SDA-Based discrete head pose estimation

H.B. Oost
November 2009
Masters Thesis

Human Media Interaction
University of Twente

Examination committee:
dr. M. Poel
dr.ir. R.W. Poppe
dr.ir. D. Reidsma

Abstract

The estimation of head position and orientation is an important building stone in many applications of Human-computer interaction. This thesis presents two variations of a monocular image head pose estimator based on Subclass Discriminant Analysis (SDA). The use of subclasses enables the application of discriminant analysis to a wider variety of high-dimensional classification problems. The difficulty in applying SDA is in determining the optimal division of the data into subclasses.

For a selected number of discrete poses, a specialised one-versus-all classifier is generated using a boosting procedure applied to feature selection. The one-versus-all classifiers are combined into a discrete head pose estimator. This approach is compared to a multi-class approach using the information learned while training the separate one-versus-all classifiers. The performance of these two approaches is evaluated on the Pointing'04 dataset and compared to the performance of the more widely used Linear Discriminant Analysis (LDA) approach.

The results show that the image features selected using the boosting procedure are similar to those that would be selected using a face mask. The multi-class approach is shown to be preferable over the one-versus-all approach. Additionally, the SDA classifiers are shown to have performance characteristics comparable to those of LDA for both approaches.

Contents

Part I

Introduction

Chapter 1

Introduction

The development of computer vision systems that rival human vision and recognition has, over the course of more than 40 years, proven to be a difficult task. During the approximately 40 years of research the availability of computing power and digital cameras has increased manifold. Simultaneously there is a growing interest in biometric identification and alternative human-computer interaction techniques. The result is a rising interest in face identification and pose estimation in diverse research fields such as image processing, pattern recognition, computer graphics and psychology. Still, there are limitations to the current state of the art and there are many remaining challenges.

This research considers the problem of estimating the pan and tilt angles of a person's head as shown in a monocular image. The next sections introduce the concept of head pose and discuss the applications for head pose estimation. We conclude the chapter by stating the questions for this research.

1.1 Head pose

Consider a situation in which a static camera is used to take images of a person who is allowed to move and rotate freely in a number of directions. Head pose estimation is concerned with only pan and tilt rotations, these are illustrated in figure 1.1.

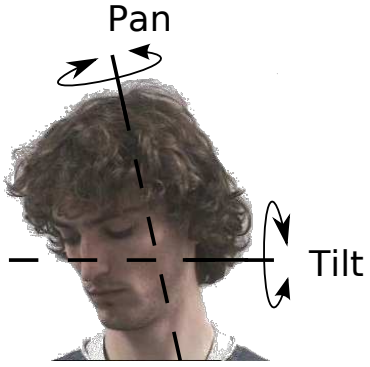


Figure 1.1: The two rotations, pan and tilt, relevant to the head pose estimation domain.

The subject's movement determines the location of his head within the captured image. The roll rotation affects the image differently than the pan and tilt rotations. If the subject performs a roll rotation, the appearance of his head would be unchanged, were as with pan and tilt rotations we see a different side of the subject's head. This is referred to as an in-plane rotation Estimating the location and roll rotation of a head in an image is generally the task of a face detection system and not of a head pose estimation system.

In this research, only the out-of-plane rotations of pan and tilt are considered. Both these rotations result in drastic changes within the image. As the head tilts or pans facial features move in or out of the image, the outline of the head changes and light reflections and shadows move across the face. These

deformations result in non-linear transformations of the image which makes pose estimation through computer vision a complex task.

1.2 Overview of head pose systems

Regardless of the large variation in head pose estimation systems, most of these systems can be divided into the same three stages shown in figure 1.2. The first stage is to detect the presence and location of a head in an image. This can be done by simple methods such as color based head detection (chapter 4) or even by the repeated application of a head pose estimation system (chapter 2). The second stage is to create a description from the image that is suitable for classification. There is a wide range of suitable descriptors. The most basic form consists of the pixel values of the image but more complex variations exist and are discussed in section 2.2. The choice of descriptors is dependent on the classification method used in the final stage. Different methods are discussed in 2.3. Because the complete set of poses available to a person are continuous and ordered, pose estimation can be considered a regression problem. However, in many systems, the ranges of poses are divided into discrete classes and head pose estimation is considered as a classification problem. For the system developed in this research we will consider the head pose estimation as a discrete classification problem.

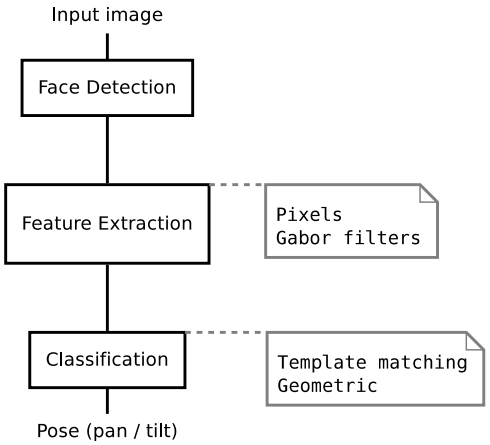


Figure 1.2: Overview of a generic pose estimation system, listing variations for the latter two stages.

1.3 Applications for head pose estimation

For successful human computer interaction to move beyond the keyboard and mouse we will likely require a multi-modal approach relying not only on head pose, but also hand gestures, speech or even brain signals. Head pose estimation is just one of these modals but it has a few specific applications as well.

Head pose assists in estimating people's gaze and focus-of-attention. This is not only important for multi-modal interfaces but also has commercial applications such as monitoring the attention given to advertisements.

The tracking of head pose over time allows the interpretation of head gestures. Besides applications in multi-modal interfaces, head pose tracking has also been used for detecting drowsiness in drivers.

There are numerous existing biometric identification methods that are easier to perform than face identification. But methods such as fingerprint analysis and retinal scans require the cooperation of the subject. Face identification, however, is a passive metric which requires no special actions by the user and can be performed outside of the controlled environments required for other biometric identification methods. Such systems would allow the identification and tracking of individuals through existing video surveillance systems. But to perform face identification under these circumstances we require a head pose estimation system.

This research concentrates on pose estimation using monocular camera images that could potentially be acquired by a basic camera. This type of pose estimation can be used for real-time pose estimation for use with webcameras or it could be used for pose estimation in photographs. This pose information can subsequently added to historical archives and other multimedia databases.

1.4 Research questions

As we will discuss in chapter 2.3, some very successful systems use a variation of discriminant analysis to perform the head pose classification. In chapter 6 we will discuss a recent variation named Subclass Discriminant Analysis, or SDA, developed by Zhu and Martinez[72]. We will apply SDA to the head pose estimation task in two variations; as an array of binary classifiers and as a multi-class classifier. Furthermore, we use the well known Gabor filter[50] in combination with GentleBoost[18] to create a compact image description which is further described in chapter 5.

In this thesis we attempt to answer the following questions regarding the application of GentleBoost and SDA to head pose estimation.

1. What are the differences in classifying the discrete poses:
   (a) Does the GentleBoost feature selection approach provide valid features for each pose?
   (b) Do the number of features required for optimal performance differ between pose classes?
   (c) Should we create a different division of subclasses for each pose class?
2. Does SDA offer a significant improvement over LDA methods in discrete head pose estimation:
   (a) for use as a one-versus-all classifier?
   (b) for use as a multi-class classifier?

Chapter 2

Overview of related work

Many of the developed systems related to head pose estimation are applicable in multiple domains, such as face recognition, and often overlap in their use of image features or statistical methods. Both Zhao et.al.[68] (of which an extended version is available as the introduction to [69]), Murphy-Chutorian and Trivedi[41] and Zhang and Gao[67] provide excellent surveys covering a wide range of systems. The work discussed here focuses on pose estimation from monocular 2D images but there are other domains with different use cases and corresponding hardware requirements such as 3D imaging techniques.

As discussed in the introduction in section 1.2, the process of head pose estimation can be divided into three stages: head detection, feature extraction and classification. We start this chapter with a summary of the available face databases which can be used for training and evaluating head pose detection systems. The next two sections discuss the variations in the second, feature extraction, and third, classification, stages of a generic pose estimation system.

2.1 Existing datasets

Heads and faces are three dimensional objects whose appearance is affected by identity, pose, expression, age, occlusion, illumination, hair and other factors. Most methods require significant numbers of training samples with variations of these factors in order to be robust against such variations. Additionally the increasing accuracy of developed methods require increasingly large test sets in order to reliably estimate and compare their accuracy.

The collection of large datasets with controlled variations over many factors is a resource-intensive task which a number of researchers have undertaken. Table 2.1 provides an overview of popular and recent facial databases. Another overview can be found in chapter 13[23] of the Handbook of Face Recognition[31] for most of the datasets released before 2005.

Name	Year	# Samples	# Subjects	# poses	Pan	Tilt	Miscellaneous
AR Face Database[37]	1998	$4,000+$	126	1	$0°$	$0°$	expressions, attributes
FERET[44]	1998	14,126	1,199	9–20	$\pm 90°$	$0°$	
XM2VTSDB[38]	1999	-	295	continuous	$\pm 90°$	$\pm 60°$	video, 3D model of faces
Yale Face Database B[20]	2001	16128	628	9	$\pm 24°$	$0°$	varying illumination conditions
CMU PIE[53]	2002	41,368	68	13	$-66°$– $+62°$	-	illumination, expressions
FacePix(30)[9, 33]	2002	16,290	30	181	$\pm 90°$	$0°$	varying illumination conditions
BANCA[3]	2003	6,240	208	1	0	0	varied conditions
Pointing'04[21]	2004	2,790	15	93	$\pm 90°$	$\pm 90°$	
3D FRGC[43]	2005	50,000	466	-	-	-	multiple datasets, 3D range data
IDIAP[2]	2005	2 hours	16	continuous	$\pm 60°$	$-60°$– $+15°$	video, natural movement
MMI Facial Expression[42]	2005	$1,500+$	19	2	$0°$,$90°$	$0°$	emotions, facial action units
CMU Multi-PIE[22]	2008	750,000	337	15	$\pm 90°$	-	illumination, expressions
HR 3D Expression[66]	2008	60,600	101	-	-	-	dynamic 3D range data, expressions
GENKI[28]	2009	7,172	$\approx 4,000$	-	-	-	uncontrolled environments, two datasets

Besides 2D images and video, there is increasing interest in the use of 3D range data for face identification and expression classification. Datasets using static range data have become available in recent years[43]. Yin et.al.[66] have developed an extensive facial expression database using dynamic 3D range data. In the remainder of this chapter we will limit the discussion to 2D image data.

2.2 Feature extraction

Before we can attempt classification of head poses from 2D images we need to represent the image data in a form which we can subject to statistical analysis. In the generic pose estimation system shown in figure 1.2, this would be the second stage. For this section we divide the image descriptors into two groups: image transformations over the whole image and descriptors which represent local salient features. This corresponds roughly to the two major categories of systems, the holistic template matching systems and geometrical local feature based systems. In many cases, the number of features extracted from the sample images is too large in relation to the number of available samples to allow reliable classification. We can reduce the dimensionality of the feature vector by selecting the most useful elements (2.2.3).

(a) Original image (b) Sobel edge filter

Figure 2.1: A sample image before and after the application of various filters.

2.2.1 Color and texture features

Early systems primarily used the image intensity values, the pixels, of digital images and these form the basis for a number of variations collectively referred to as holistic template matching systems. The classification performance can benefit from image processing techniques to negate differing illumination conditions[24] or, in the case of pose estimation, to decrease the identity specific differences by applying a Gaussian blur filter.

Alternatively, image operations can be applied to emphasize facial features. Each individual image pixel has a comparatively low information density and many systems improve their performance by using features which can represent salient structures within the image. These region features, or texture features, use a single descriptor to represent each pixel in the original image together with the neighborhood surrounding this pixel. Haar features are widely used, largely due to the popularity of the boosted cascade face detector by Viola and Jones[60, 61]. One of the most prominent texture features used within image recognition tasks are Gabor features[50], which are discussed further in section 5.1.

A large problem with colour and texture features is the very high number of features to represent a single image, as a result they are often paired with feature selection techniques[61] or dimensionality reduction techniques, such as PCA (EigenFaces[58]) and LDA (Fisherfaces[5].

2.2.2 Edge and point features

Edge features are most often a binary image showing only the parts of the original image with a high gradient. The numerous edge detectors differ in the gradient operator (kernel) used during the image convolution. The quality of the located edges are sensitive to the image quality but can be improved by using a slightly larger gradient operator or by including a Gaussian blur filter. The most used edge detectors include the Robert, Sobel (shown in figure 2.1(b), Prewitt, Canny and Krisch operators[1].

While edge features attempt to extract lines, point features focus on the points where edges intersect, examples of these are Moravec[40], Harris[25] and SUSAN[54] features. For which the more recent Scale Invariant Image Features (SIFT)[36] are popular, especially for real-time tracking of arbitrary surfaces,

and like the above have been used for face recognition tasks[8].
Point features remain popular in part due to the increased processing power which allows for realtime computation and comparison of an increasingly larger number of these features for use in realtime tracking of arbitrary surfaces.

2.2.3 Feature selection and sampling masks

For most classification algorithms the computational requirements increase dramatically with the dimensionality of the feature set. Although the computational cost can be overcome by advances in technology, as is evident by the increasing resolution of images in datasets and the increasing numbers of point features used in most SIFT based systems, the curse of dimensionality as described by Bellman[6] has severe consequences for statistical analysis which requires the number of samples to grow proportionally to the square of the dimensionality.

One approach is to apply domain knowledge to limit the number of features that are gathered. To perform the classification task we do not need to use any part of the image that represents the background, thus the image can be cropped to at least the smallest rectangle encompassing the head area and possibly just the face area[13]. Within this rectangle, not all features are equally useful and a density sampling mask can be applied. This mask selects more features from areas which are deemed to be most important to the classification and fewer, or none, from other areas.

Such a density sampling mask can be manually shaped according to the needs of the classification domain. If there is a need for multiple classifiers, such as for discrete pose estimation or view-independent face recognition it can be convenient to automatically generate the sampling masks. The dimensionality reduction methods discussed in the next section have been used for this purpose[35] as have boosting methods [56]. Boosting methods, such as AdaBoost[18] or GentleBoost[46] iteratively create a set of weak classifiers one feature at a time. They have been successfully applied in multiple systems[57, 32, 56] including the well known face detector by Viola and Jones[61]. Boosting procedures are discussed in more detail in section 5.1.

A second approach is to create multiple classifiers each trained on a subset of the available features, such as the approach by Wu and Trivedi[64] which trains one classifier for each scale of the Gabor wavelets used. This approach combines the advantages of a smaller feature space with the advantages of bagging[10].

Other approaches such as Face Bunch Graphs focus solely on the areas around specific facial features. These methods are among the flexible methods discussed in section2.3.3.

2.3 Classification

Classification methods are often divided in continuous and discrete methods. Some methods, such as the discriminant analysis method used in chapter 6, can be applied as a discrete detector array or as a continuous classifier. The first section discusses the dimensionality reduction methods, the next section discusses discrete pose recognition. For these methods, the classification stage can often be applied independently of the type of features used to describe the image. The geometric methods discussed in the final section explicitly take the physical properties of a face in account.

2.3.1 Dimensionality reduction

The feature selection techniques mentioned in section 2.2.3 heavily rely on domain knowledge or require manual intervention. Dimensionality reduction is an automatic process which requires no domain knowledge and creates a combination of the features that best explain the data. This linear combination of features results in a subspace with a lower dimensionality and, depending on the method used, can have additional benefits such as robustness against changes in illumination and increased separation of positive and negative samples.

Turk and Pentland applied PCA on the image intensity values to create Eigenfaces[58] while Belhumeur et.al use LDA to create Fisherfaces[5]. The subspace created by LDA has the additional benefit of separating the different sample classes. Numerous variations on LDA exist and have been applied for

general recognition tasks and pose estimation tasks, examples are SRDA[48] and SDA[72]. The latter is discussed in more detail in chapter6.

These methods are linear but the classification of head pose is a non-linear task. One widely used method to overcome this limitation is to map the samples onto a higher-dimensional space before applying the linear classification method. This "kernel trick" has resulted in KPCA[47], KLDA[39], KSDA[12] and many more kernelized variants of dimensionality reduction techniques. Other non-linear methods include Isomap and Locally Linear Embedding[41].

An interesting property of these methods is the ability to learn a subspace in which samples with similar poses are placed near each other on a non-linear manifold. Manifold methods hold a lot of potential for continuous pose estimation[4]. But to learn the subspace and manifold correctly these methods require large amounts of samples and they are sensitive to noise[65, 70].

2.3.2 Discrete head pose recognition

Over the years, a great variety of methods have been developed[5, 60] to recognize faces with specific (frontal) pose. An array of these systems, each trained to identify faces in a distinct pose allows multi-view face recognition and by extension, head pose estimation[68, 41, 67]. The final classification can subsequently be performed by using a voting mechanism or by comparing the sample to prototype faces learned for each pose.

The major disadvantage is the large number of detectors that need to be trained, each requiring sufficient training samples. If the system simultaneously functions as a head detector, a relatively large collection of non-face samples need to be added to the training set. As a result many detector arrays have so far been limited to estimating only a limited number of poses[27, 55], but this number is increasing with newer systems[30].

Model-based methods transform the sample image to conform to a set of prototypes. Cootes et.al. delveloped the Active Appearance Model[14] which uses a shape descriptor and PCA to create a statistical model, a prototype, of the shape of the head at each pose. The sample image is iteratively transformed to conform to the nearest prototype. After the transformation, the sample can be compared using standard template matching techniques to perform face identification, Active Appearance Models[16] manage to combine shape and texture variations and these have been used in multi-view face recognition systems[26, 15].

2.3.3 Flexible and geometric methods

With the exception of the sampling density methods from section 2.2.3, the majority of the methods described up to now have considered the problem from a purely statistical point of view, giving little consideration to the physical characteristics of the head and face.

In contrast, flexible methods, such as the Elastic Bunch Graph[29], search for a set number of major facial feature points (the nodes); eyes, nose, corner points of the mouth, and compare their relative positions (the graph) to the expected pose specific locations.

While the flexible methods compare the relative positions of feature points to prototypes to determine a discrete pose estimation, geometric methods use this information to determine a continuous pose estimate. There are multiple ways to calculate the head pose from different facial feature points but small differences between individual faces do not make this an easy task. One option is to determine the length of the nose away from the symmetry axis of the face[19]. Another method stems from the fact that the three lines from the outer corners of the eye, the inner corners of the eye, and the corners of the mouth run parallel[62].

The possible relative locations of the major facial feature points are of course constrained by the physical characteristics of the face. This information can be exploited to facilitate the search for the feature locations. Nonetheless, for flexible and geometric methods to operate reliably they require higher resolution images than which are needed for template methods. Additonally, these methods require the successful detection of all the required facial features which makes them sensitive to occlusions.

Part II

Head pose estimation with SDA

Chapter 3

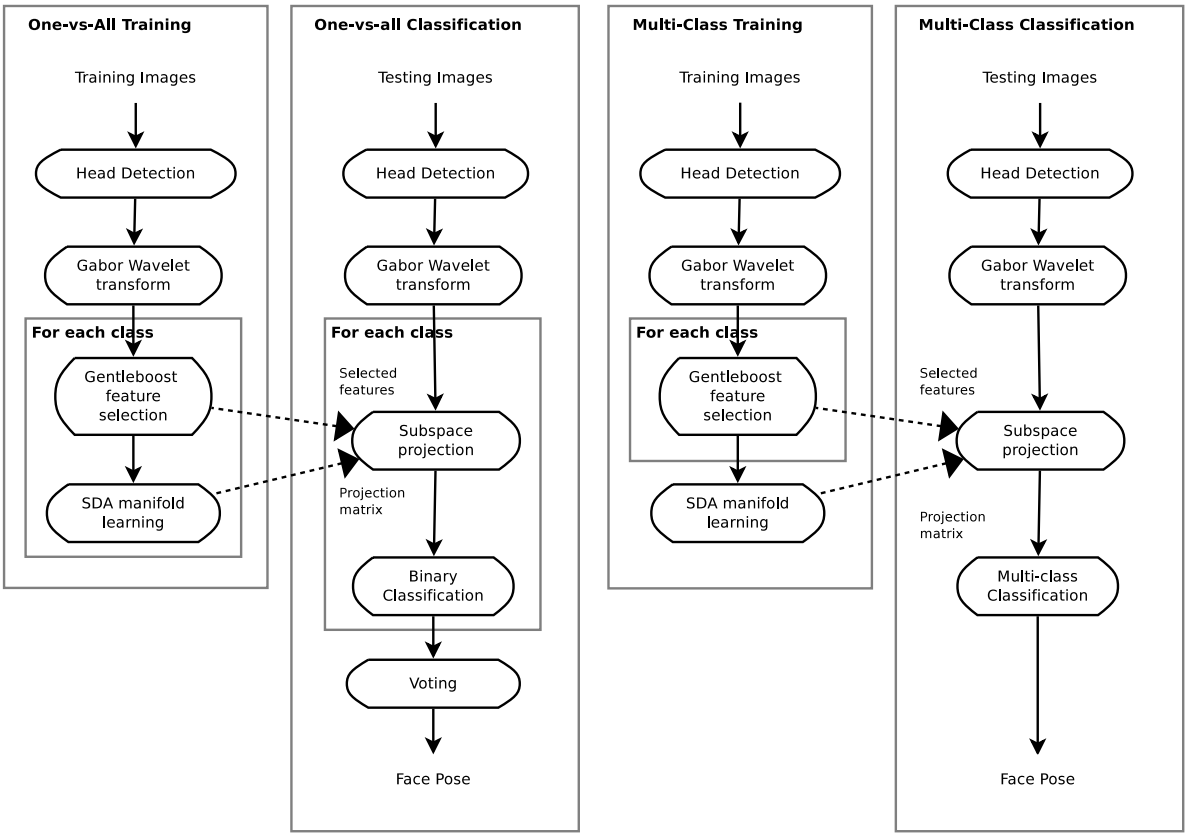
Approach

The proposed head pose estimation system takes a discrete approach to pose classification. The classifiable range of head poses, up to ±90° horizontally and ±60° vertically, is into distinct pose classes. As a first step in the training phase, in all training images, the head is located, cropped from the image and normalized in its dimensions. This is followed by a Gabor wavelet transform which results in a high-dimensional feature vector. A boosting procedure is applied once for each pose class. This results in multiple reduced feature sets specific to each pose class.

Figure 3.1: The division into pose classes, one sample from the Pointing'04 dataset is shown for each class

We investigate two approaches to pose classification, an array of one-versus-all classifiers and a multi-class classifier. For the array of classifiers, we apply the Subclass Discriminant Analysis algorithm separately for each pose class in order to learn a subspace to optimally distinguish that specific pose class from all other pose classes. The outputs of these binary classifiers are combined through voting which results in a final pose classification. The second approach combines the pose specific feature vectors in a single feature vector and applies multi-class SDA to learn a subspace in which we can perform a multi-class classification of head pose.

As can be seen in the system overview presented by figure 3.2, most of the steps are the same, or nearly the same, for both variations. Chapter 4 covers the detection of the location of the head within the sample image and chapter 5 discusses the transformation of the resulting head image into a feature vector and how we perform the pose specific feature selection. Chapter 6 discusses the application of discriminant analysis and the one-versus-all and multi-class classification of head pose.



(a) One-versus-all classification

(b) Multi-class classification

Figure 3.2: System overview, showing the two proposed classification and training systems

Chapter 4

Head detection

For face recognition, and by extension face orientation, most often only the frontal face area is used for classification[13]. Because we attempt to classify head orientation with larger horizontal and vertical angles the frontal face area would likely be obscured. Additionally, the head orientation estimation could benefit from the presence of edges of the head and the ear locations. Therefore we constructed a head detector to locate the head position. This head detector should detect heads under angles of up to $\pm 90°$ horizontally and $\pm 60°$ vertically.

This is a difficult task in general, but if we constrain the application to relatively predictable and clean images a relatively simple method can suffice. First the method locates areas of skin as determined by the color within the image. Secondly, working under the assumption that only one head is visible and this head provides the largest area of visible skin, we determine the position of the largest connected area of skin.

The first step is to perform color-based skin detection, which is most easily applied in the YCbCr color space[52]. Within the YCbCr color space we apply a threshold filter to the Cb and Cr values as described in [11, 45]. This threshold classifies pixels as either 'skin' or 'not skin':

$$M_{\text{skin}} = \begin{cases} 1, & \text{if } \text{Cb} \geq 77 \wedge \text{Cb} \leq 127 \wedge \text{Cr} \geq 133 \wedge \text{Cr} \leq 173; \\ 0, & \text{otherwise.} \end{cases} \tag{4.1}$$

with $0 \leq \text{Cb}, \text{Cr} \geq 255$. The output is a binary skin map M_{skin} (figure 4.1(b)).

(a) original image (b) skin mask (c) eroded skin mask

Figure 4.1: The outline of the head region as determined by the skin mask (b) and (c) super imposed on the original image (a)

To determine the position of the head with the use of the skin map we work under the assumption that there is only one head within the image and that this head provides the largest area of visible skin. For the Pointing'04 dataset discussed in chapter 7 a typical head width and height ranges from 160 to 240 pixels. We achieve good results if we erode the skin map by 7 pixels (figure 4.1(c)). The erosion reduces the connectivity between regions within the skin map while preserving the larger regions. The largest remaining connected region of skin is noted as the head location.

We crop the area formed by the axis aligned bounding box around this region, including the number of pixels we previously eroded, from the image. Some representative results on the dataset introduced in chapter 7 are shown in figure 4.2.

Figure 4.2: A set of representative examples of the head region detector

The cropped sample images are normalized in size to 128×128 pixels. By doing this, the same facial features, such as the eyes, nose and mouth correspond to fixed locations in the image regardless of the subject's original size in the picture (i.e. distance to the camera). Furthermore the image is turned to grayscale and normalized with regards to color. This results in sample images similar to those in figure 4.3.

Figure 4.3: The normalized images corresponding to the samples shown in figure 4.2

It is apparent from these images that the system has limitations which are important to note with regards to the feature selection. The image normalization may distort the aspect ratio, especially for faces with a very high pitch which often results in the inclusion of the neck into the image. However, the amount of distortion is similar for each pose class. The next chapter discusses the extraction of a feature vector from these images.

Chapter 5

Image representation

In this chapter we discuss the creation of the pose specific feature vectors. We apply a Gabor transform to the normalized head images from the previous chapter. This is followed by a feature selection stage using GentleBoost. GentleBoost is applied to each pose separately. This set of pose specific feature vectors is used for the classification of head pose in the next chapter.

5.1 Gabor feature extraction

Gabor features are largely insensitive to variation in lighting and contrast while simultaneously being robust against small shift and deformation in the image[49]. The 2D Gabor filter is a Gaussian kernel function modulated by a sinusoidal plane wave, which is commonly[63, 50, 59, 51] expressed as:

$$\begin{aligned} \varphi_{\Pi(f_0,\theta,\gamma,\mu)}(x,y) &= \frac{f_0^2}{\pi\gamma\mu} e^{-\left(\alpha^2 x'^2+\beta^2 y'^2\right)} e^{j2\pi f_0 x'}, \\ x' &= x\cos\theta + y\sin\theta, \\ y' &= -x\sin\theta + y\cos\theta, \end{aligned} \tag{5.1}$$

where $j=\sqrt{-1}$, f_0 is the central frequency of the sinusoidal plane wave, θ is the anti-clockwise rotation of the Gaussian and the plane wave, α is the sharpness of the Gaussian along the major axis parallel to the wave, and β is the sharpness of the Gaussian minor axis perpendicular to the wave. $\gamma=\frac{f_0}{\alpha}$ and $\mu=\frac{f_0}{\beta}$ are defined to keep the ratio between the frequency and sharpness constant. The Gabor filters are self-similar and are generated by dilation and rotation of a single mother wavelet. Each filter has the shape of a plane wave with frequency f_0, restricted by a Gaussian envelope with relative widths α and β.

Depending on the size and orientation of the specific features one is interested in, a filter with a corresponding frequency and orientation should be applied. It is unlikely that a single filter can detect all the required image features and it is common to apply a set of complementary Gabor filters, each tuned to features of a specific size and orientation:

$$\begin{aligned} \varphi_{u,v} = \varphi_{\Pi(f,\theta,\gamma,\mu)}(x,y), f_u = \frac{f_{\max}}{\sqrt{2^u}}, \theta_v = \frac{v}{V}\pi, \\ u=0,\ldots,U-1, v=0,\ldots,V-1, \end{aligned} \tag{5.2}$$

with $f_{\max}$ being the highest peak frequency, U and V being the number of desired scales and orientations respectively.

The value for the highest peak frequency $f_{\max}$ follows from Nyquist sampling theory and a good value for face related tasks is determined to be $f_{\max}=0.25$[51]. The ratio between the center frequency and the size of the Gaussian envelope is determined by γ and μ. This results in smaller filters to detect high frequency features and larger filters to detect low frequency filters. A commonly used value in face related tasks is $\alpha=\beta$ and $\gamma=\mu=\sqrt{2}$[63, 34, 51]. This results in a filter which is as long as it is wide.

There are some empirical guidelines for selecting scales and orientations[51] and common values are $U=5$ and $V=8$. This results in a filter bank, or family, of 40 different filters which is capable of

representing a wide range of facial features. Examples of these filters are shown for varying scales in figure 8.2 and for varying orientations in figure 8.3. These images only show the real part of the filters and are normalized to show negative values as black, positive values as white and zero as gray.

5.1.1 Gabor representation of sample images

The Gabor representation of an image I such as the normalized sample images shown in figure 4.3, can be obtained by convolving the image with each of the filters in the filter bank. The response of the image at location x, y to the filter with scale u and orientation v is given by:

$$G_{u,v}(x,y) = [I * \varphi_{u,v}](x,y). \tag{5.3}$$

The response $G_{u,v}$, has real and imaginary parts which are combined into the magnitude of the image response as follows:

$$G'_{u,v}(x,y) = \sqrt{\text{real}(G_{u,v}(x,y))^2 + \text{imag}(G_{u,v}(x,y))^2} \tag{5.4}$$

Once the convolution is done for all Gabor filters this results in a feature vector with a size of $U \times V$ the original number of image pixels. The response is downsampled using bi-cubic interpolation, to 16×16 pixels and normalized to zero mean and unit variance. The individual filter responses of sample t are concatenated into a single feature vector x^t with 10240 elements (5 scales $\times$ 8 orientations $\times$ 16^2 pixels). This is similar to the procedure in [34]. In the next stage of the system we use a boosting procedure to select only the most informative features from this vector in order to reduce the dimensionality and increase the classification performance.

5.2 GentleBoost feature selection

The feature vector x^t extracted by the Gabor filter has, even after downsampling, a very high dimensionality which makes classification more difficult. Simultaneously, there is a high correlation among the elements in the feature vector. Therefore we want to reduce the dimensionality of the feature vector and we want to select only the most informative features to support the SDA training. We do this by applying a boosting procedure for each pose class.

GentleBoost[18] is a variation on the original AdaBoost[17, 46] with improved performance for object detection problems[57] including the face recognition task[32]. Like AdaBoost, GentleBoost iteratively creates a committee of weak classifiers. The main difference to AdaBoost is how GentleBoost gently updates the weights for the training samples between iterations in the training procedure. Each individual weak classifier is merely a threshold function, or decision stump, operating on a single feature and has a performance possibly only slightly above mere guessing. A committee of these weak classifiers forms a strong classifier with good performance.

The unique features among those selected by each of these decision stumps form the list of selected features which we use in the SDA training. After the boosting procedure we should have a list of features for each pose most appropriate to the classification of that specific pose.

The outline of the GentleBoost algorithm is as follows. Start with weight $w^t = 1/N$ for each training sample t, with N samples in total. In each iteration m, fit the regression function $f_m(x)$ by weighted least-squares of the class labels y^t to the sample x^t using weight w^t. Update the combined classifier $F(x) = 0$ and update the weights according to the classification error:

$$F(x) \leftarrow F(x) + f_m(x) \tag{5.5}$$

$$w^t \leftarrow w^t e^{-y^t f_m(x^t)} \tag{5.6}$$

The weights should be re-normalized after each iteration. The regression function f_m is determined by minimizing the weighted error function:

$$\text{error} = \frac{\sum_t \left(w^t |y^t - f_m(x^t)|^2\right)}{\sum_t w^t} \tag{5.7}$$

By minimizing the weighted error we find the element x_i in the feature vector x with the smallest error and the appropriate values for a, b, and θ, which form f_m:

$$f_m(x^t) = a\left(x_i^t > \theta\right) + b, \tag{5.8}$$

A sample x^t can now be classified by the combined classifier as follows:

$$\text{sign}(F(x^t)) = \text{sign}(\sum_{m=1}^{M} f_m(x^t)) \tag{5.9}$$

Where M is the number of weak classifiers after M iterations.

Normally, when GentleBoost is applied as a classifier, we would continue to add new decision stumps until the strong classifier's performance stops improving. In that case it would be safe to continue boosting because the GentleBoost strong classifier does not overfit. In our case however, we do not want to select the maximum number of features and the point at which to stop boosting is determined empirically in chapter 8.

Chapter 6

Discriminant analysis

Once we have the image representation in the form of GentleBoost selected Gabor features we use discriminant analysis to find a subspace in which we can more easily perform the pose classification. Linear Discriminant Analysis (LDA) has been applied to face recognition before and is commonly known as a Fisherface[5]. LDA is a subspace projection technique and maps the high-dimensional image feature space to a low-dimensional subspace which simultaneously optimizes the class separability of the original data.

In this chapter we first review LDA followed by a recent variation on LDA named Subclass Discriminant Analysis (SDA). Once SDA has been introduced we discuss classification using a detector array which consists of a set of one-versus-all classifiers In the last section we discuss the application of SDA for multi-class head pose classification.

6.1 Linear Discriminant Analysis

Linear Discriminant Analysis and its derivatives are based on maximizing Fisher-Rao's criterion:

$$J = \max \frac{|W^T \mathbf{A} W|}{|W^T \mathbf{B} W|}. \tag{6.1}$$

Where W is the projection matrix we are looking for. The various variations usually differ in their definition of the matrices $\mathbf{A}$ and $\mathbf{B}$. For example, the well known Linear Discriminant Analysis uses the between and within-class scatter matrices $\mathbf{A} = S_B$ and $\mathbf{B} = S_W$, defined as,

$$S_B = \sum_{i=1}^{C} (\mu_i - \mu)(\mu_i - \mu)^T, \tag{6.2}$$

$$S_W = \frac{1}{N} \sum_{i=1}^{C} \sum_{j=1}^{n_i} (x_{ij} - \mu_i)(x_{ij} - \mu_i)^T, \tag{6.3}$$

where C is the number of classes, μ_i is the sample mean for class i, μ is the global mean, x_{ij} is the jth sample of class i and n_i the number of samples in class i.

Using this definition the objective function J attempts to maximize the Euclidean distance between the samples belonging to different classes while simultaneously minimizing the difference between samples of the same class. The objective function J is maximized when the column vectors of W are the eigenvectors of $S_W^{-1} S_B$. If the dimensionality of the feature vector is larger than the number of available samples, S_W becomes singular and its inverse does not exist. It is due to this "curse of dimensionality" that we applied the feature selection approach outlined in the previous chapter.

6.2 Subclass Discriminant Analysis

LDA assumes that the data is generated by a multivariate normal distribution which is not a valid assumption for either face identification or head pose estimation. Subclass Discriminant Analysis, developed by Zhu and Martinez[72], attempts to improve on LDA by modelling the data not as a single

Gaussian distribution but as a mixture of Gaussians. This mixture of Gaussians is represented by subclasses which are introduced by redefining the matrix $\mathbf{A}$ from the Fisher-Rao criterion shown previously in equation 6.1:

$$\mathbf{A} = \mathbf{\Sigma}_B = \sum_{i=1}^{C-1} \sum_{j=1}^{H_i} \sum_{k=i+1}^{C} \sum_{l=1}^{H_k} p_{ij} p_{kl} (\mu_{ij} - \mu_{kl})(\mu_{ij} - \mu_{kl})^T, \tag{6.4}$$

where H_i is the number of subclasses of C_i, μ_{ij} and p_{ij} are the mean and prior of the jth subclass of class i, respectively. The prior $p_{ij} = \frac{n_{ij}}{N}$ with n_{ij} as the number of samples in the jth subclass of class i.

This redefinition allows us to divide the training set into subclasses. The subspace resulting from the subsequent optimization of the Fisher-Rao criterion will maximize the class separability as with LDA, but also separate the subclasses. However, the subclass separation will not come at the cost of class separability.

As a first step in the SDA training procedure, the training set must be grouped into subclasses of their respective classes. It is difficult to know up front which division into subclasses is preferred. Zhu and Martinez[71, 72] use the nearest-neighbor method to order the training samples and subsequently divide them into subclasses of equal size which is not without problems[7]. For both the one-versus-all classifiers and for the multi-class classifier we experiment with a division based on k-means and a division based on refined pose classes. It should also be noted that assigning all samples to a single subclass makes SDA identical to LDA. This is convenient in order to perform the comparisons to the LDA method in chapters 9 and 10.

6.3 Classification with a detector array

The detector array consists of a single binary classifier for each of the 15 pose classes we want to be able to classify. The classification task for binary classifier i is to distinguish samples of pose class i (A_1) from samples of all other pose classes (A_2). Using the output of multiple of these one-versus-all classifiers we can then perform the classification of each pose class.

We examine two options to divide the samples into subclasses. First is the the application of k-means to divide the in-class and out-of-class samples into H_1 and H_2 subclasses, respectively. In chapter 9 it is shown how k-means clusters the samples partially by pose and partially by identity. The clustering by pose is of most use to us and because the dataset supports a more refined division than our 15 pose classes, we can use this to create pose subclasses within each pose class. For the in-class samples we divide the samples into as refined subclasses as the dataset allows. For the out-of-class samples we create one subclass for each of the pose classes directly surrounding the relevant pose class and an additional four subclasses subclasses for all poses with a tilt or pan smaller or larger than the surrounding pose classes. This is illustrated for pose class 1 (maximal pan and tilt) in table 6.1, with H_{1i} as the positive subclasses and H_{2j} the negative subclasses. Similarly, for pose class 8 this would result in 9 positive subclasses and 14 negative subclasses.

Tilt	Pan					
	90	75	60,45,30	15,0,-15	-30,-45,-60	-75,-90
60	$H_{1,4}$	$H_{1,3}$	$H_{2,2}$	$H_{2,6}$	$H_{2,6}$	$H_{2,6}$
30	$H_{1,2}$	$H_{1,1}$				
15,0,-15	$H_{2,1}$		$H_{2,3}$	$H_{2,4}$	$H_{2,4}$	$H_{2,4}$
-30,-60	$H_{2,5}$		$H_{2,4}$	$H_{2,4}$	$H_{2,4}$	$H_{2,4}$

Table 6.1: Subclass division for pose class 1.

Once the training samples are divided into subclasses and the subspace has been learned we can classify a given sample by projecting this sample into the learned subspace. Once the sample has been projected into the learned subspace we can perform the classification by locating the subclass nearest to the given sample. Mahalanobis distance and Normalized Euclidean distance have both proven to be reliable distance metrics for this purpose but they are not the top performers in all cases[34, 51]. In addition, because SDA models each subclass as a Gaussian distribution, each class distribution is a mixture of Gaussians. While the Euclidean or Mahalanobis distance metrics will calculate the subclass

closest to a given sample x, what we really want to know is the most likely class. Therefore we also test a third distance metric in which we use a mixture model to calculate the class probability. The three distance metrics are:

1. **Normalized Euclidean**, For each subclass l of class k calculate $P(A_{kl}|x^t) = |\frac{x^t - \mu_{kl}}{\Sigma_{kl}}|$ and select the class corresponding to the closest subclass.

2. **Mahalanobis**, For each subclass l of class k calculate $P(A_{kl}|x^t)$ as the Mahalanobis distance $P(A_{kl}|x^t) = (x^t - \mu_{kl})\Sigma_{kl}^{-1}(x^t - \mu_{kl})^T$, select the class corresponding to the closest subclass.

3. **Mixture Model**, Each subclass A_{kl} is a Gaussian distribution so we can calculate $P(A_k|x^t)$ using a mixture model:

$$\begin{aligned}
\chi &= \{x^t, y^t\}_{t=1}^N, && (6.5)\\
y_k^t &= 1, \text{if } x^t \in A_k \text{ and 0 otherwise}, && (6.6)\\
y_{kl}^t &= 1, \text{if } x^t \in A_{kl} \text{ and 0 otherwise}, && (6.7)\\
P(A_{kl}) &= \frac{\sum_t y_{kl}^t}{N}, && (6.8)\\
P(A_k) &= \frac{\sum_t y_k^t}{N}, && (6.9)\\
P(x^t|A_{kl}) &= \frac{1}{2\pi^{\frac{x^t}{2}}|\Sigma_{kl}|^{\frac{1}{2}}} e^{-\frac{1}{2}(x^t-\mu_{kl})\Sigma_{kl}^{-1}(x^t-\mu_{kl})^T}, && (6.10)\\
P(x^t|A_k) &= \sum_j P(x^t|A_{kl})P(A_{kl}), && (6.11)\\
P(A_i|x^t) &= \frac{P(x^t|A_i)P(A_i)}{\sum_k P(x^t|A_k)P(A_k)}. && (6.12)
\end{aligned}$$

Despite the dimensionality reduction by using the boosting feature selection procedure, the covariance matrices are at times singular, the results of this can be seen in chapter 9. Therefore, we also test two variations which use a common covariance matrix shared between all subclasses:

4. **Mahalanobis-Shared**, As **Mahalanobis** but with a covariance matrix shared between all subclasses.

5. **Mixture Model-Shared**, As **Mixture Model** but with a covariance matrix shared between all subclasses.

Once we have the output for each binary classifier, we select the final pose classification through a simple voting scheme. In the case of a tie between multiple binary classifiers we break the tie by assigning the pose classification to one of these classifiers at random.

6.4 Classification using multi-class SDA

The previous section discussed the application of SDA in an array of binary classifiers. Alternatively, SDA can be applied for multiple classes simultaneously to create a single subspace in which we can classify all of the pose classes. As with the one-versus-all classifiers, we experiment with a subclass division through k-means and through refined pose subclasses. For k-means we divide each pose class into an equal amount of subclasses. For the division according to pose we divide each pose class as we did for the in-class samples of the one-versus-all classifiers.

The interesting part of SDA applied for all pose classes simultaneously is the position of the samples within the learned subspace. Within the learned subspace, similar samples will be near each other. If the training is successful and the feature vector expresses the correct information, 'similar' means similar in pose. As a result, if we take a number of samples representing a subject panning his head from left to right, these samples plot a curve in the subspace. This result would be a single line from figure 6.1. If the subject also tilts his head, the samples will represent a two-dimensional manifold.

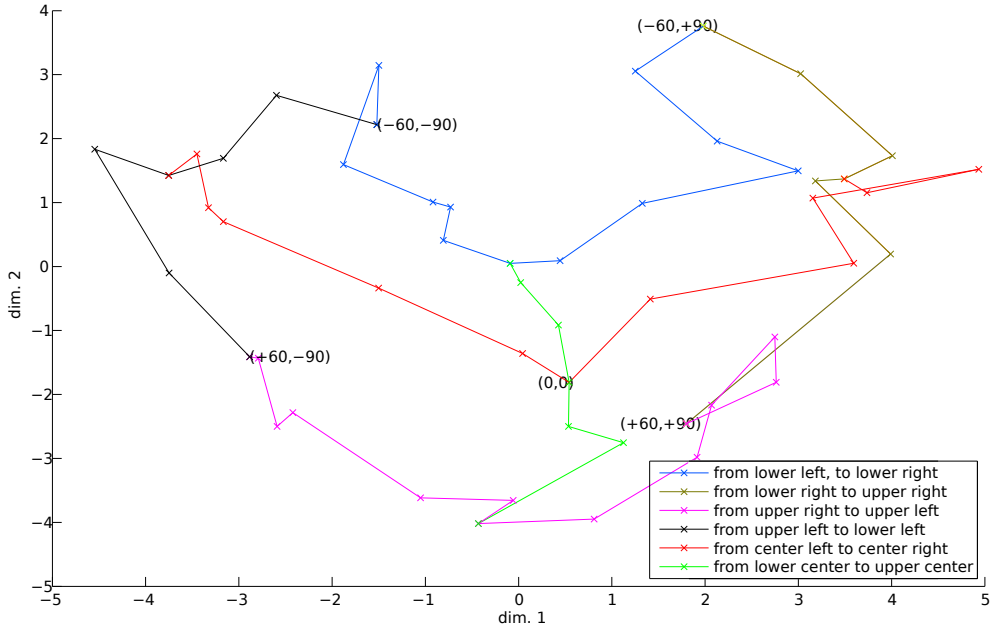


Figure 6.1: The movement (pan and tilt) of subject 2 from the Pointing'04 dataset through the first two dimensions of the subspace learned from this dataset.

One benefit of the existence of this manifold is that if we have misclassified a sample, the misclassified pose is likely close to the actual pose. This decreases the error in degrees of the classification error. But a greater benefit might be the potential for continuous pose estimation. This can be done in either of two ways. We can either estimate the continuous pose based on the likelihood of the discrete classification or we can find a two-dimensional (pan and tilt) mathematical description of the manifold. If we have such a description of the manifold, the coordinates of a sample on the manifold corresponds to a continuous estimate of the subject's pan and tilt angles.

Part III

Experimental results

Chapter 7

Pointing '04 dataset and SDA subclass division

For the training and testing of the classifiers the Pointing'04[21] was used. This dataset contains pictures of 15 different persons, photographed twice in each pose. Each image has 384×288 pixels. The available horizontal angles range from -90° to +90° and the tilt ranges from -90° to +90° in steps of 15°. For the negative angles the subject faces downwards and to its left.

The samples are divided into 15 classes according to the orientation of the head, as shown in table 7.1. Sample with a ±90° tilt are only available with 0° pan angle and are not part of these classes. As can be seen in figure 3.1, some subjects wear glasses but no other face coverings, the expressions are fairly consistent as is the lighting, with the exception of one image series. Each discrete pose has 15subjects × 2series = 30 images, as a result the four corner classes (1, 3, 13 and 15) each contain 120 samples, classes 5, 8 and 11 (frontal and left and right from the frontal pose) contain 270 samples. The remaining classes contain 180 samples each. With the exception of approximately ≈ 10 samples, all with a pitch of ±60 degrees, all samples were cropped quite well by the head detector. The cropped images of these ≈ 10 samples do show the major facial features and were not discarded.

Tilt	Pan				
	90,75	60,45,30	15,0,-15	-30,-45,-60	-75,-90
60,30	1 (120)	4 (180)	7 (180)	10 (180)	13 (120)
15,0,-15	2 (180)	5 (270)	8 (270)	11 (270)	14 (180)
-30,-60	3 (120)	6 (180)	9 (180)	12 (180)	15 (120)

Table 7.1: Class labels corresponding with their poses, the number of samples per class are noted within brackets

Tilt	Pan				
	90,75	60,45,30	15,0,-15	-30,-45,-60	-75,-90
60,30	4 (6)	6 (9)	6 (10)	6 (9)	4 (6)
15,0,-15	6 (8)	9 (11)	9 (14)	9 (11)	6 (8)
-30,-60	4 (6)	6 (9)	6 (10)	6 (9)	4 (6)

Table 7.2: The number of in-class (and out-of-class) subclasses for the one-versus-all classifiers for each pose class when refined according to pose annotations.

As discussed in section6.2, we divide the samples in subclasses by either k-means or by their pose annotations. Table 7.2 shows the number of subclasses for each pose class while using the pose annotation of the Pointing'04 dataset. The multi-class classifier uses the same subclass division as used for the in-class samples of each of the one-versus-all classifiers.

Chapter 8

GentleBoost feature selection

In this chapter we explore the characteristics of the GentleBoost feature selection. We first look at the performance characteristics of the GentleBoost strong classifier as the number of decision stumps increases. Next, we look at the features selected by those decision stumps. Finally we take the binary SDA classifier and train it on different combinations of selected features in order to determine a good feature set to use in combination with the SDA classifiers of the next chapters.

8.1 GentleBoost strong classifier

For each of the fifteen pose classes we created a pose specific dataset consisting of all in-class samples and two times the number of random out-of-class samples. The GentleBoost classifiers were trained on this pose specific dataset using a k-fold procedure with $k = 3$.The ROC curves for four representative pose classes are shown in figure 8.1. They show the expected increase in performance as the number of stumps increases. As can be seen more clearly in figure 8.1(e), the performance increase levels off well before the 500th decision stump and there is very little gain in using the next 1500 stumps. We also found no decrease in performance due to overlearning, as is consistent with the literature. The performance characteristics are similar for all pose classes.

The locations of the first 100 selected features of each of the three folds are shown by scale on the heatmaps in figure 8.2 and by orientation in figure 8.3. Although features which cover the background of the image instead of the face are selected, the majority of the selected features correspond to pose specific locations of facial features. The Gabor filters with a smaller scale mainly follow the contours of the head while the larger filters cover the major facial features such as the eyes, nose and mouth.

8.2 Selecting features for discriminant analysis

The previous section discussed the GentleBoost strong classifier performance with regards to the number of decision stumps. This section shows how the uniquely selected features corresponding to these decision stumps correlate with the performance of a SDA classifier trained on these features.

Three separate feature sets are created to investigate three variations on gathering the features from the boosting process:

G_1 The first n features selected by the GentleBoost classifier.

G_2 The first n features selected more than once by the same classifier.

G_3 The first n features selected by more than one application of the GentleBoost classifier.

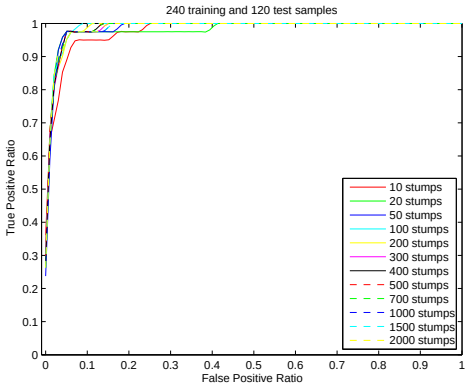
The training and testing method is again a k-fold with $k = 3$. The value of n is equal to the number of in-class samples within the training set. For G_3 we divide the training set into four folds and train a separate GentleBoost classifier on each fold.

For this experiment we don't divide the data into subclasses, which makes SDA identical to LDA. The classification is performed by the Mixture Model metric in the first dimension of the SDA manifold,

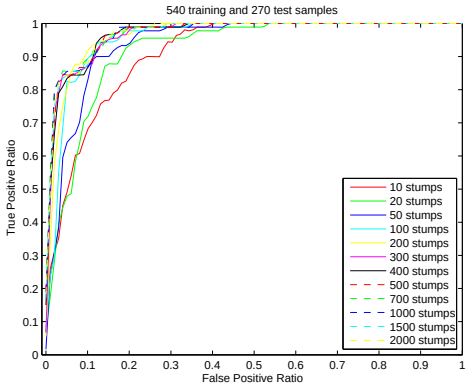
as is common practice with LDA. The ROC curves corresponding to the tree feature sets are shown in figure 8.4. The ROC curves show no clear performance difference between the three feature sets. In contrast, either a random selection of n features or a sparse subsampling of the whole feature set does show a significant drop in performance compared to either of the three feature sets above. It appears that feature selection through a combination of multiple short boosting runs (G_3) is not an improvement over selection through a single, longer, boosting run (G_1). The property of the GentleBoost classifier as a feature selector to select a single feature multiple times does not imply an increased benefit of this feature with regards to feature selection. This is consistent with existing empirical evidence that boosting procedures do not overfit and do not create bad decision stumps.

The boosting process allows for a single feature to be part of multiple decision stumps. This is a necessary part of the boosting classifier but of no use for feature selection. Figure 8.5(a) shows the ratio of unique features versus the number of decision stumps and 8.5(b) shows the performance of the resulting SDA classifiers. Selecting a larger number of features thus becomes an increasingly resource intensive task for diminishing returns.

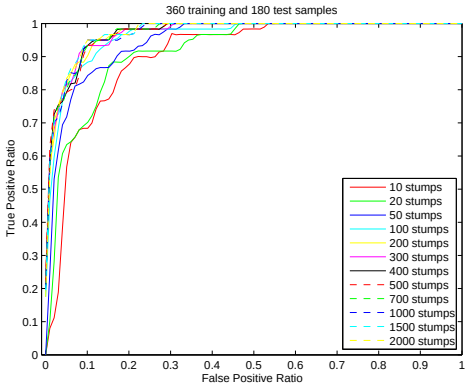
For multi-class SDA we require a feature set which is able to represent all 15 pose classes. This set is created by concatenating the features from all 15 pose specific feature sets, in sequence, to a combined feature set. The first 100 features from this combined feature set are shown in figures 8.6 and 8.7. Even though the heatmaps show only a small number of features specific to each pose class, it can be seen that the feature set is largely symmetrical, covers the whole range of poses and has very little coverage of noise areas such as background and hair.



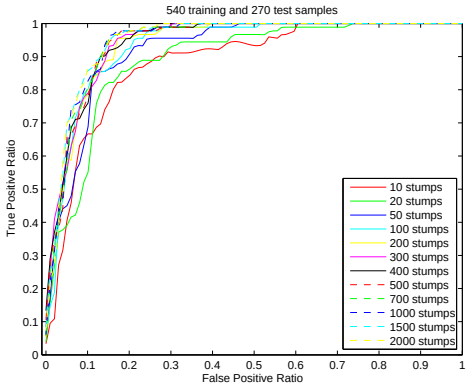
(a) Pose class 1



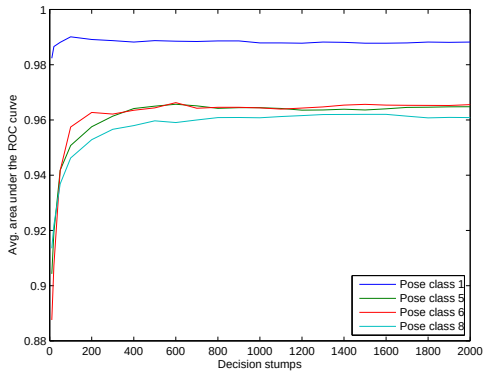
(b) Pose class 5



(c) Pose class 6



(d) Pose class 8



(e) The average area under the ROC curves over 3 folds

Figure 8.1: ROC curves show how the strong classifier's performance increases with the number of decision stumps

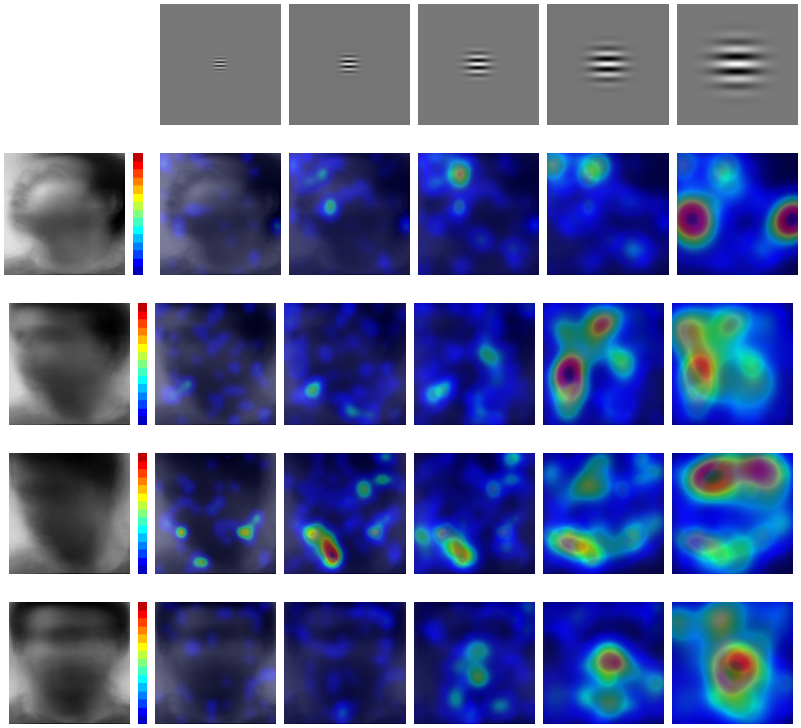


Figure 8.2: Selected features shown separately for (per row) classes 1, 5, 6 and 8 and (per column) each of the five scales

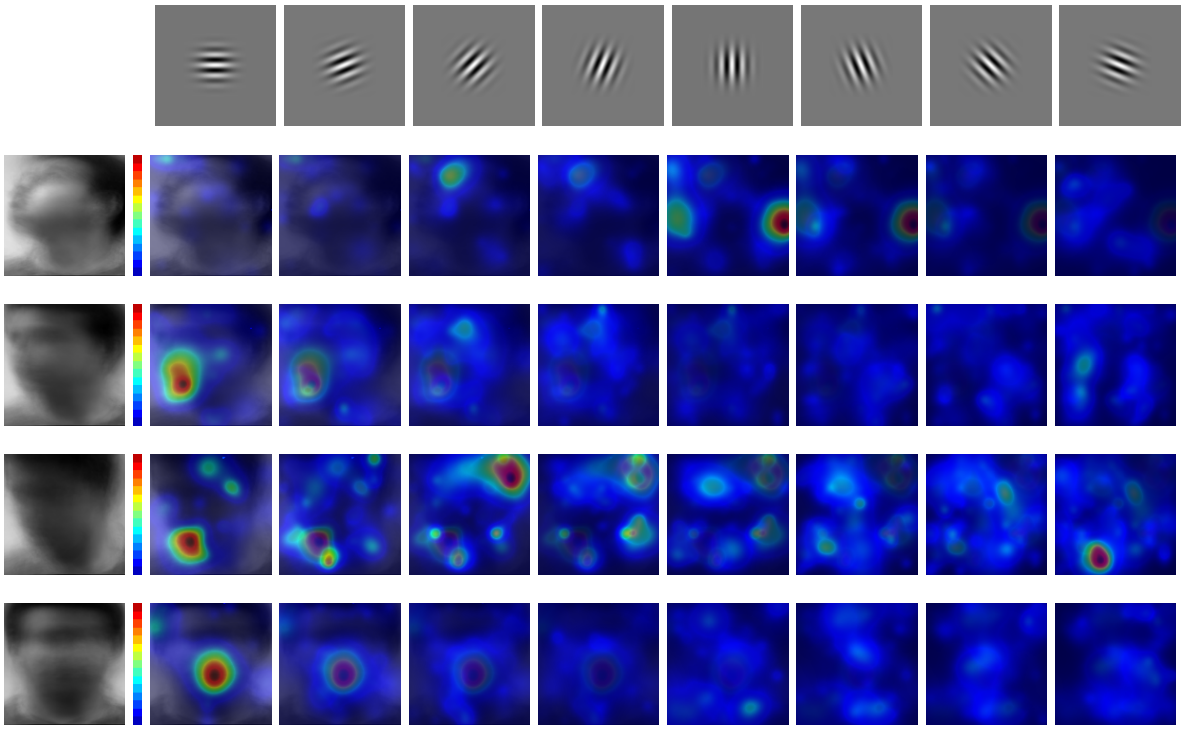
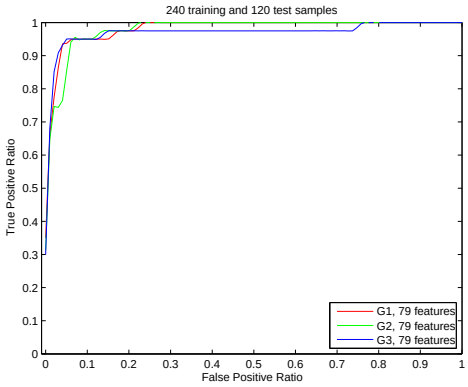
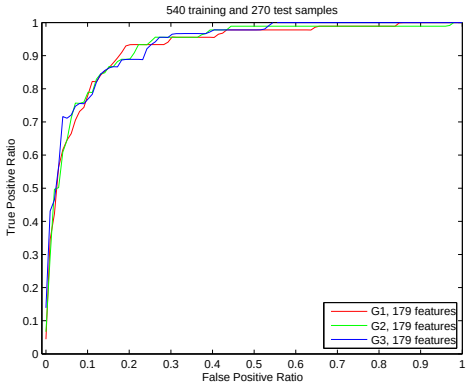


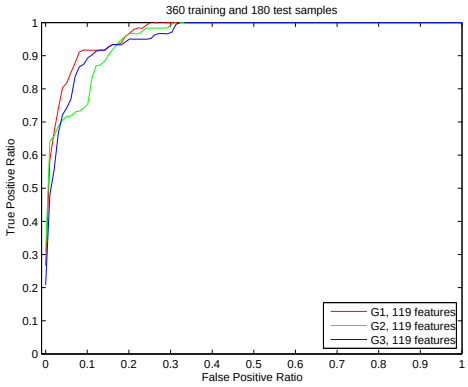
Figure 8.3: Selected features shown separately for (per row) classes 1, 5, 6 and 8 and (per column) each of the eight orientations



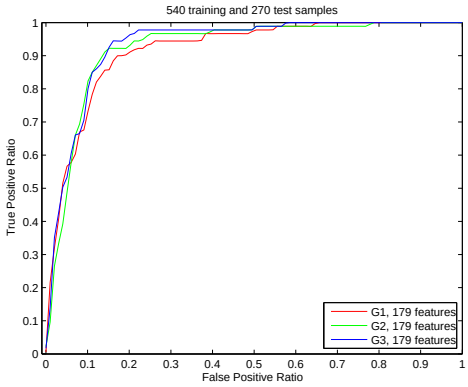
(a) Pose class 1



(b) Pose class 5

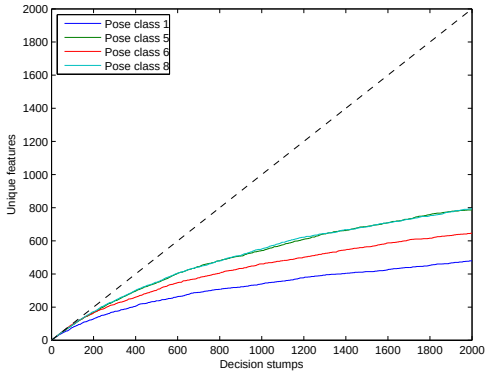


(c) Pose class 6

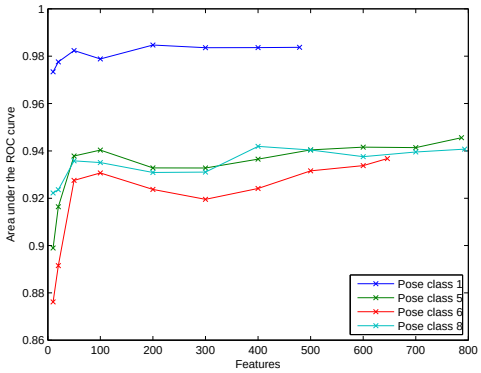


(d) Pose class 8

Figure 8.4: ROC curves showing the variation in SDA classification performance when varying the origin of the used features



(a) The number of unique features versus the number of decision stumps in the boosting classifier.



(b) Average area under the ROC curve over 3 folds versus the number of features used.

Figure 8.5: Diminishing returns for increasing numbers of decision stumps.

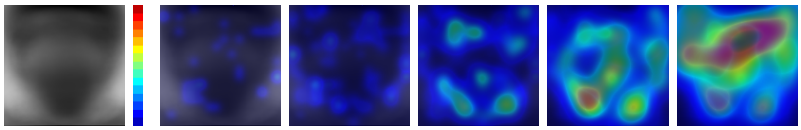


Figure 8.6: Selected features shown separately for each of the five scales

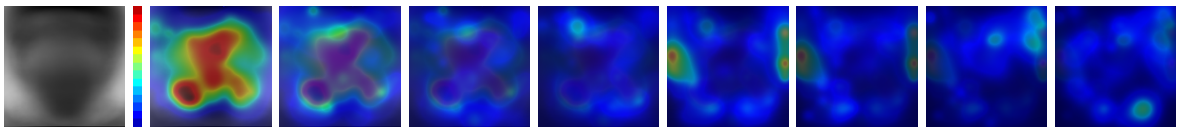


Figure 8.7: Selected features shown separately for each of the eight orientations

Chapter 9

Subclass Discriminant Analysis

In the previous chapter we discussed the application of GentleBoost and the influence of the number of features on performance. In this chapter we discuss the dispersal of the training samples within the learned SDA subspace and how this is influenced by the division of the training samples into subclasses. We also look at the different classification metrics and the optimal number of dimensions to apply them in. We start with the one-versus-all classifiers and end the chapter with the multi-class classifier.

9.1 One-versus-all classification

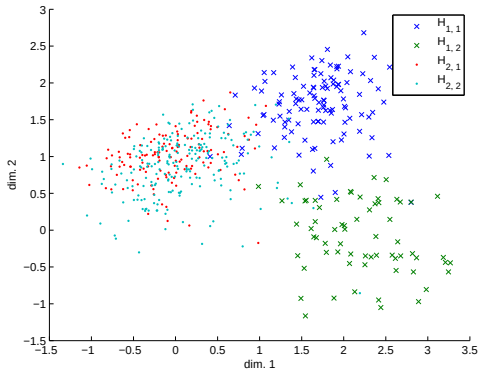
The one-versus-all classification in this section uses the features learned from the GentleBoost feature selection of section 8.2. Figure 9.1 shows the position of the training samples for pose class 8 in a subspace learned by training SDA on 100 features. The total number of dimensions in a subspace created by SDA is equal to the total number of subclasses minus one. For this experiment the positive samples and the negative samples were both divided into two subclasses by the application of k-means clustering. This experiment used k-means to group the positive samples into subclasses $H_{1,1}$ and $H_{1,2}$ and the negative samples into $H_{2,1}$ and $H_{2,2}$. Any set of two dimensions of the subspace shows a clear separation of three out of four subclasses. These subclasses would not be discernible if the subspace were limited to a single dimension, as would be the case with LDA.

The clustering of the subclasses shows some correlation to the subjects pose. In this example $H_{1,1}$ has relatively fewer samples with a pan of 0° while $H_{1,2}$ has only a small number of samples with a pan of 15°. The negative samples are also grouped around the horizontal orientation. Additionally, there are a number of subjects which occur frequently in one subclass and hardly in the other. When we increase the number of subclasses to four for both the positive and negative samples, the grouping around the orientation is more distinct, this can be seen most clearly for pose class 1 in figure 9.2(a). The vast majority of samples in $H_{1,1}$ and $H_{1,4}$ have a 30° tilt while the samples in $H_{1,2}$ and $H_{1,3}$ have a 60° tilt. The differences between $H_{1,1}$ and $H_{1,4}$ and $H_{1,2}$ and $H_{1,3}$ appear to be due to identity. If we use the pose annotation on the training data to group the samples into more refined pose subclasses we can see a similar layout of the subclasses in figure 9.2. This time there is a strong division due to the tilt angle of the samples $H_{1,1}$ and $H_{1,2}$ (30° tilt) versus $H_{1,3}$ and $H_{1,4}$ (60° tilt). The separation due to the pan angle is less pronounced ($H_{1,1}$ versus $H_{1,2}$ and $H_{1,3}$ versus $H_{1,4}$) but still visible. Additionally, the negative samples that have both a different pan and a different tilt angle ($H_{2,4}$) form a distinct cluster.

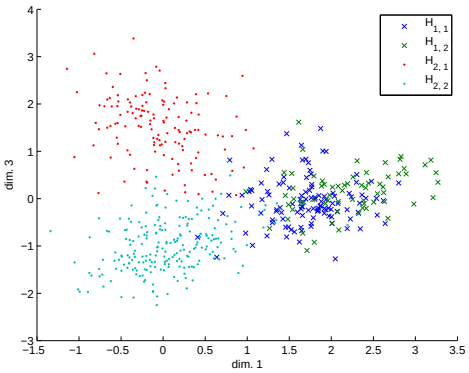
Dim.	Subclasses			
	$k=1$	$k=2$	$k=4$	by pose
1	93.5	93.7	93.8	93.8
2	-	93.2	93.6	94.4
3	-	93.9	93.3	94.2

Table 9.1: Average area under the ROC curve for pose class 8 measured over 3 folds.

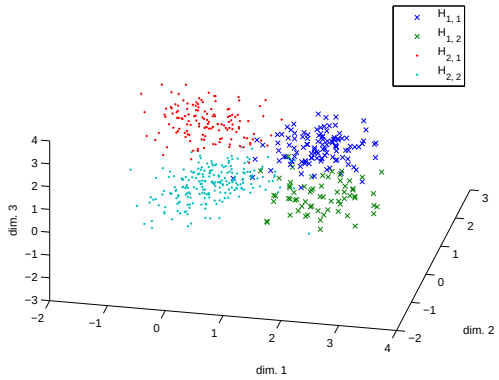
These figures show how the subclasses and multiple dimensions of the SDA subspace help to cluster the samples. However, the effect on the classification performance is less pronounced. As previously,



(a) The first two dimensions.

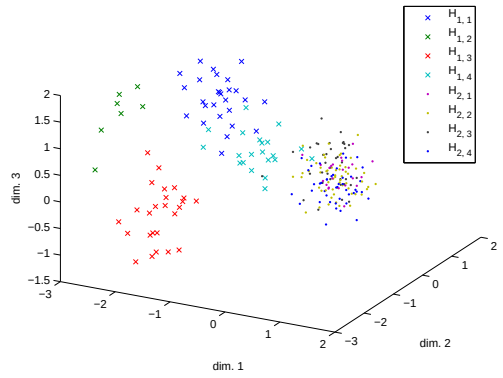


(b) The first and third dimension.

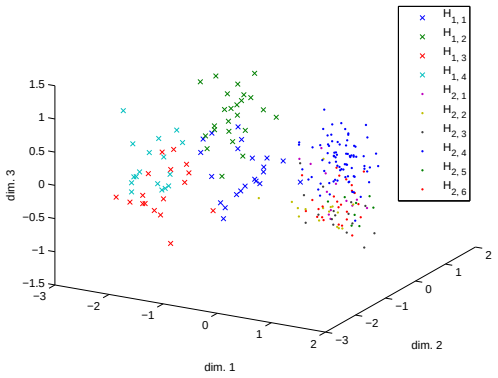


(c) All three dimensions.

Figure 9.1: The training samples in the learned subspace for pose class 8 using two subclasses for both the positive (crosses) and the negative samples (dots).



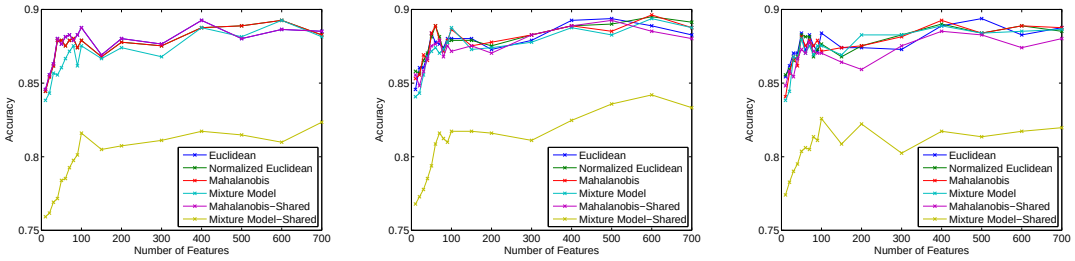
(a) Subclass division through k-means.



(b) Subclass division based on pose annotation.

Figure 9.2: The training samples in the learned subspace for pose class 1 using four subclasses for the positive samples.

we divide the training samples by either k-means or by their pose annotation. Table 9.1 shows the classification results, using the Mixture Model, for four variations of subclass division. Despite the differences in the sample dispersal seen in the previous figures, the actual differences in classification are very minor. The table also suggests that performing the classification in more than one dimension brings little benefit to the classification performance. This can also be seen in figure 9.3 which shows similar performance results for classification in up to the first three dimensions. The figures also show



(a) One dimension. (b) Two dimensions. (c) Three dimensions.

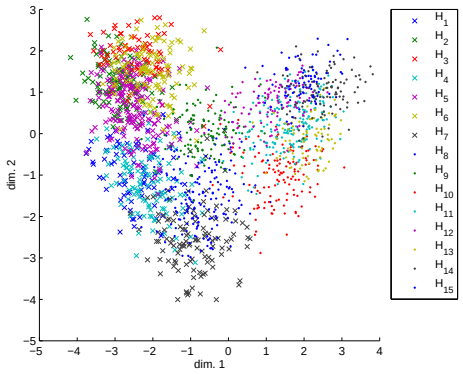
Figure 9.3: Average classification accuracy for pose class 8 over 3 folds, as a function of the number of features used for learning the SDA subspace.

some clear results with regards to the classification metrics. The shared covariance matrix is detrimental to the performance of the Mixture Model while this does not have the same effect on the Mahalanobis distance. This suggests that an alternative method of calculating the shared covariance matrix might be preferable. The other metrics all have very similar performance profiles.

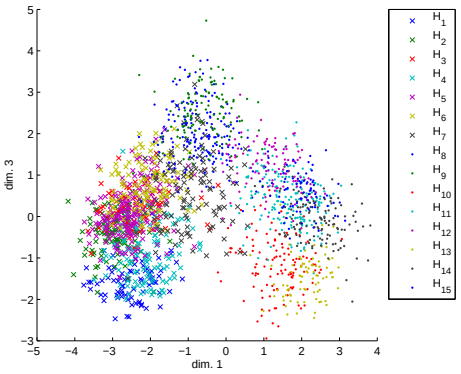
9.2 Multi-class SDA

In the previous section we have seen how SDA groups the subclasses by pose in the subspace learned for the one-versus-all classifiers. In the multi-class situation this property is even more pronounced. Figure 9.4 shows the dispersal of the training samples in a SDA subspace trained on 300 features without subclasses.

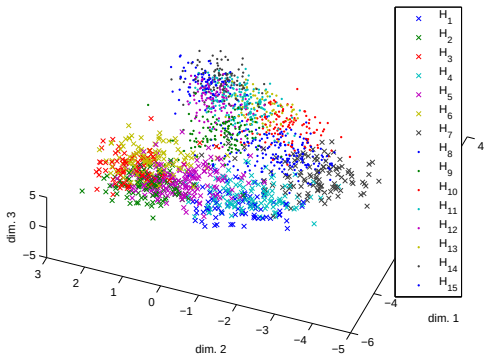
Even without subclasses, the samples are largely ordered in a "v" or wing shape according to the subject's pose. As figure 9.5 shows, the addition of subclasses does have some slight performance benefits. However, as we add more subclasses, and perform the classification in more dimensions the condition of the covariance matrix degrades, this is shown in figure 9.6. The effects of this degradation on the Mixture Model and Mahalanobis metrics is easily visible in figure 9.5. The metrics that do not require a covariance matrix or have a shared covariance matrix do not share these difficulties and their performance remains stable.



(a) The first two dimensions.

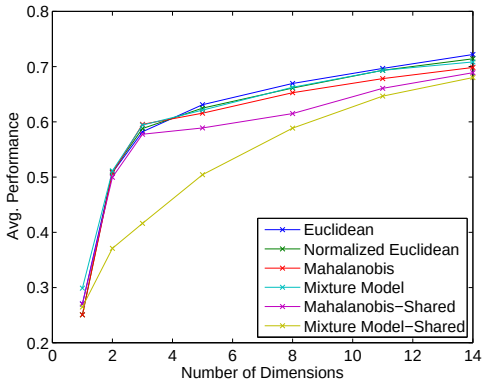


(b) The first and third dimension.

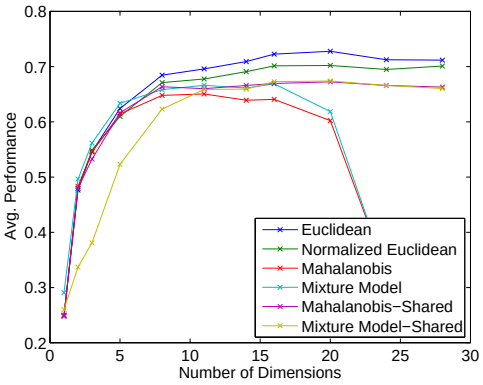


(c) The first three dimensions.

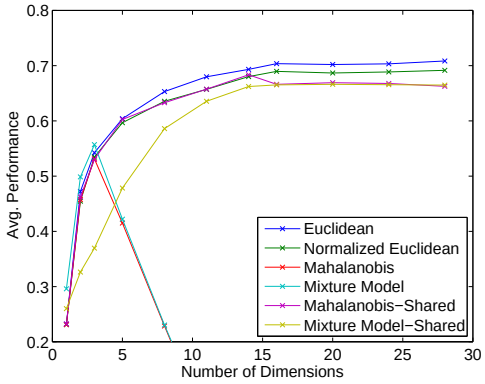
Figure 9.4: The training samples in the SDA subspace learned without subclasses.



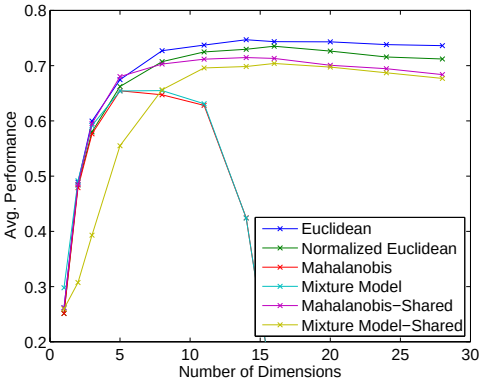
(a) No subclasses.



(b) Two subclasses per pose class.



(c) Four subclasses per pose class.



(d) Subclasses divided by pose.

Figure 9.5: The average performance of multi-class SDA trained on 600 features.

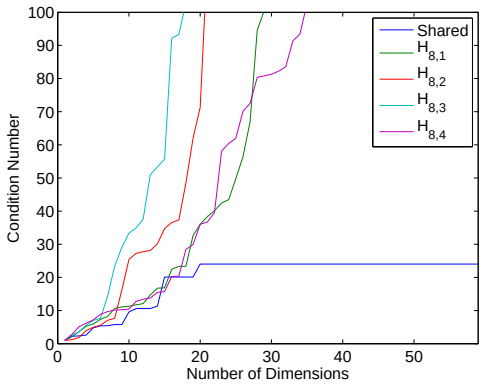


Figure 9.6: The condition of the covariance matrix degrades as the number of dimensions increases.

Chapter 10

Comparison of classifiers

This chapter compares the one-versus-all approach with the multi-class approach. For both approaches we apply the best settings for GentleBoost and SDA as determined in the previous chapters. We also apply both approaches with only a single subclass for each pose class. This creates a LDA based variation to both approaches. For additional comparisons we also combine the GentleBoost strong classifiers through the same basic voting mechanism as the one-versus-all SDA approach. First we discuss the final settings and classification results for each of the variations trained with the same three-fold cross validation procedure as used in the previous chapters. In the final section we show the comparison between the variations.

10.1 Combined GentleBoost classifier

We combine the GentleBoost strong classifiers we used to select the features for the binary SDA classifiers through a voting scheme. If more than one GentleBoost strong classifier return a positive result, we assign the classification to one of their respective pose classes at random. The results are shown in table 10.1. The strong classifiers in this experiment have approximately 120 decision stumps on average. The mean absolute horizontal error is 4.6° and the vertical error is 13.1°.

Tilt	Pan				
	90,75	60,45,30	15,0,-15	-30,-45,-60	-75,-90
60,30	81.7	68.3	57.8	64.4	89.2
15,0,-15	65.6	57.8	63.3	63.7	73.9
-30,-60	71.7	55.6	70.6	53.3	82.5

Table 10.1: Classification results for the combined GentleBoost classifier, the overall accuracy is 66.2%

10.2 Combined SDA classifier

For the multi-class classifier we combine the binary classifiers with a standard voting scheme. The individual binary classifiers use 100 features each, selected by the GentleBoost classifiers from the previous section. The classification results are shown for the Mixture Model applied in the first dimension and the first and second dimensions of the subspace in table 10.2. This again shows that the additional dimensions do not benefit the classification. The mean absolute horizontal error is 5.8° and the vertical error is 13.7° when the classification is performed in the first dimension. Extending the classification to the second dimension results in errors of 6.3° and 16.0° for the horizontal and vertical directions respectively.

(a) [The average classification in the first dimension, the overall accuracy is 64.7%.

Tilt	Pan				
	90,75	60,45,30	15,0,-15	-30,-45,-60	-75,-90
60,30	72.5	58.9	64.4	65.6	84.2
15,0,-15	64.4	63.7	54.1	65.6	70.6
-30,-60	70.8	56.1	68.3	54.4	77.5

(b) The average classification results in two dimensions, the overall accuracy is 62.8%.

Tilt	Pan				
	90,75	60,45,30	15,0,-15	-30,-45,-60	-75,-90
60,30	74.2	58.3	63.3	63.9	83.3
15,0,-15	60.0	60.4	53.7	61.5	67.2
-30,-60	65.8	53.3	69.4	52.2	78.3

Table 10.2: Classification results for the combined SDA classifier using the Mixture Model.

10.3 Combined LDA classifier

In this section we show the classification results with multiple LDA based classifiers. We use the same setup as in section 10.2, except that we do not group the training samples into subclasses. In this case SDA is identical to LDA[72] and as can be seen in table 10.3, the results are similar as well. The learned subspace is limited to a single dimension in which we apply the Mixture Model for classification. The mean absolute horizontal error is 7.3° and the vertical error is 14.7°.

Tilt	Pan				
	90,75	60,45,30	15,0,-15	-30,-45,-60	-75,-90
60,30	70.0	59.4	61.1	62.2	88.3
15,0,-15	67.8	57.0	55.2	56.3	68.3
-30,-60	68.3	50.6	63.3	50.6	80.8

Table 10.3: Classification results for the combined LDA classifier, the overall accuracy is 62.1%.

10.4 Multi-class SDA classification

For the multi-class application of SDA we combine the feature sets selected by the GentleBoost classifiers from section 10.1. From this combined feature set we used the first 600 features for learning the subspace. To allow a direct comparison with the multivariate LDA classifier in section 10.5 we limit the classification to 14 dimensions. The classification results with the euclidean distance metric are shown in 10.4. The mean absolute horizontal error is 3.3° and the vertical error is 10.1°.

Tilt	Pan				
	90,75	60,45,30	15,0,-15	-30,-45,-60	-75,-90
60,30	73.3	77.8	70.0	80.0	82.5
15,0,-15	78.3	74.4	79.6	80.0	75.6
-30,-60	54.2	68.9	77.2	75.6	57.5

Table 10.4: Classification results for multi-class SDA classification with an average of 6 subclasses, using the Mixture Model in 14 dimensions, the overall accuracy is 74.7%.

10.5 Multi-class LDA classification

As with the previous section, we show the results for the LDA variation on the multi-class SDA classifier from section 10.5. The classification is performed with the euclidean distance metric and shown for 14 dimensions in table 10.5. The mean absolute horizontal error is 3.2° and the vertical error is 10.9°.

Tilt	Pan				
	90,75	60,45,30	15,0,-15	-30,-45,-60	-75,-90
60,30	77.5	68.3	68.9	70.0	82.5
15,0,-15	76.1	69.6	71.9	74.4	75.0
-30,-60	66.7	66.7	78.9	67.8	72.5

Table 10.5: Classification results for multi-class LDA classification, over 3 folds, using the Mixture Model in 14 dimensions, the overall accuracy is 72.2%.

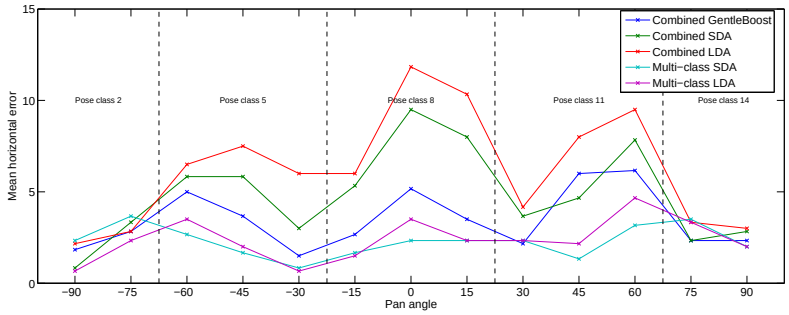
10.6 Comparison

The classification results for the five different classifiers are reasonably similar. There is also considerable overlap in the classification output of each classifier. This is illustrated in table 10.6. The table shows the combined SDA and LDA classifiers having a particularly strong agreement between their respective classification outputs. Less strong, but still evident is the agreement between the multi-class SDA and LDA classifiers. The GentleBoost classifier has no particularly strong agreement with the other variations. This pairing can also be seen in figures 10.1 and 10.2. Figure 10.1 shows a comparison of the

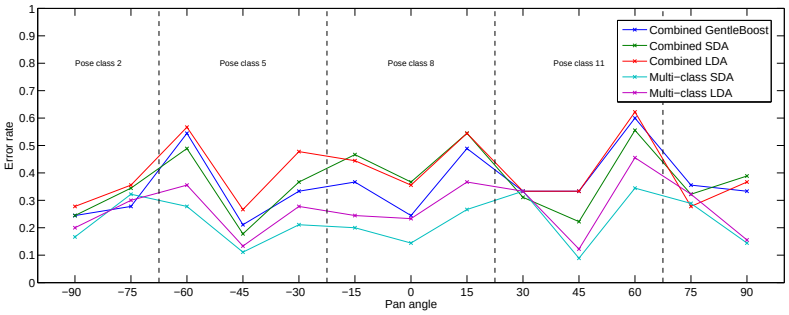
	Actual	Comb. Boost	Comb. SDA	Comb. LDA	Mult. SDA	Mult. LDA
Actual	-	66.2	64.7	62.1	74.7	72.2
Comb. Boost	66.2	-	71.9	70.7	67.4	71.1
Comb. SDA	64.7	71.9	-	91.2	67.2	69.6
Comb. LDA	62.1	70.7	91.2	-	65.0	67.7
Mult. SDA	74.7	67.4	67.2	65.0	-	84.0
Mult. LDA	72.2	71.1	69.6	67.7	84.0	-

Table 10.6: Agreement in percentages between the classifier variations and the actual values.

mean error in pan estimation for each classifier as the subjects tilt is approximately 0° while the subject pans its head ±90°, which corresponds to pose classes 2, 5, 8, 11 and 14. Similarly, the mean error in tilt estimation for pose classes 7, 8 and 9 is shown in figure 10.2. These figures show an increase in the error rate when the subject's movement nears a class boundary, as one would expect. Despite the lower error rate near the class centres, figure 10.1 shows an increase in the horizontal error in degrees near the class centres.

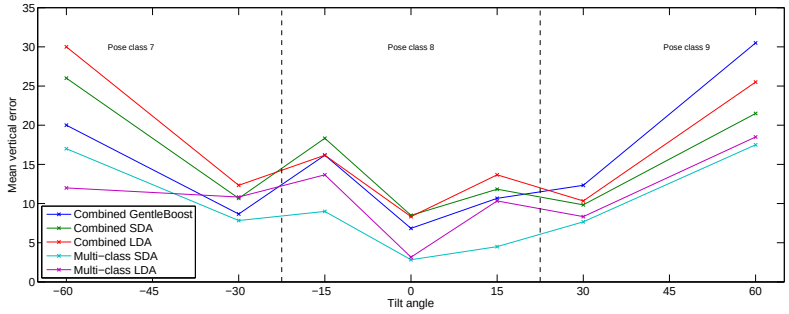


(a) Horizontal error in degrees versus pan.

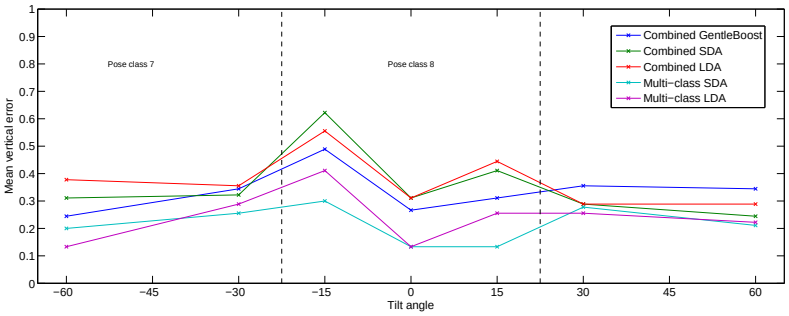


(b) Error rate versus pan.

Figure 10.1: Horizontal rotation of the subject's head.



(a) Horizontal error in degrees versus tilt.



(b) Error rate versus tilt.

Figure 10.2: Vertical rotation of the subject's head.

Part IV

Conclusion

Chapter 11

Final discussion

For this thesis we have created and analysed an automatic feature selection method and two classification approaches. Combined with a head detector these form a head pose detector whose performance has been compared to similar pose detection methods.

The boosting procedure succesfully selects informative features that enable succesful pose classification. The multi-class approach is shown to be preferable over the one-versus-all approach. Additionally, the SDA classifiers are shown to have performance characteristics comparable to those of LDA for both the one-versus-all and the multi-class approach.

11.1 Feature selection

The automatic feature selection through GentleBoost results in a feature vector covering the essential facial areas and ignoring the background noise left over from the head crops. This coverage would be similar as one would achieve with a manually generated density sampling mask. The automatic generation of sampling masks is very convenient for any object recognition tasks and especially when we need multiple masks such as for an array of pose classifiers.

The experimental results further indicate that a single application of the boosting procedure is sufficient to achieve optimal results from the GentleBoost feature selection. It is possible to select fewer features for pose classes that are easier to classify, such as the pose classes on the edge of the allowable range of motion. However, a number of features larger than necessary does not immediately result in a decrease in performance. Furthermore, because the features are selected in order of importance we can choose how many of the selected features to use for training the SDA classifier and be confident that we supply the SDA procedure with sufficient information to perform the classification. As a result, we successfully selected a limited number of features from each of the pose specific feature sets to combine into a single feature set for multi-class SDA classification.

11.2 Subclass Discriminant Analysis

The experiments show that a division of classes into subclasses can result in a clear spread of these subclasses in the learned subspace of the one-versus-all classifiers. This occurs when the division is performed by either k-means or by pose annotations. However, different subclass divisions do not result in an increase in performance. There is also no large difference in the classification performance between the SDA variations and the LDA variant. The additional benefit of SDA which allows us to perform the binary classification in more than one dimension doesn't result in a performance difference either. A large factor in the performance of the classification procedure is the GentleBoost feature selection. Selecting the right number of features has a larger influence on SDA performance than finding the optimal subclass division. The effects of the feature selection are similar for LDA and SDA, regardless of the specific subclass division applied.

In the multi-class case, we also see little effect of the subclass division on the classification performance. The samples are spread along a single manifold whose characteristic shape is not influenced by the

subclass division. We do see a benefit in performing the classification in more than one dimension, however in the multi-class case, LDA has this ability as well.

The SDA variants consistently show a smaller performance increase over the LDA variants but this difference might not be significant. The one-versus-all and the multi-class approaches do have a significant performance difference to the benefit of the multi-class approach. This is consistent with the expectations due to the ordered nature of the data.

11.3 Discussion

When we perform the subclass division by k-means we note that the subclasses are based partially on pose and partially on identity. Subsequently we manually assigned samples to subclasses according to their pose. Assigning the samples according to the subjects identity results in less distinct subclasses and no performance benefits. This method would result in a larger number of subclasses and thus reduce the number of training samples per subclass available in the training set, which would negatively affect the training process. Additonally, large parts of the identity specific image features are lost due to the application of the Gabor filters and the subsequent feature selection which favours pose specific features over identity specific features. An option more likely to be successful is a subclass division according to different categories of occluders. Large occluders such as glasses and hats as well as the non-occluded face could each form a subclasses. The features required to distinguish these subclasses could more easily be selected in the feature selection stage than the smaller identity specific features. The possibilities are mostly limited by the diversity and quantity of the samples in the data set.

The combined feature set for multi-class SDA is sufficient for pose classification but it may not be optimal. The one-versus-all classifiers show that the required number of features to correctly identify a specific pose class is not equal for all pose classes. This suggest the creation of a combined feature set with more features dedicated to the identification of the difficult poses and less features for the easier poses.

There is a significant performance difference between the one-versus-all approach and the multi-class approach. This is likely due to the ordered nature of the data and the capability of discriminant analysis to order the data on a manifold. However, the performance gap between the multi-class approaches and the combined GentleBoost approach is not very large. When the capabilities of the manifold are not fully used, as in discrete classification as opposed to continuous classification, the one-versus-all approach remains valid.

Bibliography

[1] Tinku Acharya and Ajoy K. Ray, *Image processing: Principles and applications*, Wiley-Interscience, September 2005.

[2] Sileye O. Ba and Jean Marc Odobez, *Evaluation of multiple cues head pose tracking algorithm in indoor environments*, International Conference on Multimedia & Expo (ICME), 2005, IDIAP-RR 05-05.

[3] E. Bailly-Bailliere, S. Bengio, F. Bimbot, M. Hamouz, J. Kittler, J. Mariethoz, J. Matas, K. Messer, V. Popovici, and F. Poree, *The BANCA database and evaluation protocol*, Proc. Audio-and Video-Based Biometric Person Authentication (AVBPA) (2003), 625–638.

[4] Vineeth Nallure Balasubramanian, Sreekar Krishna, and Sethuraman Panchanathan, *Person-independent head pose estimation using biased manifold embedding*, EURASIP J. Adv. Signal Process **8** (2008), no. 2, 1–15.

[5] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, *Eigenfaces vs. fisherfaces: Recognition using class specific linear projection*, IEEE Transantions on Pattern Analysis and Machine Intelligence **19** (1997), no. 7, 711.

[6] R. Bellman, *Adaptive control processes: A guided tour*, Princeton University Press, 1961.

[7] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft, *When is Nearest neighbor meaningful?*, Database Theory ICDT99, 1999, pp. 217–235.

[8] M. Bicego, A. Lagorio, E. Grosso, and M. Tistarelli, *On the use of SIFT features for face authentication*, Computer Vision and Pattern Recognition Workshop, 2006. CVPRW '06. Conference on, 2006, p. 35.

[9] J. Black, M. Gargesha, K. Kahol, P. Kuchi, and S. Panchanathan, *A framework for performance evaluation of face recognition algorithms*, ITCOM, Internet Multimedia Systems II, Boston (2002).

[10] Leo Breiman, *Bagging predictors*, Machine Learning **24** (1996), no. 2, 123–140.

[11] D. Chai and K.N. Ngan, *Face segmentation using skin-color map in videophone applications*, Circuits and Systems for Video Technology, IEEE Transactions on **9** (1999), no. 4, 551–564.

[12] Bo Chen, Li Yuan, Hongwei Liu, and Zheng Bao, *Kernel subclass discriminant analysis*, Neurocomputing **71** (2007), no. 1-3, 455–458.

[13] Li-Fen Chen, Hong-Yuan Mark Liao, Ja-Chen Lin, and Chin-Chuan Han, *Why recognition in a statistics-based face recognition system should be based on the pure face portion: a probabilistic decision-based proof*, Pattern Recognition **34** (2001), no. 7, 1393–1403.

[14] T. F Cootes, C. J Taylor, D. H Cooper, J. Graham, et al., *Active shape models-their training and application*, Computer Vision and Image Understanding **Vol. 61, No. 1** (1995), pp. 38–59.

[15] T. F. Cootes, G. V. Wheeler, K. N. Walker, and C. J. Taylor, *View-based active appearance models*, Image and Vision Computing **20** (2002), no. 9-10, 657–664.

[16] T.F. Cootes, G.J. Edwards, and C.J. Taylor, *Active appearance models*, Computer Vision ECCV98, 1998, p. 484.

[17] Y. Freund and R. E. Schapire, *Experiments with a new boosting algorithm*, Machine Learning: Proceedings of the Thirteenth International Conference (1996), 148–156.

[18] J. Friedman, T. Hastie, and R. Tibshirani, *Additive logistic regression: a statistical view of boosting*, Annals of Statistics **28** (2000), no. 2, 337–374.

[19] Andrew Gee and Roberto Cipolla, *Determining the gaze of faces in images*, Image and Vision Computing **12** (1994), no. 10, 639–647.

[20] Athinodoros S. Georghiades, Peter N. Belhumeur, and David J. Kriegman, *From few to many: Illumination cone models for face recognition under variable lighting and pose*, IEEE Trans. Pattern Anal. Mach. Intell. **23** (2001), no. 6, 643–660.

[21] N. Gourier, D. Hall, and J. L. Crowley, *Estimating face orientation from robust detection of salient facial structures*, Proceedings of Pointing 2004 (Cambridge, UK), 2004.

[22] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, *Multi-PIE*, Automatic Face & Gesture Recognition, 2008. FG '08. 8th IEEE International Conference on, 2008, pp. 1–8.

[23] Ralph Gross, *Face databases*, Handbook of Face Recognition, 2005, pp. 301–327.

[24] Ralph Gross and Vladimir Brajovic, *An image preprocessing algorithm for illumination invariant face recognition*, Audio- and Video-Based Biometric Person Authentication, 2003, p. 1055.

[25] C. Harris and M. Stephens, *A combined corner and edge detector*, Alvey vision conference, vol. 15, 1988, p. 50.

[26] Jingu Heo and Marios Savvides, *Face recognition across pose using view based active appearance models (VBAAMs) on CMU Multi-PIE dataset*, Computer Vision Systems, 2008, pp. 527–535.

[27] J. Huang, X. Shao, and H. Wechsler, *Face pose discrimination using support vector machines (SVM)*, Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on, vol. 1, 1998, pp. 154–156 vol.1.

[28] Machine Perception Laboratory, *GENKI*, http://mplab.ucsd.edu, 2009.

[29] M. Lades, J.C. Vorbruggen, J. Buhmann, J. Lange, C. von der Malsburg, R.P. Wurtz, and W. Konen, *Distortion invariant object recognition in the dynamic link architecture*, Computers, IEEE Transactions on **42** (1993), no. 3, 300–311.

[30] S. Z. Li, X. G. Lu, X. Hou, X. Peng, and Q. Cheng, *Learning multiview face subspaces and facial pose estimation using independent component analysis*, Image Processing, IEEE Transactions on **14** (2005), no. 6, 705–712.

[31] Stan Z. Li and Anil K. Jain, *Handbook of face recognition*, 2005.

[32] Rainer Lienhart, Alexander Kuranov, and Vadim Pisarevsky, *Empirical analysis of detection cascades of boosted classifiers for rapid object detection*, Pattern Recognition, 2003, pp. 297–304.

[33] D. Little, S. Krishna, J. Black, and S. Panchanathan, *A methodology for evaluating robustness of face recognition algorithms with respect to variations in pose angle and illumination angle*, IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05), vol. 2, 2005.

[34] Chengjun Liu and H. Wechsler, *Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition*, Image Processing, IEEE Transactions on **11** (2002), no. 4, 467–476.

[35] Dang-Hui Liu, Kin-Man Lam, and Lan-Sun Shen, *Optimal sampling of gabor features for face recognition*, Pattern Recognition Letters **25** (2004), no. 2, 267–276.

[36] D.G. Lowe, *Object recognition from local scale-invariant features*, Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on, vol. 2, 1999, pp. 1150–1157 vol.2.

[37] A. M. Martinez and R. Benavente, *The AR face database*, Tech. Report Technical report 24, Computer Vision Center (CVC), Barcelona, Spain, June 1998.

[38] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre, *XM2VTSDB: the extended M2VTS database*, Second International Conference on Audio and Video-based Biometric Person Authentication, vol. 964, Citeseer, 1999, pp. 965–966.

[39] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K.R. Mullers, *Fisher discriminant analysis with kernels*, Neural Networks for Signal Processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop, 1999, pp. 41–48.

[40] H. P. Moravec, *Towards automatic visual obstacle avoidance*, Proceedings of the 5th International Joint Conference on Artificial Intelligence, 1977, p. 584.

[41] Erik Murphy-Chutorian and Mohan Manubhai Trivedi, *Head pose estimation in computer vision: A survey*, Pattern Analysis and Machine Intelligence, IEEE Transactions on **31** (2009), no. 4, 607–626.

[42] Maja Pantic, Michel Valstar, Ron Rademaker, and Ludo Maat, *Web-based database for facial expression analysis*, IEEE Int'l Conf. on Multimedia and Expo 2005, July 2005, pp. 317–321.

[43] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek, *Overview of the face recognition grand challenge*, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005, vol. 1, 2005.

[44] P. Jonathon Phillips, Harry Wechsler, Jeffery Huang, and Patrick J. Rauss, *The FERET database and evaluation procedure for face-recognition algorithms*, Image and Vision Computing **16** (1998), no. 5, 295–306.

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

[46] R. E. Schapire, *The boosting approach to machine learning: An overview*, Nonlinear Estimation and Classification (2003), 149–172.

[47] B. Scholkopf, A. Smola, and K. R Muller, *Nonlinear component analysis as a kernel eigenvalue problem*, Neural computation **10** (1998), no. 5, 12991319.

[48] Caifeng Shan and Wei Chen, *Head pose estimation using spectral regression discriminant analysis*, Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, 2009, pp. 116–123.

[49] Shiguang Shan, Wen Gao, Yizheng Chang, Bo Cao, and Pang Yang, *Review the strength of gabor features for face recognition from the angle of its robustness to mis-alignment*, Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, vol. 1, 2004, pp. 338–341 Vol.1.

[50] Linlin Shen and Li Bai, *A review on gabor wavelets for face recognition*, Pattern Analysis & Applications **9** (2006), no. 2, 273–292.

[51] LinLin Shen, Li Bai, and Michael Fairhurst, *Gabor wavelets and general discriminant analysis for face identification and verification*, Image and Vision Computing **25** (2007), no. 5, 553–563.

[52] Yun Q. Shi and Huifang Sun, *Image and video compression for multimedia engineering: Fundamentals, algorithms, and standards*, 1 ed., CRC, December 1999.

[53] T. Sim, S. Baker, and M. Bsat, *The CMU pose, illumination, and expression (PIE) database*, Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on, 2002, pp. 46–51.

[54] Stephen M. Smith and J. Michael Brady, *SUSANA new approach to low level image processing*, International Journal of Computer Vision **23** (1997), no. 1, 45–78.

[55] S. Srinivasan and K.L. Boyer, *Head pose estimation using view based eigenspaces*, Pattern Recognition, 2002. Proceedings. 16th International Conference on, vol. 4, 2002, pp. 302–305 vol.4.

[56] Xu-Sheng Tang, Tie-Ming Su, and Zong-Ying Ou, *Optimal gabor features for face recognition*, Machine Learning and Cybernetics, 2006 International Conference on, 2006, pp. 3266–3270.

[57] A. Torralba, K.P. Murphy, and W.T. Freeman, *Sharing features: efficient boosting procedures for multiclass object detection*, Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, vol. 2, 2004, pp. II–762–II–769 Vol.2.

[58] Matthew Turk and Alex Pentland, *Eigenfaces for recognition*, Journal of Cognitive Neuroscience **3** (1991), no. 1, 71–86.

[59] Michel Valstar and Maja Pantic, *Fully automatic facial action unit detection and temporal analysis*, IEEE Int'l Conf. on Computer Vision and Pattern Recognition 2006, vol. 3, June 2006.

[60] P. Viola and M. Jones, *Fast and robust classification using asymmetric AdaBoost and a detector cascade*, (2002).

[61] P. Viola and MJ Jones, *Robust Real-Time face detection*, International Journal of Computer Vision **57** (2004), no. 2, 137–154.

[62] Jian-Gang Wang and Eric Sung, *EM enhancement of 3D head pose estimated by point at infinity*, Image and Vision Computing **25** (2007), no. 12, 1864–1874.

[63] L. Wiskott, J. M. Fellous, N. Kruger, and C. Malsburg, *Face recognition by elastic bunch graph matching*, IEEE Trans. on PAMI **19** (1997), no. 7, 764–768.

[64] Junwen Wu and Mohan M. Trivedi, *A two-stage head pose estimation framework and evaluation*, Pattern Recognition **41** (2008), no. 3, 1138–1158.

[65] M.-C. Yeh, I.-H. Lee, G. Wu, and E.Y. Chang, *Manifold learning, a promised land or work in progress?*, Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on, 2005, p. 4 pp.

[66] L. Yin, X. Chen, Y. Sun, T. Worm, and M. Reale, *A high-resolution 3d dynamic facial expression database*, Tech. report, Technical Report, CS Department, Binghamton University, 2008.

[67] Xiaozheng Zhang and Yongsheng Gao, *Face recognition across pose: A review*, Pattern Recognition **42** (2009), no. 11, 2876–2896.

[68] W. Zhao, R. Chellappa, PJ Phillips, and A. Rosenfeld, *Face recognition: A literature survey*, ACM Computing Surveys **35** (2003), no. 4, 399–458.

[69] Wenyi Zhao and Rama Chellappa, *Face processing: Advanced modeling and methods*, Academic Press, October 2005.

[70] M. Zhu and A. M. Martinez, *Using the information embedded in the testing sample to break the limits caused by the small sample size in microarray-based classification*, BMC Bioinformatics **9** (2008), no. 1, 280.

[71] Manli Zhu and A.M. Martinez, *Optimal subclass discovery for discriminant analysis*, Computer Vision and Pattern Recognition Workshop, 2004 Conference on, 2004, p. 97.

[72] Manli Zhu and A.M. Martinez, *Subclass discriminant analysis*, Transactions on Pattern Analysis and Machine Intelligence **28** (2006), no. 8, 1274–1286.