30 frames each. A person's face is recorded from 45^o side view to 135^o side view from different distances to camera with a neutral facial expression under the day-light illumination. SVMs are trained with these video data and the resulting *modal file* of LIBSVM package is used for classifying the test features in real-time. The proposed human face detection system is implemented in .NET C++ environment, and it works in real-time with approximately 12 fps on a standard personal computer. We used a Philips web camera with output resolution of 320x240 pixels throughout all our real-time experiments.

The third database that we used in our experiments is the Facial Recognition Technology (FERET) Face Database. We used the same dataset used by Uzunov et al. [19]. Some example face images from this dataset are shown in Fig. 2.10. This dataset contains 10556 gray-scale images of size 32x32 pixels face and non-face samples. There are 3156 face samples where each instance has a single sample of frontal upright human face. These images were collected from the FERET Face Database by [19], including human faces from all races with different face expressions, some wearing glasses, having beard and/or mustaches. The non-face dataset contains 7400 samples of random sampling images of size 32x32 of indoor or outdoor scenes which are collected randomly from the World Wide Web. The success rate achieved by SVMs using a variable threshold on edge images is 99.90%. Our detection rate is better than the best Haar wavelet packet dictionary test results on this dataset which is 99.74% with 150 atoms. Symmlet-2, Symmlet-3 and Symmlet-5 wavelet test results for 150 atoms are

Table 2.1: Face detection results.

Method	**Database**	**Success Rate**
Our method (edge projections with DP)	CVL	95.80%
Our method (edge projections with SVMs)	CVL	99.60%
Our method (edge projections with AdaBoost)	FERET	94.90%
Our method (edge projections with SVMs)	FERET	99.90%
Haar wavelet dictionary with 150 atoms	FERET	99.74%
Symmlet-2 with 150 atoms	FERET	99.25%
Symmlet-3 with 150 atoms	FERET	99.51%
Symmlet-5 with 150 atoms	FERET	99.47%

99.25%, 99.51% and 99.47%, respectively. We also tried an AdaBoost classifier on the edge images with the same feature vectors, and achieved 94.90% success rate. All the experimental results are given in Table 2.1.

In order to obtain a false positive rate comparison, we randomly collected 1000 face and 1000 non-face training samples to train SVMs and AdaBoost classifiers. And then, 1000 non-face test samples are tested with these classifiers accordingly. AdaBoost classifier has 11.30% false positive rate while SVMs has that of 2.40%. Some examples of used non-face images are shown in Fig. 2.11.

2.2 Discussion

In this chapter, we proposed a human face detection algorithm based on edge projections of face candidate regions in color images and video. A set of detailed experiments in both real-time and well-known datasets are carried out. The experimental results indicate that support vector machines work better than dynamic programming and AdaBoost classifiers for the proposed feature extraction method. Our algorithm is also better than any other wavelet domain methods including Haar wavelet dictionary, Symmlet-2, Symmlet-3 and Symmlet-5 with 150 atoms.

Figure 2.10: Examples of FERET frontal upright face images. Wavelet domain images are used for training and testing.

Figure 2.11: Examples of non-face images. Wavelet domain images are used for training and testing.

Chapter 3

Eye Localization System

In this chapter, we present a human eye localization system in images and video with the assumption that a human face region in a given still image or video frame is already detected by means of a face detector. Our method is basically based on the idea that eyes can be detected and localized from edges of a typical human face as in face detection algorithm in Chapter 2.

The proposed algorithm works with edge projections of given face images. After an approximate horizontal level detection, each eye is first localized horizontally using horizontal projections of associated edge regions. Horizontal edge profiles are then calculated on the estimated horizontal levels. Eye candidate points are determined by pairing up the local maximum point locations in the horizontal profiles with the associated horizontal levels. After obtaining eye candidate points, the verification is carried out by a support vector machine (SVM) based classifier. The locations of eyes are finally estimated according to the most probable point for each eye separately.

This chapter is organized as follows. We first describe our eye localization system where each step is briefly explained for the techniques used in the implementation. The experimental results of the proposed algorithm are then presented and the detection performance is compared with currently available eye localization methods. Finally, a brief discussion is given at the end of the chapter.

3.1 The Algorithm

In this section, we develop a human eye localization scheme for faces with frontal pose and upright orientation. After detecting a human face in a given color image or video frame using edge projections method as described in Chapter 2, the face region is decomposed into its wavelet domain sub-images. The detail (high frequency) information within local facial areas, e.g., edges due to eyes, nose and mouth, is obtained in low-high, high-low and high-high sub-images of the face pattern. The wavelet domain processing of a face pattern is discussed in Section 2.1.2.

After analyzing horizontal projections and profiles of horizontal- and vertical-crop edge images, the candidate points for each eye are detected as explained in Section 3.1.1. All the candidate points are then classified using a support vector machine based classifier. Finally, the locations of each eye are estimated according to the most probable ones among the candidate points.

3.1.1 Feature Extraction and Eye Localization

The first step of feature extraction is de-noising. The detail image of a given face region is de-noised by soft-thresholding using the method by Donoho and Johnstone [48]. The threshold value t_n is obtained as follows:

$$t_n = \sqrt{\frac{2\,\log(n)}{n}}\,\sigma \tag{3.1}$$

where n is the number of wavelet coefficients in the region and σ is the estimated standard deviation of Gaussian noise over the input image signal. The wavelet coefficients below the threshold are forced to become zero and those above the threshold are kept as are. This initial step removes the noise effectively while preserving the edges in the data.

The second step of the algorithm is to determine the approximate horizontal position of eyes using the horizontal projection in the upper part of the detail image as eyes are located in the upper part of a typical human face. An example frontal upright human face region is shown in Fig. 3.1-a, and its detail image is shown in Fig. 3.1-b. This provides robustness against the effects of edges due to neck, mouth (teeth) and nose on the horizontal projection. The index of the global maximum in the smoothed horizontal projection in this region indicates the approximate horizontal location of both eyes as shown in Fig. 3.1-d. By obtaining a rough horizontal position, the detail image is cropped horizontally according to the maximum as shown in Fig. 3.1-c. Then, vertical-crop edge regions are obtained by cropping the horizontally cropped edge image into two parts vertically as shown in Fig. 3.1-e.

The third step is to compute again horizontal projections in both right-eye and left-eye vertical-crop edge regions in order to detect the exact horizontal positions of each eye separately. The global maximum of these horizontal projections for each eye provides the estimated horizontal levels. This approach of dividing the image into two vertical-crop regions provides some freedom on detecting eyes in oriented face regions where eyes are not located on the same horizontal level.

Since a typical human eye consists of white sclera around dark pupil, the transition from white to dark (or dark to white) area produces significant jumps in the coefficients of the detail image. We take advantage of this fact by calculating horizontal profiles on the estimated horizontal levels for each eye. The jump locations are estimated from smoothed horizontal profile curves. An example vertical-crop edge region with its smoothed horizontal projection and profile are shown in Fig. 3.2. It is worth mentioning that, the global maximum in the smoothed horizontal profile signals is due to the transition both from white sclera to pupil and pupil to white sclera region. The first and last peaks correspond to outer and inner eye corners. Since there is a transition from skin to white sclera (or white sclera to skin) region, these peak values are small compared to those of white sclera to pupil (or pupil to white sclera) region. However, this may not be the case in some eye regions. There may be more (or less) than three peaks depending on the sharpness of the vertical-crop eye region and eye glasses.

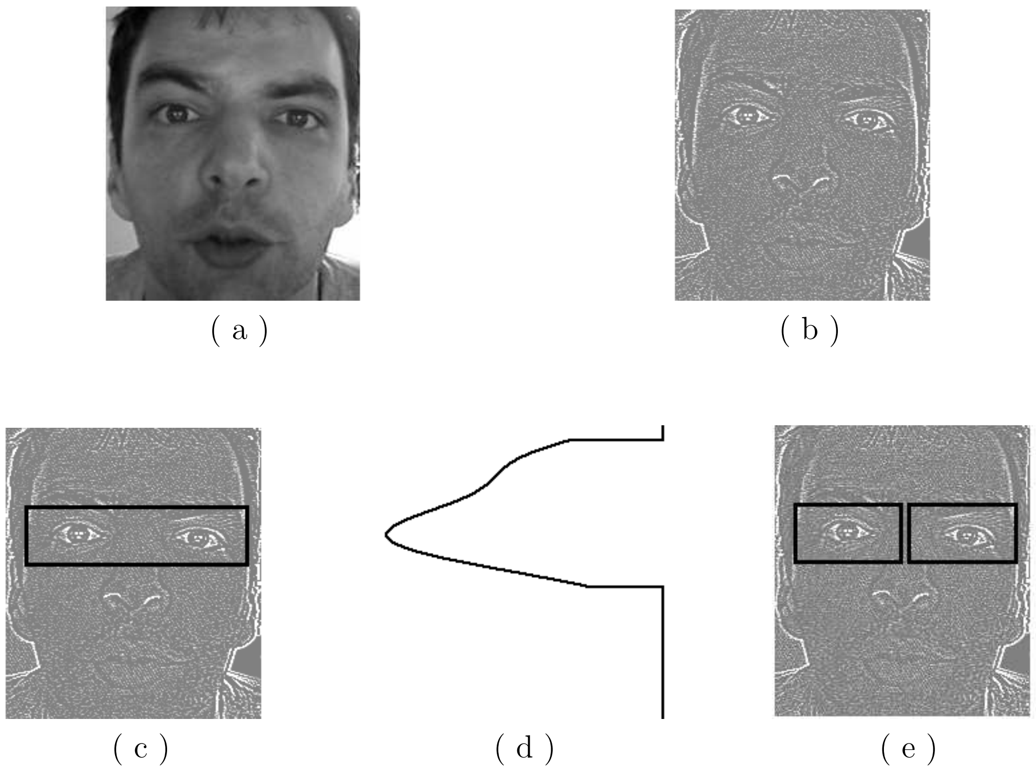


Figure 3.1: (a) An example face region with its (b) detail image, and (c) horizontal-crop edge image covering eyes region determined according to (d) smoothed horizontal projection in the upper part of the detail image (the projection data is filtered with a narrow-band low-pass filter to obtain the smooth projection plot). Vertical-crop edge regions are obtained by cropping the horizontal-crop edge image vertically as shown in (e).

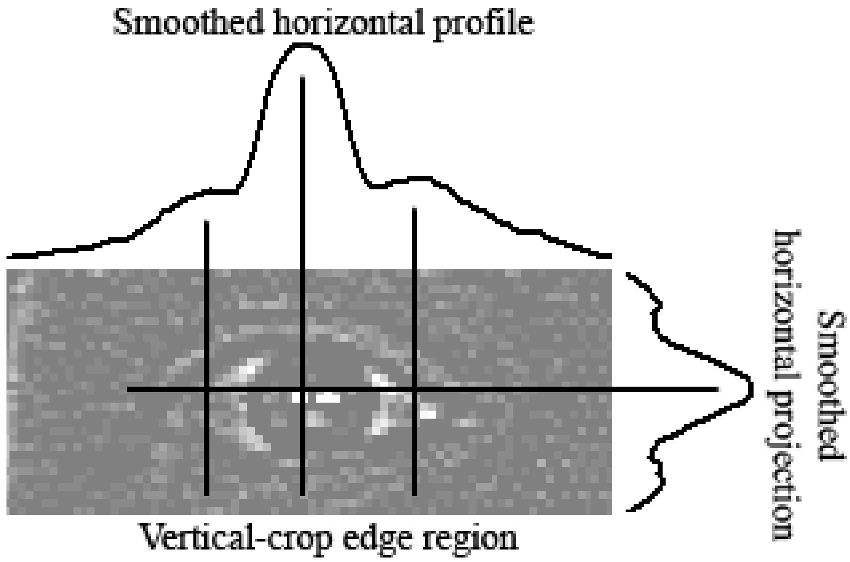


Figure 3.2: An example vertical-crop edge region with its smoothed horizontal projection and profile. Eye candidate points are obtained by pairing up the maximums in the horizontal profile with the associated horizontal level.

Eye candidate points are obtained by pairing up the indices of the maximums in the smoothed horizontal profiles with the associated horizontal levels for each eye. An example horizontal level estimate with its candidate vertical positions are shown in Fig. 3.2.

An SVM based classifier is used to discriminate the possible eye candidate locations. A rectangle with center around each candidate point is cropped and fed to the SVM classifier. The size of rectangles depends on the resolution of the detail image. However, the cropped rectangular region is finally resized to a resolution of 25x25 pixels. The feature vectors for each eye candidate region are calculated similar to the face detection algorithm by concatenating the horizontal and vertical projections of the rectangles around eye candidate locations. The points that are classified as an eye by SVM classifier are then ranked with respect to their estimated probability values [47] produced also by the classifier. The locations of eyes are finally determined according to the most probable point for each eye separately.

3.1.2 Experimental Results

The proposed eye localization algorithm is evaluated on the CVL and BioID[2] Face Databases in this chapter. All the images in these databases are with head-and-shoulder faces.

A detailed description of the CVL Face Database is given in Section 2.1.5. Since the developed algorithm can only be applied to faces with frontal pose and upright orientation, our experimental dataset contains 335 frontal view face images from this database. Face detection is carried out using edge projections method (as explained in Chapter 2) for this dataset since the images are color.

The BioID database consists of 1521 gray-scale images of 23 persons with a resolution of 384x286 pixels. All images in this database are the frontal view faces with a large variety of illumination conditions and face size. Face detection is carried out using Intel's OpenCV face detection method[3] since all images are gray-scale.

The estimated eye locations are compared with the exact eye center locations based on a relative error measure proposed by Jesorsky et al. [27]. Let C_r and C_l be the exact eye center locations, and $\tilde{C}_r$ and $\tilde{C}_l$ be the estimated eye positions. The *relative error* of this estimation is measured according to the formula:

$$d = \frac{\max\left(\| C_r - \tilde{C}_r \|, \| C_l - \tilde{C}_l \|\right)}{\| C_r - C_l \|} \tag{3.2}$$

In a typical human face, the width of a single eye roughly equals to the distance between inner eye corners. Therefore, half an eye width approximately equals to a relative error of 0.25. Thus, in this study we considered a relative error of $d < 0.25$ to be a correct estimation of eye positions.

[2] More information is available on http://www.bioid.com/

[3] More information is available on http://www.intel.com/technology/itj/2005/volume09issue02/art03_learning_vision/p04_face_detection.htm

Table 3.1: Eye localization results.

Method	**Database**	**Success Rate** (d<0.25)	**Success Rate** (d<0.10)
Our method (edge projections)	CVL	99.70%	80.90%
Our method (edge projections)	BioID	99.46%	73.68%
Asteriadis et al. [29]	BioID	97.40%	81.70%
Zhu and Geng [23]	BioID	94.81%	–
Jesorsky et al. [27]	BioID	91.80%	79.00%
Cristinacce et al. [26]	BioID	98.00%	96.00%

Our method has 99.46% overall success rate for $d < 0.25$ on the BioID database while Jesorsky et al. [27] achieved 91.80% and Zhu and Geng [23] had a success rate 94.81%. Asteriadis et al. [29] also reported a success rate 97.40% using the same face detector on this database. Cristinacce et al. [26] had a success rate 98.00% (we obtained this value from their distribution function of relative eye distance graph). However, our method reaches 73.68% for $d < 0.10$ while Jesorsky et al. [27] had 79.00%, Asteriadis et al. [29] achieved 81.70%, and Cristinacce et al. [26] reported a success rate 96.00% for this strict d value. All the experimental results are given in Table 3.1 and the distribution function of relative eye distances on the BioID database is shown in Fig. 3.3. Some examples of estimated eye locations are shown in Fig. 3.4.

The proposed human eye localization system with face detection algorithm is also implemented in .NET C++ environment, and it works in real-time with approximately 12 fps on a standard personal computer.

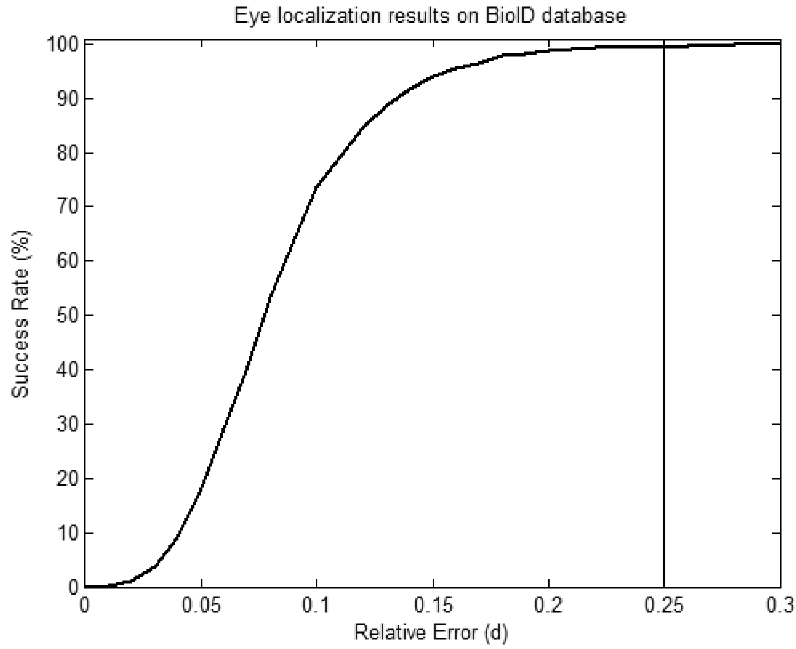


Figure 3.3: Distribution function of relative eye distances of our algorithm on the BioID database.

3.2 Discussion

In this chapter, we presented a human eye localization algorithm for faces with frontal pose and upright orientation. The performance of the algorithm has been examined on two face databases by comparing the estimated eye positions with the ground-truth values using a relative error measure. The localization results show that our algorithm is not affected by both illumination and scale changes since the BioID database contains images with a large variety of illumination conditions and face size. To the best of our knowledge, our algorithm gives the best results on the BioID database for $d < 0.25$.

(a)

(b)

Figure 3.4: Examples of estimated eye locations from the (a) BioID, (b) CVL Face Database.

Chapter 4

Conclusion

In this work, we presented wavelet domain signal and image processing algorithms for two specific applications: (a) Detection of human faces in images and video, and (b) localization of eyes in the detected face regions.

The proposed human face detection algorithm is based on edge projections of face candidate regions in color images and video. The algorithm segments skin color regions out, extracts feature vectors in these regions and detects frontal upright faces using dynamic programming or support vector machine based classifiers. A set of detailed experiments in both real-time and well-known datasets are carried out. The experimental results indicate that support vector machines work better than dynamic programming based classifiers for the proposed feature extraction method. This algorithm is better than any other wavelet domain methods including Haar wavelet dictionary, Symmlet-2, Symmlet-3 and Symmlet-5 with 150 atoms on the FERET Face Database. However, the current implementation is limited to the detection of frontal pose and upright orientation human faces. A possible and interesting extension would be expanding the proposed algorithm to include side-view faces as well.

An eye localization algorithm, which is based on the idea that eyes can be detected and localized from edges of a typical human face, is proposed assuming that a human face region in a given still image or video frame is already detected.

This algorithm also works with edge projections of given face images. The performance of the developed system has been examined on two face databases, i.e., CVL and BioID, by comparing the estimated eye positions with the ground-truth values using a relative error measure. The localization results show that the algorithm is not affected by both illumination and scale changes since the BioID database contains images with a large variety of illumination conditions and face size. To the best of our knowledge, the proposed eye localization algorithm gives the best results on the BioID database for $d < 0.25$. Therefore, it can be applied to human-computer interaction applications, and be used as the initialization stage of eye trackers. In eye tracking applications, e.g., Bagci et al. [49], a good initial estimate is satisfactory as the tracker further localizes the positions of eyes. For this reason, $d < 0.25$ results are more important than those of $d < 0.10$ from the tracker point of view.

The proposed human face detection and eye localization algorithms in this thesis provide little improvements over the previous methods appear in literature. This may not look that great at first glance but it is a significant improvement in a commercial application as it corresponds to one more satisfied customer in a group of hundred users.

Bibliography

[1] I. Pitas, *Digital Image Processing Algorithms and Applications.* John Wiley & Sons, Inc., NY, 2001.

[2] C. Heil and D. F. Walnut, eds., *Fundamental Papers in Wavelet Theory.* Princeton University Press, 2006.

[3] H. A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 23–38, 1998.

[4] K. K. Sung and T. Poggio, "Example-based learning for view-based human face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 39–51, 1998.

[5] E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: an application to face detection," in *CVPR '97: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 130–136, IEEE Computer Society, 1997.

[6] G. Guo, S. Z. Li, and K. Chan, "Face recognition by support vector machines," in *FG '00: Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 196–201, IEEE Computer Society, 2000.

[7] A. V. Nefian and M. H. Hayes, "Face detection and recognition using hidden markov models," in *ICIP '98: Proceedings of the IEEE International Conference on Image Processing*, vol. 1, pp. 141–145, 1998.

[8] F. Samaria and S. Young, "HMM based architecture for face identification," *Image and Computer Vision*, vol. 12, no. 8, pp. 537–543, 1994.

[9] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

[10] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *CVPR '91: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586–591, IEEE Computer Society, 1991.

[11] E. Hjelmas and B. K. Low, "Face detection: a survey," *Computer Vision and Image Understanding 83*, pp. 236–274, 2001.

[12] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: a literature survey," *ACM Computing Surveys*, vol. 35, no. 4, pp. 399–458, 2003.

[13] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, 1989.

[14] I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis," *IEEE Transactions on Information Theory*, vol. 36, no. 5, pp. 961–1005, 1990.

[15] C. Garcia and G. Tziritas, "Face detection using quantized skin color regions merging and wavelet packet analysis," *IEEE Transactions on Multimedia*, vol. 1, no. 3, pp. 264–277, 1999.

[16] Y. Zhu, S. Schwartz, and M. Orchard, "Fast face detection using subspace discriminant wavelet features," in *CVPR '00: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 636–642, IEEE Computer Society, 2000.

[17] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *CVPR '01: Proceedings of the IEEE Conference on*

Computer Vision and Pattern Recognition, vol. 1, pp. 511–518, IEEE Computer Society, 2001.

[18] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, pp. 119–139, 1997.

[19] V. Uzunov, A. Gotchev, K. Egiazarian, and J. Astola, "Face detection by optimal atomic decomposition," in *Proceedings of the SPIE: Mathematical Methods in Pattern and Image Analysis*, vol. 5916, pp. 160–171, SPIE, 2005.

[20] S. P. Lee, *Facial Animation System with Realistic Eye Movement Based on a Cognitive Model for Virtual Agents*. PhD thesis, Computer and Information Science, University of Pennsylvania, 2002.

[21] Z. Zhu and Q. Ji, "Robust real-time eye detection and tracking under variable lighting conditions and various face orientations," *Computer Vision and Image Understanding*, vol. 98, pp. 124–154, 2005.

[22] C. H. Morimoto and M. R. M. Mimica, "Eye gaze tracking techniques for interactive applications," *Computer Vision and Image Understanding*, vol. 98, pp. 4–24, 2005.

[23] Z. H. Zhou and X. Geng, "Projection functions for eye detection," *Pattern Recognition*, vol. 37(5), pp. 1049–1056, 2004.

[24] J. Wu and Z. H. Zhou, "Efficient face candidates selector for face detection," *Pattern Recognition*, vol. 36(5), pp. 1175–1186, 2003.

[25] W. Huang and R. Mariani, "Face detection and precise eyes location," in *ICPR '00: Proceedings of the International Conference on Pattern Recognition*, vol. 4, pp. 722–727, IEEE Computer Society, 2000.

[26] D. Cristinacce, T. Cootes, and I. Scott, "A multi-stage approach to facial feature detection," in *BMVC '04: Proceedings of the 15th British Machine Vision Conference*, pp. 277–286, 2004.

[27] O. Jesorsky, K. J. Kirchberg, and R. W. Frischholz, "Robust face detection using the hausdorff distance," in *AVBPA '01: Proceedings of the Third International Conference on Audio- and Video-based Biometric Person Authentication*, vol. 2091, pp. 90–95, Springer, Lecture Notes in Computer Science, 2001.

[28] W. Rucklidge, *Efficient Visual Recognition Using the Hausdorff Distance*, vol. 1173. Springer, Lecture Notes in Computer Science, 1996.

[29] S. Asteriadis, N. Nikolaidis, A. Hajdu, and I. Pitas, "An eye detection algorithm using pixel to edge information," in *ISCCSP '06: Proceedings of the Second IEEE-EURASIP International Symposium on Control, Communications, and Signal Processing*, IEEE, 2006.

[30] A. E. Cetin and R. Ansari, "Signal recovery from wavelet transform maxima," *IEEE Transactions on Signal Processing*, vol. 42, pp. 194–196, 1994.

[31] L. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition.* Prentice-Hall, Inc., NJ, 1993.

[32] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *COLT '92: Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pp. 144–152, 1992.

[33] C. Cortes and V. N. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[34] V. N. Vapnik, *The Nature of Statistical Learning Theory.* Springer-Verlag New York, Inc., 1999.

[35] J. Kovac, P. Peer, and F. Solina, "Human skin colour clustering for face detection," in *EUROCON '03: Proceedings of the IEEE Region 8 Computer as a Tool*, vol. 2, pp. 144–148, 2003.

[36] J. C. Terrillon, M. N. Shirazi, H. Fukamachi, and S. Akamatsu, "Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images," in *FG*

'00: Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition, pp. 54–61, IEEE Computer Society, 2000.

[37] H. Greenspan, J. Goldberger, and I. Eshet, "Mixture model for face-color modeling and segmentation," *Pattern Recognition Letters*, vol. 22, no. 14, pp. 1525–1536, 2001.

[38] M. J. Jones and J. M. Rehg, "Statistical color models with application to skin detection," *International Journal of Computer Vision*, vol. 46, no. 1, pp. 81–96, 2002.

[39] S. L. Phung, A. Bouzerdoum, and D. Chai, "Skin segmentation using color pixel classification: analysis and comparison," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 1, pp. 148–154, 2005.

[40] M. H. Yang and N. Ahuja, "Detecting human faces in color images," vol. 1, pp. 127–130, 1998.

[41] H. Kruppa, M. A. Bauer, and B. Schiele, "Skin patch detection in real-world images," in *DAGM '02: Proceedings of the 24th Annual Symposium on Pattern Recognition*, Springer, Lecture Notes in Computer Science.

[42] B. Jedynak, H. Zheng, M. Daoudi, and D. Barret, "Maximum entropy models for skin detection," Springer, Lecture Notes in Computer Science, 2003.

[43] V. Vezhnevets, V. Sazonov, and A. Andreeva, "A survey on pixel-based skin color detection techniques," in *Proceedings of the Graphicon '03*, 2003.

[44] C. W. Kim, R. Ansari, and A. E. Cetin, "A class of linear-phase regular biorthogonal wavelets," in *ICASSP '92: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, pp. 673–676, 1992.

[45] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification.* Canada: John Wiley & Sons, Inc., 2001.

[46] C. C. Chang and C. J. Lin, *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[47] T. F. Wu, C. J. Lin, and R. C. Weng, "Probability estimates for multi-class classification by pairwise coupling," *The Journal of Machine Learning Research*, vol. 5, pp. 975–1005, 2004.

[48] D. L. Donoho and I. M. Johnstone, "Ideal spatial adaptation via wavelet shrinkage," *Biometrika*, vol. 81, pp. 425–455, 1994.

[49] A. M. Bagci, R. Ansari, A. Khokhar, and A. E. Cetin, "Eye tracking using markov models," in *ICPR '04: Proceedings of the 17th International Conference on Pattern Recognition*, vol. 3, pp. 818–821, IEEE Computer Society, 2004.