Face recognition, a landmarks tale

Gert M. Beumer

Graduation committee:

chair and secretary:	
Prof.dr.ir. A.J. Mouthaan	Universiteit Twente, EWI
promoter and assistant promoter:	
Prof.dr.ir. C.H. Slump	Universiteit Twente, EWI
Dr.ir. R.N.J. Veldhuis	Universiteit Twente, EWI
referee:	
Dr.ir. G.C. van den Eijkel MBA	Mecal Focal
members:	
Prof.dr. P.H. Hartel	Universiteit Twente, EWI
Dr. M. Poel	Universiteit Twente, EWI
Prof.dr.ir. P.H.N. de With	Technische Universiteit Eindhoven
Prof.dr.ir. M.J.T. Reinders	Technische Universiteit Delft

This research is conducted within the IOP-GenCom Project IGC03003: BASIS
Signals & Systems group, P.O. Box 217, 7500 AE Enschede, the Netherlands

ISBN 978-90-365-2891-7

FACE RECOGNITION,
A LANDMARKS TALE

DISSERTATION

To obtain
the degree of doctor at the University of Twente,
on the authority of the rector magnificus,
prof.dr. H. Brinksma,
on account of the decision of the graduation commitee,
to be publicly defended on October 16th 2009 at 16:45.

by

Gerrit Maarten Beumer
born on 29 August 1975
in Ede, The Netherlands

This dissertation has been approved by:

the promotor: Prof.dr.ir. C.H. Slump
the assistant promotor: Dr.ir. R.N.J. Veldhuis

It does not do to leave a live dragon out of your calculations,
if you live near him.

-J.R.R. Tolkien, The Hobbit-

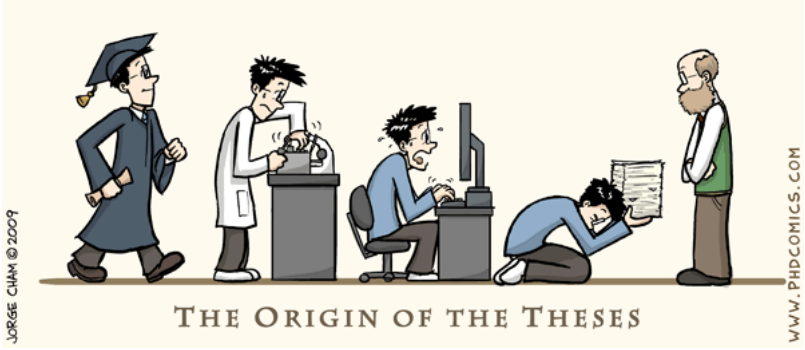
Abstract

Face recognition is a technology that appeals to the imagination of many people. This is particularly reflected in the popularity of science-fiction films and forensic detective series such as Crime Scene Investigation (CSI), CSI New York, CSI Miami, Bones and Naval Criminal Investigative Service (NCIS).

Although these series tend to be set in the present, their application of face recognition should be considered science-fiction. The successes are not, or at least not yet, realistic. This does, however, not mean that it does not, or will never, work. To the contrary, face recognition is used in places where the user does not need or want to cooperate, for example entry to stadiums or stations, or the detection of double entries into databases. Another important reason to use face recognition is that it can be a user-friendly biometric security.

Face recognition works reliably and robustly when there is little variance in pose in the images used. In order to eliminate variance, the faces are aligned to a reference. For this we will use a set of landmarks. Landmarks are points which are easy recognisable locations on the face such as the eyes, nose and mouth.

A probabilistic, maximum a posteriori approach to finding landmarks in a facial image is proposed, which provides a theoretical framework for template based landmarkers. One such landmarker, based on a likelihood



"Piled Higher and Deeper" by Jorge Cham. www.phdcomics.com

ratio detector, is discussed in detail. Special attention is paid to training and implementation issues, in order to minimize storage and processing requirements. In particular, a fast approximate singular value decomposition method is proposed to speed up the training process and an implementation of the landmarker in the Fourier domain is presented that will speed up the search process. A subspace method for outlier correction and an alternative implementation of the landmarker are shown to improve its accuracy. The impact of carefully tuning the many parameters of the method is shown. The method is extensively tested and compared with alternatives.

Although state of the art face recognition still has a giant leap to make, before it is as good as on television, small steps are made by men all the time.

Contents

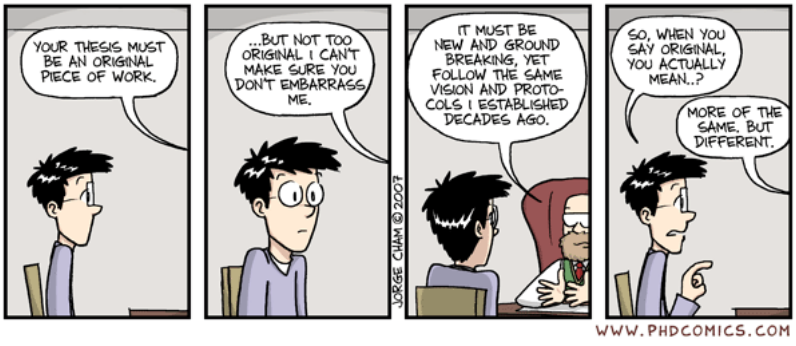
Chapter 1

Biometrics and Face recognition

This chapter is loosely based on previously published material that was presented at ProRISC 2004 conference in Veldhoven [12]

User-convenience, or ease of use, is an important issue when considering security in the residential environment of the year 2010. Biometric authentication, i.e. verifying the claimed identity of a person based on physiological characteristics or behavioural traits, has the potential to contribute to both security and user-convenience.

In this chapter we will start with a short introduction into biometrics in Section 1.1. The use of biometrics will be discussed from a non-technical point of view. In Section 1.2 a short introduction into the terminology of biometrics will be given. Furthermore we will explore, in Section 1.3, face recognition as the biometric tool to use and explore the challenges it gives us. In Section 1.5 we will give an outlook onto this thesis. Finally in Section 1.6 we briefly recapitulate this chapter.



"Piled Higher and Deeper" by Jorge Cham. www.phdcomics.com

1.1 Introduction

The BASIS [50] project, IOP-GenCom Project IGC03003, addresses the use of biometrics in the home environment. The goal of this project is *to investigate the possibilities of biometric authentication for securing the access to information and services in the personal environment, with a focus on user-convenience and privacy protection*. The project was split into three work packages:

1. The problem of transparent biometric authentication as a means to improve user-convenience.

2. The problem of biometric template protection as a means to protect the user's privacy.

3. The specific problems of the use of biometric authentication in the home environment.

In this thesis the first item, the use of face recognition in the home environment, will be discussed. An inventory of some problems and possible solutions will be given. At the University of Eindhoven, Ignatenko *et al.* addressed the second work package [36], [37], [38], [39]. The third work package was covered by the Centre for Mathematics and Computer Science (CWI) by Ambekar *et al.* [3].

In this section we will start with a possible scenario of waking up in a, with BASIS technology equipped, smart house and then continue with various aspects of a house of the future showing applications of biometrics both for security and convenience.

1.1.1 Waking up in a smart home

The alarm clock rings and Susan has to get up. She goes to the shower and the water temperature is adjusted to her preferences. Then Peter steps out the bed and goes to the kitchen to prepare breakfast. As he enters the kitchen the radio switches on at his favourite music channel and the light is adjusted according to his preferences in the morning. Susan enters the kitchen and tells that the message from her aunt Nori came that her plane arrives at 10:30 and she wants to be picked up by car. During the breakfast Susan and Peter discuss who will pick up Nori and who will bring Peter's father, Raymond, to the house, since he would like to see Nori too. Peter says that he will pick up Raymond and that Susan can get Nori and introduce her to BASIS, because she has to stay for two weeks in the house as their guest. At 9:00 Peter and Susan leave the house while Dori is still in the bed, but they left a message for him. BASIS will notify Susan and Peter when Dori gets out of the bed. Dori wakes up and gets out of the bed and hears the message

that his parents left for him. At the moment he leaves the bed, Susan and Peter get the message that Dori is out of bed. He goes to the living room to watch television. It switches immediately to Cartoon Network. It is new to him - his parents only allowed BASIS to give him access to it last week. During a commercial Dori goes looking for some sweets knowing that there are some in the house. He tries every cupboard in the kitchen, but he is not allowed to open the safe cupboard, since only parents have an access to the toxic cleaning materials. Susan and Nori arrive and the door opens, because it recognizes Susan. Inside Susan enrols Nori to BASIS, granting her access to communication devices and the house. At a certain moment Susan gets a message that there is a guest at the door. On the screen she sees Raymond. He is alone, Peter is parking the car. Susan allows Raymond inside.

1.1.2 User-convenience

From a user-convenience point of view, biometric authentication has the advantage that it does not make use of tokens, personal identification numbers or passwords that can be forgotten or lost. Another advantage that biometric authentication offers is the possibility of personalisation, because a device or service can recognise a user and adapt its settings to the user's preferences. Here one could think of the temperature in the house or playing music that everyone present will like. User-convenience can be further increased, when biometric recognition is made transparent. This means that it does not require any specific user action, such as placing a finger on a sensor in order to present a fingerprint.

1.1.3 Security

From a security point of view, biometric authentication offers the possibility to verify whether or not a user is physically present. However, it must be noted that biometric authentication has an intrinsic trade-off between security and user-convenience. We will go into this trade-off more in Section 1.2. Because of this trade-off, not all biometric recognition methods will be able to achieve the same level of security as for example personal identification numbers, passwords, keys, key-cards or any combination of those. Most biometrics, under ideal circumstances, are no more secure then a 4 digit personal identification number, i.e. 1:10000 per attempt. These numbers vary strongly between different biometrics with iris and finger print recognition being very secure while gait and face recognition are less accurate and thus less secure.

1.1.4 Privacy

Considering user privacy, the use of biometric authentication also introduces new problems and raises user concerns. Namely, when used for privacy-sensitive applications, biometric data are a highly valuable asset. When such data are available to unauthorised persons, these data can potentially be used for impersonation purposes, defeating the security aspects that are supposed to be associated with biometric authentication. European privacy legislation provides various protection regimes that cover biometric personal data, depending on their degree of vulnerability and the purpose of their processing. Initial results from studies, done in the context of the European project BIOVISION [2], show that there is a variety of user concerns, associated with loss of privacy, reuse of electronically stored fingerprints and written signatures and the fear that biometric data might reveal medical conditions. One of the most promising privacy enhancing solutions is biometric template protection. Biometric data are called privacy enhanced when the data cannot be traced back to the user or reveals any information about the owner. This means that privacy sensitive information about physiological characteristics cannot be derived from the data. This topic is outside the scope of this thesis, but is covered by Work Package 2 of the BASIS project.

1.1.5 The home environment

The home is a challenging environment for the introduction of biometric authentication. First of all, it is a place where user-convenience and personalisation are highly appreciated or even demanded. Biometric authentication, in particular transparent biometric authentication, seems the security mechanism to achieve this. Secondly, electronic banking and electronic voting will be typically done from the home. These applications require the privacy protection that anonymous biometric authentication can offer. Finally, the home environment poses some specific challenges that need to be addressed. For example, in contrast to access-control or banking applications, there is no professional system manager, who can assist with the enrolment and withdrawal of users, or who can set up and maintain biometric databases. This conflict of interests is not unique for the home environment. It extends to many other fields such as video surveillance at airports, stadiums, public transport etcetera, where the intrusion upon people must be minimal. The application of biometrics in the home environment is covered by work package 1. The system integration of multiple biometrics and application in the home are covered by Work Package 3 of the BASIS project.

1.2 Terminology

In this section we discuss some of the terminology in biometrics. To show this more easily we recall two persons from our previous example, Dori and Nori, and a recognition system or application, called Guardian.

1.2.1 Training, enrolment and testing

In order for Guardian to be able to recognise persons by their biometric features he first has to learn what makes individuals different from each other. The process of learning these distinguishable features is called training. For commercial systems this has often already been done by the manufacturer.

Guardian is now capable of discriminating between different individuals but has knowledge of neither Nori nor Dori. In the next phase, enrolment, Guardian learns the individual characteristics of Dori, Nori and others. This is typically done by the owner of the system during installation. After this, Guardian is ready to recognise people.

Now that Guardian can recognise Nori and Dori the system is operational. A possible evaluation of the system is called testing. Testing is often done on a large, representative data set. Figure 1.1 shows the three phases, namely: training, enrolment and recognition. If the system were to be installed the third phase would be recognising users.

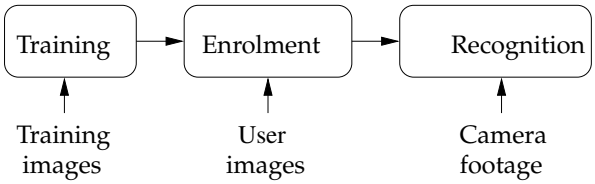


Figure 1.1: Three phases of operation of a camera based biometric system.

1.2.2 Identification and verification

Guardian can work in identification mode, verification mode or a combination of both. In identification mode Guardian tries to identify a person without any prior identity claim. Guardian will decide whether the person is Dori, Nori or one of the other persons that have been enrolled. In that case, even if it is someone who has not been enrolled, it will simply state whom the person is most similar to. In verification mode Guardian will verify the claim that Nori is indeed person Nori with sufficient certainty. Guardian can work in a combination. Then it will first determine the identity

in identification mode followed by a verification step, using the result from the identification as the identity claim. This is shown in Figure 1.2.

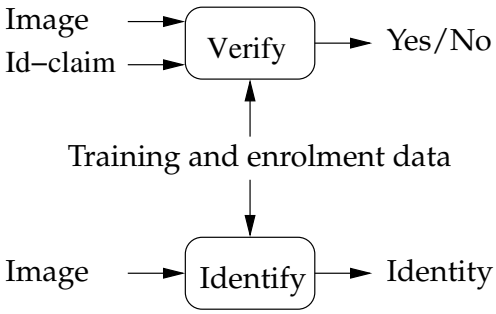


Figure 1.2: Identification and verification.

1.2.3 Genuine and imposter attempts

If Dori claims to be Dori and asks Guardian to verify his identity this is a so called genuine attempt. On the other hand, if Nori or anyone else, except Dori, would claim to be Dori this is called an imposter attempt.

1.2.4 False Accept Rate and False Reject Rate

When Guardian is in verification mode there are a few measures that characterise its performance. The basic operation of Guardian is that it evaluates an identity claim and the measured data. This evaluation will result in a similarity score. This similarity score is a measure for the likelihood that the person is indeed who he claims to be. A high score means that it is probable that the identity claim is true. A low score shows little confidence in the validity of the identity claim. Acceptance or rejection of this claim will depend on a threshold. If the similarity score is higher than the threshold, the identity claim is accepted, otherwise rejected. Plotting the probability densities for both genuine attempts and imposter attempts gives us a graph as shown on the left in Figure 1.3. In most realistic systems both densities, imposter and genuine, will overlap. When we choose a threshold, some of the genuine attempts will be wrongfully denied access resulting in a false reject. At the same time some of the imposter attempts will results in a similarity score which is over the threshold, resulting in a false accept. In Table 1.1 the four outcomes are schematically given in a confusion matrix. A False Accept Rate (FAR) is the portion of imposter attempts which has a score over the threshold. Likewise, the False Reject Rate (FRR) is the portion of genuine attempts which is erroneously rejected. It is easy to see that by

increasing the threshold the FAR is reduced and the FRR is increased. This increases the security of the system. Lowering the threshold the FRR becomes smaller and the FAR grows. This will reduce the security but increase the convenience because the users will experience less false rejects. Access to the vault of a bank will require a low FAR, the slightly higher FRR is an acceptable loss. Grip pattern recognition on a police firearm [61] will require a low FRR because the implications of a false reject are life threatening. In the right part of Figure 1.3 we show a Receiver Operating Characteristic (ROC). It gives the relation between the FRR and the FAR for all possible values of the threshold. The ROC is a characteristic of Guardian. In order to compare different verification systems the ROCs could be plotted together. If we want a single number as indication of the performance the FAR is given for a given FRR or vice versa. A point often used for this is the Equal Error Rate (EER), where both are the same. A lower EER indicates less overlap between the genuine and imposter probability densities, which is good.

Table 1.1: Confusion matrix.

	Genuine attempt	Imposter attempt
Claim accepted	True positive	False positive
Claim rejected	False negative	True negative

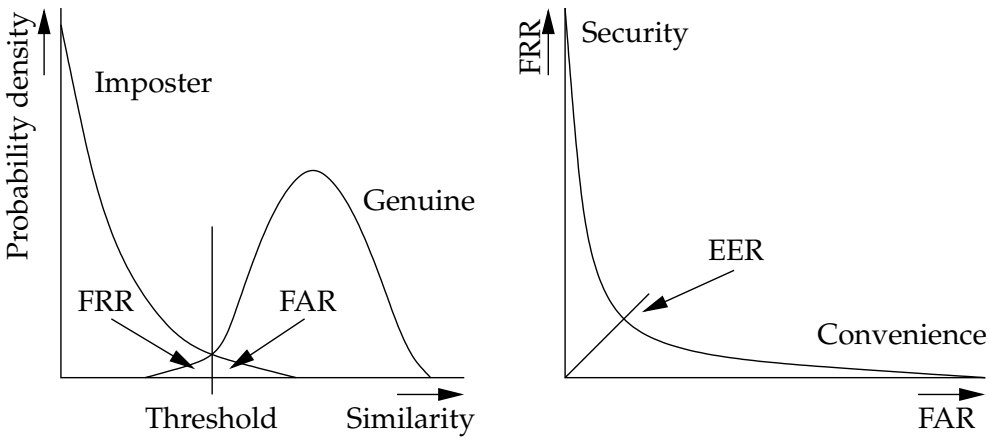


Figure 1.3: Left: probability densities for both imposter and genuine attempts. Right: an ROC curve.

1.3 Face recognition

Most people will know biometrics, especially face and fingerprint recognition, only from the biometric passport or popular media, mostly from films and series such as CSI. This has led to the perception the technology is far more powerful and accurate than current state of the art. However, this does not mean that biometrics is not a good and useful technology for identifying people reliably. Face recognition, for example, can work on very small images [14]. Biometrics can, under the right circumstances, identify someone or confirm someone's identity with acceptable certainty.

In this section we will first discuss what the requirements for a biometric technology are, in order to be considered transparent. After that, we will briefly outline the working of a face recognition system. Finally, we discuss various sources of variability because it is a source of problems for face recognition. Understanding these problems will enable us to make face recognition more robust and accurate.

1.3.1 Transparent biometrics

As said, biometrics is a way to identify a person by body characteristics or traits. Already a lot of biometric recognition methods are known such as fingerprint, face, iris, speaker, odour, gait, posture, grip recognition etcetera. A good overview of various biometrics and their basic operation was given by Jain *et al.* [42]. Not all are as suitable for the home environment, due to costs, performance, transparency requirements and other reasons.

Transparency means that in order to be recognised a person does not have to perform any explicit action. Thus any biometric that does require user action such as fingerprint, grip and iris recognition is, at least with current technology, unsuitable because the person has to present a finger, hand or eye to a sensor. Face, posture, gait recognition are examples of biometrics that can be applied in a transparent way. Our research focuses on face recognition. This is because in our opinion it offers the possibility to be adapted to transparency and does not involve patented technology. Face recognition lends itself well for transparent use because it is based on cameras. An additional is that cameras can also be used for other biometrics such as gait recognition or posture recognition.

1.3.2 Face recognition system

A real face recognition system could work as follows:

1. **Find the face.** A typical face recognition system will work on images that contain a face. The exact location of the face is usually not known. Therefore the face needs to be located first.

2. **Find landmarks in the face.** In this step we try to locate landmarks in the face. Landmarks are stable and recognisable points in the face like the nose, mouth and both eyes. This is done because the next step, registration, needs it.

3. **Register the face.** Registering the face is preprocessing the image in order to correct for certain variations. It can correct for small variations in pose and expression. It uses the locations of landmarks to do this. This is done to make the last step, recognition, more accurate and robust. It is a rigid or deformable alignment to a reference.

4. **Feature reduction.** The preprocessed face is taken and the number of features is reduced. Usually this is done for two reasons. The first is to reduce the amount of data. The second reason is to create a maximal separation between the classes, or individuals, in order to boost the performance.

5. **Recognise the face.** During the last step the feature vectors are taken and then classified. From this follows either an identity or confirmation of an identity claim.

The first three steps actually are often composed into one step called preprocessing.

In this project we started building a complete face recognition framework. First we implemented steps 3, 4 and 5: images with known landmarks and an available face recognition algorithm. This resulted in a demonstrator which we used to show that the quality of the landmarks is of key importance for the recognition.

1.3.3 Variability

Variability is the fact that two images of a person taken for identification can differ due to numerous reasons. An example of how this variability makes it, even for humans, difficult to see the difference between two persons is shown in Figure 1.4. As a consequence of the transparency requirement, users in the house will not perform any action to be recognised but just follow their daily routine. Also, in the house the environmental conditions cannot be controlled as in a laboratory. In most face-recognition systems there is a controlled situation with controlled illumination conditions, a fixed frontal pose, neutral expression. In a transparent environment this is not the case and the conditions are far from ideal, which leads to a high variability. We list a few causes for problematic recognition. Examples of the first four causes are shown in Figure 1.5.

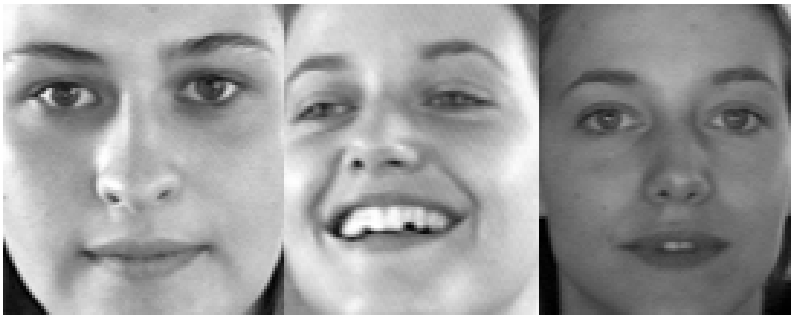


Figure 1.4: There are three images of two persons. Even for a human it is not easy to tell them apart.

1. **Pose** The pose is how the person is facing the camera. This is often not in a frontal or prescribed way. This means that the images on which the recognition is to be based will be frontal, profile, from the back of the head or from any other angle. Also it is obvious that the distance to a camera will give big variability in scale.

2. **Illumination** Apart from pose there are also differences in the illumination conditions. This is caused not only by the difference in position of cameras in a multi camera system. The conditions in the house itself may vary where one can think of sunlight coming through the window or the switching on or off the lights in the house.

3. **Occlusion** When an image of a person is an unobstructed frontal shot all his facial features are visible. However, part of the face can be hidden behind the head itself. Also, parts could be hidden behind objects such as furniture or other persons in the room. The fact that not all of the face is visible means that certain features are unknown.

4. **Expression** People are living and interacting with emotions showing from their faces. This makes that images taken from persons in the home environment will contain images of people with different expressions. This can cause problems if the system is not trained well enough to be able to cope with the expressions. These variations are in general fast and could change over seconds.

5. **Temporal changes** "People change" is a well known saying. This is also true for their faces. It can be anything from people starting to grow a beard to ageing effects. In face recognition this is not a good thing. When the system is trained to recognise people and the people change, the recognition might start to fail or stop working properly.

Figure 1.5: Variability of faces. Pose, illumination, occlusion and expression.

The variability encountered can be roughly separated into two groups; intrinsic and extrinsic variability. Intrinsic variability is variability coming from the person itself. They can be fast, such as expression, or slow as ageing. Extrinsic variability are basically variabilities which are caused by the position of the camera, illumination or other outside influences.

1.3.4 Registration

In order to correct for variations we should register the face. Not many face recognition methods explicitly state which methods for registration they use. Face localization methods can be seen as simple holistic, i.e. based on the entire face, registration methods. The level of information in here is limited. A good overview of overview of face detection methods is given in [72]. Most methods only provide location and scale, while some also provide orientation, width/height aspect ratio or a subset of these. Sometimes these methods are used in combination with finding landmarks such as the eyes to rule out false positives from the face finder [60]. This type of holistic

registration methods therefore lacks accuracy when it comes to registration.

To obtain a more accurate result a second step is needed. A more accurate method could be based on rigid or deformable registration. Rigid registration allows only translation, rotation and scaling. Deformable registration non-linearly changes the proportions within the face. Both rigid and deformable methods often use landmarks for this.

Because in the home environment the variability present creates a demand for accurate registration we need an accurate landmarking method. Using accurate landmarking will result in more accurate registration and thus in higher security or user-convenience levels. Focus on landmarking not only will benefit biometrics in the home environment, but it will also benefit related fields of research such as (3D)-pose correction, biometrics for mobile devices [62], video surveillance and expression analysis. Expression analysis could play a role in the home environment to enable it to become mood-aware, adapting the environmental settings in the house to one's state of mind.

1.4 Purpose of the research

As stated in Section 1.1 BASIS Work Package 1 deals with *The problem of transparent biometric authentication as a means to enhance user-convenience.* As explained in Section 1.3 we chose face recognition as the biometric modality for the home environment. Thus the context of our research is to uncover how face recognition can be applied in the home environment. For this the challenges specific for the home environment need to be identified.

A large amount of research has been carried out on face recognition methods and many good academic [34], [47], [41], [71] and commercial systems exist [54]. There are many different methods, all with their own strengths and weaknesses. Few or no methods target the home environment specifically. Most commercial systems integrate all stages detailed in Section 1.3 into one system. This makes these systems less suited for us because they are not optimized for the home environment and are not flexible enough to be adapted. Also, a commercial system is not transparent enough for our purposes, often due to a lack of knowledge of the used methods. Therefore we choose to build our own recognition system. A combination of PCA [64] and LDA [6] is a well proven and robust method for feature reduction. It can easily be followed by a likelihood ratio classifier. The face recognition system that we used will be described in Chapter 2. Because the biggest problems are due to variability, it seems prudent to address this in the preprocessing stage as much as possible. Therefore, the most important steps in the preprocessing will be face localization and registration. For the first step we used the Viola and Jones algorithm [69], which has been

proven to work well and fast. This step is clearly not the bottle neck of the system. While setting up a complete face recognition system we discovered that a large step forward could be achieved by improving landmarking, as we will show in Chapter 2. Because of the variability encountered in the home environment it is likely that some landmarks will be subject to distortions. The information provided by the other landmarks can help to make the estimation of the distorted landmark location more accurate and efficient by limiting the search space. Our work on landmarking has been laid down in Chapter 3, Chapter 4 and Chapter 5.

In sum, our research will focus on the importance of registration for dealing with the variability encountered in the home environment. Special attention will be given to the development of landmarking methods, as a cornerstone of accurate registration methods. **The research questions addressed in this thesis are:**

1. *What is the relation between landmarking accuracy and face recognition performance?*
2. *Can a statistical classifier approach be used for landmark detection?*
3. *Can the underlying statistical relationship between landmark locations be used to improve landmarking?*
4. *Which methods can be used to reduce computational complexity and thus also overcome the computational problems which arise from very large training sets?*

1.5 Overview of the thesis

1.5.1 Registration

Once we have localized a face in an image we can use it for training, enrolment or recognition. There will be some pose variations in the images. These variations are caused by inaccuracy of the face finder and the fact that people may not look directly into the camera. It is wise to remove small variations in pose instead of modelling them prior to training or recognition. This is done in a separate step called 'registration'. Usually this means aligning it to a reference. The alignment process consists of translation, rotation and scaling. The reference is very often a set of landmarks, for example the average shape. Not all registration methods are landmark based. Zitova *et al.* [73] give an extensive, though not complete, survey of the different registration methods. Not all registration methods use landmarks. Registration on the entire face is called holistic registration. Boom *et al.* [15] got good results by registering on the matching score

in face recognition. Other holistic registration methods can be based on rotation invariant correlation in the spectral domain [48, 59, 58], correlation on super resolution images [44] or using correlation to find the optimal rigid transformation [45, 49].

We however aim at landmark based registration because the variance within the landmarks is smaller than within the entire face. The landmarks therefore can be found more precisely than the face. Also using more landmarks will reduce noise and errors. This will be discussed in Chapter 2.

Research by Riopka *et al.* [57] and Cristinacce *et al.* [23] showed that precise landmarks are essential for a good recognition performance. In Chapter 2 we will also show that proper landmarking is of prime importance for the improvement of registration. We will show it to be the weakest link in our entire face recognition system and therefore make it the focus of our research. Chapter 2 is an adaptation of work previously published [7].

1.5.2 Landmarking

In Chapter 3 we present a statistical method for landmarking. We show that good and accurate results in landmarking can be obtained by means of a simplified Bayesian classifier. Much attention is given to the proper implementation, tuning and training of the algorithm in Chapter 4. Chapter 3 and Chapter 4 are a continuation of work which has been presented at the FG2006 [8]. Both are combined into one paper which has recently been accepted by the Journal of Multimedia for publication [11].

1.5.3 Prior knowledge

Prior knowledge is a tricky thing to define. When working with trained classifiers, one could argue that all data used is prior knowledge. This is however, not how we would like to define it. We define prior knowledge as knowledge about the outcome of the classifier, which is not part of the input of the classifier. In our case: the locations of the landmarks and their underlying relationships. Each landmarker is trained on images of either eyes, noses or mouths. They do not use information from other landmarks when training a landmarker. This results in landmarkers trained to find a nose, mouth or eye. Training the classifiers we only used the data relevant to the particular landmark. The prior knowledge we now use, is the underlying relationship between the landmarks. Imagine: both eyes are roughly on the same height, the nose and the mouth are below each other, etcetera. This prior knowledge can be modelled statistically and used.

In Chapter 5 we expand the methods from Chapter 3 and we will show that the proper use of prior knowledge of the inter landmark relationship is useful. Using this information explicitly instead of implicitly can make the

landmarking algorithms more efficient and theoretically more sound. We will show that the use of prior information in landmarking improves the results. Chapter 5 is loosely based on previously published work [10]

1.6 Discussion

In this chapter we introduced the context of our research and gave a short outline of the terminology in biometrics. In Section 1.3 we outlined face recognition within our research and analysed its potential and weaknesses. This leads to the focus on registration in Section 1.5.

The proposed landmarking methods are not only useful to find features in a face. They can be used to refine any machine vision application where accurate positioning is needed but where registration on the whole object is for any reason not practical. A few examples could be to register the picture of certain types of fruit prior to inspection, industrial inspection of parts or the alignment of custom print work prior to cutting.

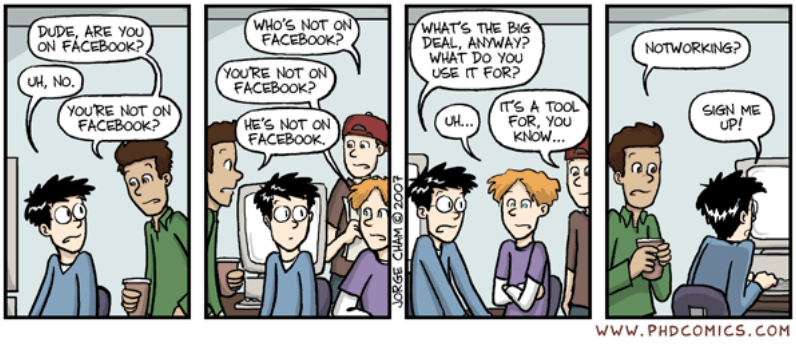
Chapter 2

On the recognition performance importance of registration

This chapter is loosely based on previously published material that was presented at SPS-DARTS 2005 conference in Antwerp. [7]

2.1 Introduction

Imagine that you and your companions embarked on an adventure into unknown lands, with only a map to guide you to your goal. If unsure about the road ahead you would turn to the map. The first thing you would do is look around, to see which landmarks are there. Your group is travelling east with misty mountains in the distance to the west. There is a river, running from north to south, with a ford, which you just crossed. A dark forest arises in the east. With this information, your relative position to all these landmarks, you will be able to find your location on the map and continue



"Piled Higher and Deeper" by Jorge Cham. www.phdcomics.com

your journey. The more accurately you know your relative position to the landmarks, the better you can determine your position on the map. The reverse is also true; the accuracy of the landmarks on the map is equally important for your navigation. Determining one's position on the map is actually registering the map to one's surroundings. Both the accuracy of your estimate of the landmark locations and the accuracy of the landmarks on the map, is of direct influence on how well you will register the map to your surroundings.

2.1.1 Accuracy of the verification rate

Before we evaluate the impact of the accuracy of landmarks and registration on face recognition, we need a good method and measure to evaluate their impact in a statistically valid way. For the performance of the face recognition system we will use the EER as discussed in Section 1.2.

An indication of the reliability of error rates is seldom given, though they depend strongly on the number of tests and the way in which the data are split into a training and testing set. How we split the datasets is explained in Section 2.3. Error rates, such as the EER, easily vary by a factor of two as a result of different splits between training set and testing set. Therefore, a single error rate without the information on how it has been estimated or an estimate of its reliability is hardly informative. We will propose to include an estimate of the reliability of performance measures with the measure itself. This is discussed in Section 2.4.

2.1.2 Robustness to noise

In the field of face recognition, registering a face to a reference is not much different from registering a map to one's surroundings. In the cartography example the quality of navigation depends on the registration of the map. Likewise, we expect the quality of face recognition to depend on the quality of the registration. Since we want better face recognition we argue that it is worthwhile to examine the relationship between registration accuracy and face recognition robustness.

In order to do this, we perform some recognition experiments where the registration is distorted by noise on the landmarks. Riopka and Boult [57] performed similar experiments with noise added to the position of the eyes during registration. We will discuss the face recognition algorithm that we used in Section 2.2. The experiments determine the relation between landmarking accuracy and face recognition performance. These experiments are discussed in Section 2.3 and in Section 2.4 the results will show that the recognition performance is sensitive to proper registration.

2.2 Face recognition

In this section the algorithm used for face recognition is discussed. It should be noted that it has not been optimised and that tuning of parameters most likely will improve the overall performance. This is, however, not necessary in order to evaluate the sensitivity to landmarking accuracy during registration.

2.2.1 The algorithm

Preprocessing

The first step is registration. We tested two different registration methods. One uses rigid transformation while the other uses a deformable method to generate a so-called shape free patch (SFP) [21].

Rigid registration The registration is rigid. This means that by means of rotation, translation and scaling the Euclidean distance of some or all landmarks to a set of reference landmarks is minimised. Affine transformation can only correct for in-plane variations. Both rigid registration and the SFP are explained here.

Shape free patch A deformable method deforms the image so that all landmarks are at fixed positions. This is useful to compensate for a wider range of pose variations and to a limited extent, expressions. We apply a non-linear transformation, using thin-plate splines [13]. This transformation warps each face image to an SFP, in which the texture has been made invariant of shape variations. Note here that warping to a SFP is not a rigid transformation.

Vectorization from ROI

After registration the images containing the faces are cropped to 251 pixels high and 231 pixels wide. The centres of the eyes in the reference image are 100 pixels apart. From this image a fixed region of interest (ROI) that contains most of the face is selected. All grey scale values in the ROI are put into a feature vector $\vec{x}$. The ROI is visualised in Figure 2.1. In order to return to a full description of the face image, the shape free patch-based feature vector can, optionally, be extended with the shape information: the deviations of 20 landmark locations with respect to their means.

Figure 2.1: Region of Interest.

Linear transformation

To each measurement vector a linear transformation is applied. The transformation, under Gaussian assumptions, reduces the dimensionality, turns the total covariance matrix into an identity matrix and diagonalizes the within class covariance matrix. We assume all persons to have identical within class covariance matrices.

From the images we calculate the probability density function (PDF) of all users called the total PDF, or background PDF. This is a multi-variate Gaussian of which we determine the total covariance matrix, Λ_T. The images from all persons are placed over the feature space. In Figure 2.2 we schematically illustrate this. In the upper left corner we show the initial PDFs prior to linear transformation. The large oval denotes an equal probability contour of the PDF while the smaller ovals represent equal probability contours of the PDFs of individual users. We transform the data by rotation, scaling and again rotation. After this, the total variance is identical in all directions and the individual users can de projected onto the horizontal axes without loosing separability. This is illustrated in the lower right corner of Figure 2.2. The transformation matrix is determined during training as explained in Section 2.2.3.

Log likelihood ratio

The extracted feature vector, $\vec{y}$, is then compared to class i. This is done by calculating a log likelihood based matching score S:

$$S_i(\vec{y}) = -(\vec{y} - \vec{\mu}_i)^T \Lambda_W^{-1} (\vec{y} - \vec{\mu}_i) + \vec{y}^T \vec{y} - \log|\Lambda_W|, \tag{2.1}$$

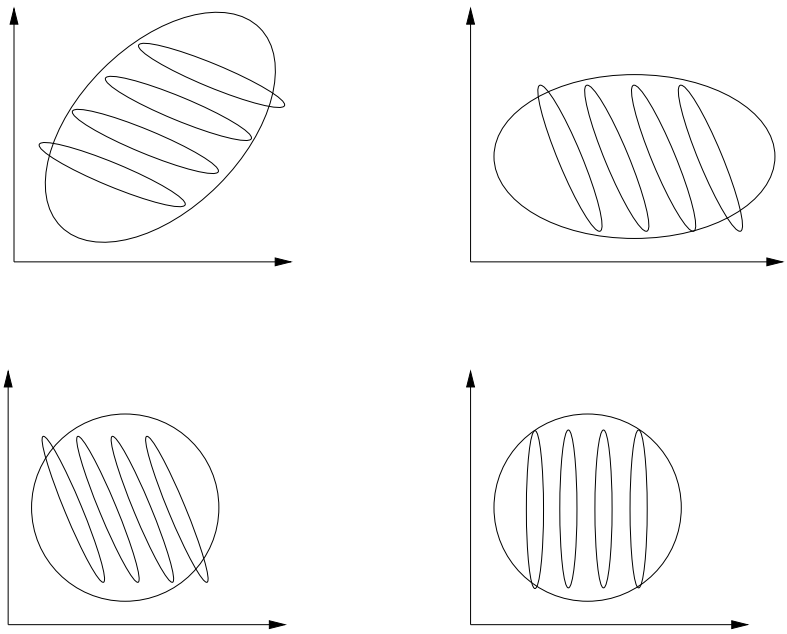


Figure 2.2: Transformation of the feature space by means of rotations and scaling. After transformation and projection onto the horizontal axes, the data are still separable.

where $\vec{\mu}_i$ denotes the, as template enrolled, class mean and Λ_W the within class covariance matrix.

Accept of reject

By comparing $S_i(\vec{y})$ to a threshold, L, we determine whether the identity claim is accepted or rejected.

2.2.2 Enrolment

The stored templates are the class means of the feature vectors in the reduced feature space. For each class to be enrolled, the linear transformation of the class mean is determined.

2.2.3 Training

The transformation matrix is calculated in a training phase. The training is done using a combination of the Eigenfaces [64] and Fisherfaces [6] method:

- First apply Principal Component Analysis (PCA) on the training data after subtracting the mean. After a subsequent dimension reduction the number of features is chosen to be twice the number of classes.
- Then apply a linear discriminant analysis (LDA), making the total covariance matrix, Λ_T, unity. After a subsequent feature reduction the number of features is the number of classes in the training set minus one [67]. Store the within class covariance matrix, Λ_W, total average, $\vec{\mu}_T$, and the transformation matrix, T.

In the testing phase a feature vector, $\vec{x}$, is projected onto the reduced feature space by premultiplying it with the transformation matrix, i.e.

$$\vec{y} = T(\vec{x} - \vec{\mu}_T) \tag{2.2}$$

2.3 Experiments

In this section we describe the experiments. In one experiment we investigate both the sensitivity to noise and the accuracy of the EER. First we will explain the recognition set-up followed by a brief explanation of the details for both experiments.

2.3.1 Experimental set-up

We used repeated random sub-sampling cross-validation with random partitioning [26], [46]. This means that the data are split into a training set and a testing set. A fixed fraction (e.g. 50%) of each class is randomly selected and put into the training set. The remainder is put in the testing set. The training set is also used for the enrolment. After each split we perform one run.

A run consists of splitting the data into a training set and a testing set, training the classifier, enrolling the data and running the classification experiment on all images in the testing set. One run thus gives us matching scores for both the imposter and genuine attempts.

2.3.2 Accuracy of the error rate

The EER calculated after one run is not the reliable estimate one might expect. A more reliable estimate as well as an indication of the standard deviation can be obtained from more runs. There are two methods to use the results of n simulation runs.

1. Calculate an EER for each run. Average all the EERs from the individual runs and calculate the standard deviation:

$$EER \approx \frac{1}{n} \sum_{i=1}^{n} EER_i, \tag{2.3}$$

$$\sigma_{\texttt{calc}} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(EER_i - EER)^2}. \tag{2.4}$$

2. Accumulate the matching scores S from all n runs. After that, determine the EER of the system. The estimated standard deviation can be calculated using the results from the first method:

$$\sigma_{\texttt{est}} = \frac{\sigma_{\texttt{calc}}}{\sqrt{n}} \tag{2.5}$$

.

The first method makes it possible to calculate the standard deviation as a reliability measure of the average EER, but the estimate of the EER in method 1 may be biased. For each run the EER is found as an optimum at a different threshold value L. In reality L is fixed. The true EER can thus significantly differ from the average EER for the first method. The second method does not have this problem and will give a better or equally reliable estimate of the real EER. The drawback is that because there is only one EER a standard deviation cannot be calculated. A combination of both methods solves this problem. For n large enough we expect $\sigma_{\texttt{est}}$ to converge to the same number for both methods. Then the standard deviation of the first method can be used to make an estimate of the standard deviation of the second method.

Part of the experiment aims to investigate the accuracy and validity of the EER. We therefore group the data in bins. A bin is defined as a number of runs over which all similarity scores are accumulated. The EER that is given is an average EER over all bins. The EERs of all runs divided over several bins are then averaged and the standard deviation is determined. One should note that the standard deviation may not be an ideal measurement to indicate the reliability of the EER because the distribution of the EER is possibly not Gaussian and therefore we do not know which portion of the EERs is within one standard deviation.

This experiment used two landmarks for rigid registration and had no noise added to the labelled landmarks.

2.3.3 Robustness to noise

In this section we discuss how to examine the robustness to the noisy registration. Gaussian distributed zero mean noise with a known standard deviation was added to the landmark coordinates before the registration. In total four different registrations are used.

We expect that the performance degrades severely when the noise level is increased. This was also observed by Riopka and Boult [57]. Furthermore registering 20 landmarks promises better recognition than registering only

two landmarks because the noise on the landmarks is Gaussian and equally distributed in all directions and is averaged out in the 20 landmarks. We choose to use two -both eyes- and 20. The maximum number of landmarks was chosen because we expect it to result in the lowest error. The lowest number of landmarks to determine registration was chosen to maximize the influence of noise. These two will give upper and lower bounds for the error. Apart from two rigid registrations we used two deformable methods. One is the SFP, the other is the SFP with the coordinates added to the feature vector. We expect the SFP and the SFP with shape information to outperform the rigid registration, when no noise is added. They may be more sensitive to noise than the rigid registration, because there is no averaging out of the noise on the registration landmarks and all the noise contributes to the SFP and shape.

2.4 Results

The results were obtained by simulations on the BioID [43, 35] database which comes with 20 labelled landmarks in the face, as illustrated in Figure 2.3. These landmarks are around the eyes, nose, mouth and chin. In the database there are images of 23 individuals, with high diversity in number of images per person. The minimal number of persons per class is two and the maximum is 150.

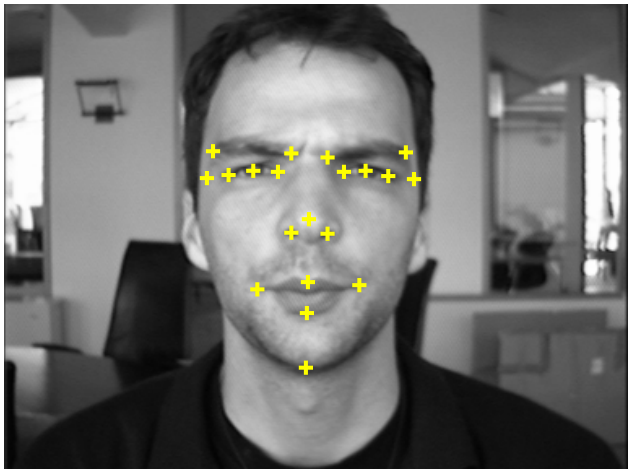


Figure 2.3: Landmarks as provided by the BioID database.

Not all 1521 faces are completely within the ROI due to the fact that parts of the face may be outside the original image. Because of this we used a fixed subset of 1389 images of which 689 for the training and enrolment set and 700 for the testing set. Per run there are 700 genuine attempts and 15400 imposter attempts.

2.4.1 Accuracy of the error rate

In Table 2.1 the results of one simulation of 250 runs are given. The average EER for different bin sizes is given. It should be noted that the EER given, is the average EER from all bins.

It shows that for bin size 10 and 50, σ_{est} is a reasonable approximation of σ_{calc}. It should be noted that $\sigma_{\texttt{calc.}}$ for binning size 50 is calculated over only 5 EERs. This makes the estimate of the standard deviation on the EER of 250 runs per bin acceptable. The average EER does not change significantly. Strong bias effects cannot be found on the BioID database.

Table 2.1: EERs for rigid registration using two landmarks without added noise.

Bin size	EER [%]	$\sigma_{\texttt{calc.}}$ [%]	$\sigma_{\texttt{est.}}$ [%]
1	2.94	0.47	-
10	2.94	0.15	0.15
50	2.94	0.06	0.07
250	2.94	-	0.03

A standard deviation of 0.5% on an EER of 2.9% is large. Table 2.1 shows that in order to be sure of the first two digits, around 50 runs is the minimum. It should be noted that this number applies only to this database. For different databases different EERs and standard deviations apply and thus a different number of runs are needed in order to obtain an EER with an acceptable reliability.

For Gaussian distributions it is known that approximately 68% or 95% of the results lie within one respectively two times the standard deviation. It is safe to assume that for our real world problem it is not ideally Gaussian distributed. However, for the example worked out in Table 2.1 the distribution is unimodal. For this particular split in training set and testing set, we observed that of the EERs $\frac{170}{250} = 68\%$ or $\frac{239}{250} \approx 95\%$ are inside the one or two standard deviation respectively. This complies with what could be expected from a Gaussian distribution. Still we assume that the estimated standard deviation is an acceptably reliable measure for the variance of the EER and should be presented along with the EER.

We, therefore, conclude that, when publishing error rates the number of runs and, or at least, the $\sigma_{\texttt{est.}}$ should be given in order to be able to make valid comparisons to other work.

2.4.2 Robustness to noise

We added noise with standard deviations of 0 to 5 pixels to the landmark coordinates. After registration a check was done to see whether we had not included a region into the ROI that was not in the original image. This is unlikely because all images which do not contain a full face were rejected but due to the noise on the labelled landmarks this could occur. If this occurred it was simply reported and the results were ignored. For each experiment with different settings 250 runs were done. An attempt to include parts outside the original image dimensions into the ROI only occurred a few times. For noise with a standard deviation of 4 pixels it occurred 13 times and for noise with a standard deviation of 5 pixels it occurred 38 times out of $1389 \times 50 = 69450$ generated images. Both for alignment on two landmarks. For alignment on 20 landmarks or the SFPs this effect was not detected.

In Figure 2.4 some examples of badly warped or registered images are shown. The noise has a standard deviation of 3 pixels.

Figure 2.4: Registration which wrong zoom and rotation (left) and SFP showing strange deformations (right).

The results for the robustness to noise simulations are as was expected: the error rates increase when the amount of noise on the labelled landmarks rises. The alignment with 20 landmarks performs better than with only two landmarks and is more robust to noise. This can be seen in Figure 2.5 and Table 2.2. For the alignment on two landmarks this is also what Riopka and Boult [57] found but the results for our PCA/LDA implementation do not appear to degrade as fast as the PCA implementation in [57].

Using SFPs the performance is about the same as the alignment on 20 landmarks but it is less robust to noise. This concurs with our expectations. The results for the SFP with shape information are the best. At low distortion they outperform all the other methods but the equal error rate as function

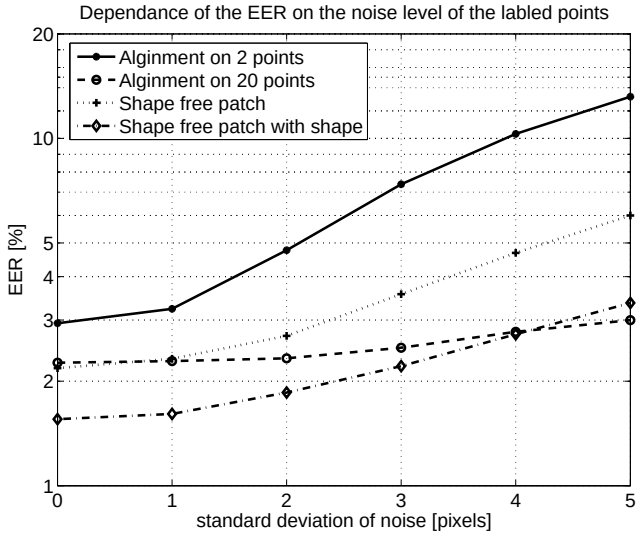


Figure 2.5: The EER as a function of the noise on the labelled points prior to registration.

Table 2.2: The EER over 250 runs and for zero added noise.

	EER [%]	$\sigma_{\text{est.}}$ [%]
Registration on two landmarks	2.94	0.03
Registration on 20 landmarks	2.26	0.03
SFP	2.18	0.03
SFP+shape	1.55	0.02

of the noise grows faster than for the alignment on 20 landmarks and it is therefore less robust. The sensitivity of SFP to noise is caused by the fact that the noise is not averaged over the number of landmarks, causing unpredictable distortions of the face. This was illustrated in Figure 2.4

It is interesting to note that for all systems the performance for added noise with a standard deviation of one pixel and without noise is approximately equal. This leads to the conclusion that the labelled landmarks have an intrinsic noise with a standard deviation in the order of one pixel.

2.5 Conclusions

We aimed to evaluate the relationship between the quality of landmarks used for registration and the outcome of a recognition experiment. In order to do so we also proposed to present the numerical results, such as error rates, in

a statistically valid format. In our case, when EERs are presented, both the number of runs and the estimated standard deviation should be given, in order to estimate the confidence interval of the results. When evaluating the results one should be aware of possible bias effect in the results.

Registration on two labelled landmarks is most sensitive to noise. The overall performance is less than that of other methods. Registration on 20 landmarks however is much more robust and also performs a lot better. Using more landmarks seems to improve registration. Using a shape free patch and the shape information combined does outperform all other methods for low noise but is less robust to noise than straight forward alignment on 20 landmarks. When using an automated face finder for an automatic face recognition system it is important to find enough landmarks which are reliable enough. If this is not done the error rate will be too high.

We also showed that by using bins a good estimate of the standard deviation of the error rates, and thus their accuracy, can be made.

The positive influence of both using more and more accurate landmarks on the outcome of a face recognition experiment confirms our expectations that better registration leads to better face recognition for all registration methods and it underlines the importance of accurate landmarking.

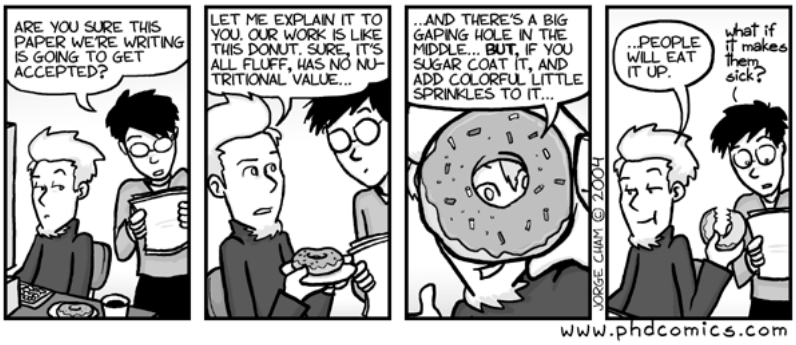
Chapter 3

A Practical Subspace Approach To Landmarking

3.1 Introduction

In his book 'Climbing Mount Improbable' [25], Richard Dawkins nicely illustrates that the evolutionary path that most likely leads to survival, is like climbing a mountain: Mount Improbable. At the bottom of Mount improbable we find the first simple life. All paths up the mountain start here. The more evolved a species is, the higher it is located on the mountain. Every species that lives and ever lived has his own unique spot on the mountain. The evolution of a species travels up the mountain by walking the easy road, not by taking the shortest route from the bottom to the top via the steep side. One small step at a time. Wolves have their own spot on the mountain. Near the wolf are his cousins such as foxes, dogs, coyotes and jackals.

Assume, for sake of the argument, that the wolf is at the highest top of



"Piled Higher and Deeper" by Jorge Cham. www.phdcomics.com

Mount Improbable. This means that we can say: the higher on the mountain, the more likely it is that it is indeed a wolf. Close to the summit we find the wolf's cousins such as jackals, foxes, dogs and coyotes. Evolution within an ecological niche favours certain features over others, namely the ones that enhance its chance of survival. This is similar to how a classifier works. Instead of an ecological niche we have training data to favour the features that make up an eye. Our classifier, Mount Eye, has the ideal eye at the very top. If something looks like an eye, and it looks like an eye, it probably is an eye.[1] The higher something ends up on Mount Eye, the more likely it is an eye. For each possible location in the face we estimate how high it would score on Mount Eye. We assume that the location with the highest score, the most likely location, is the location of the Eye. This example shows the general working of any landmarker.

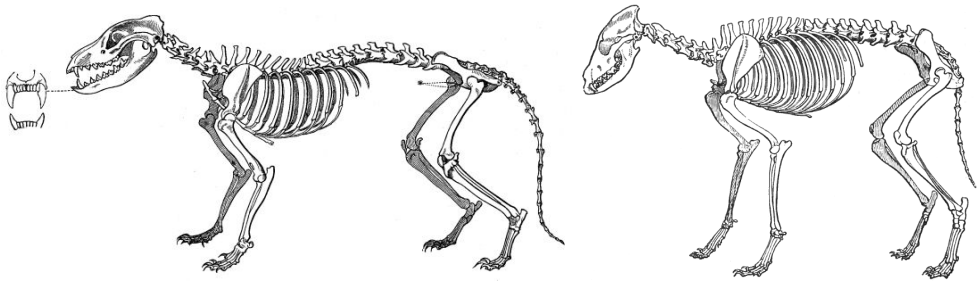


Figure 3.1: Left: Skeleton of a Tasmanian tiger. Right: Skeleton of a wolf. Images from [16]

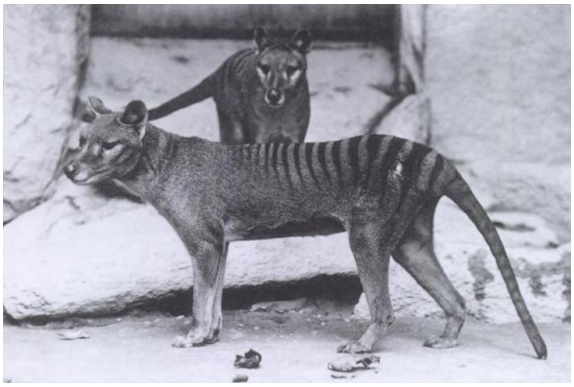


Figure 3.2: Tasmanian tigers.

[1] This is not a typo.

The Tasmanian tiger, see Figure 3.2, is a, since 1936 extinct, marsupial from Australia and Tasmania. It looks very similar to the wolf, it takes an expert to see the difference between the skull of a wolf and that of a Tasmanian tiger. Both evolved within the same ecological niche after having drifted apart since the beginning of life on earth. In Figure 3.1 we can see that both have very similar skeletons. Even though the Tasmanian tiger and the wolf look very similar, the paths up Mount Improbable are completely different. Still, both paths ended very close to each other on the mountain, possibly even closer to the wolf than his cousins the fox, coyote and jackal.

This shows the weakness of a classifier. Give an expert two skulls and he can tell you which one is the wolf and which one is the Tasmanian tiger. The task becomes more difficult when one or both skulls are damaged. The details which enable the expert to determine the difference are gone and he is much more likely to make mistakes. The image samples of the eye, the mouth or both can be damaged, by whatever possible cause. When this happens, to a classifier, an eyebrow can look more like an eye than the real eye, even though it is at a completely illogical location in the face. We give another example: imagine that there is a half open mouth in the image. It shows upper lip, lower lip, teeth and a dark spot between them. It is not hard to see that with some distortion this will look like an eye to the classifier. We assume that the most likely landmark location is the real landmark location and not something which accidentally gets the highest score from the classifier. In Chapter 5 we will address this problem, and how to reduce the likeliness of this kind of mistake.

In Chapter 2 we showed the importance of accurate landmarking. Here, in Chapter 3, we present a simplified Bayesian method for landmarking, namely the Most Likely Landmark Location (MLLL). In Chapter 4 we will show this method to be a good method for accurate landmarking. MLLL is a continuation of work by Bazen *et al.* [5]. It was proposed first at the Face and Gesture recognition conference in Southampton in 2006 [8]. Continuation of this work has recently been accepted by the Journal of Multimedia publication [11] and is the basis of Chapter 3 and Chapter 4. The text has been included without major changes, except layout, typos and that both appendices and references have been moved to the end of this thesis.

A first step towards an accurate landmarker

At the FG2006 [8] we showed that good and accurate results in landmarking can be obtained by means of a simplified Bayesian classifier. From ongoing research we learned that MLLL could be improved further. Several possible improvements were identified. First of all the dataset, BioID, which was used to train MLLL, would be a good upgrading candidate since it contained only 1521 images from only a very small number of people; 23 individuals.

Replacing this small training database with a larger one turned out to give rise to several challenges such as memory constraints. At the same time evaluating the large amounts of images gave rise to the need for a more efficient version of the MLLL algorithm. Also, within the MLLL algorithm there are numerous parameters that can be tuned for more efficient and accurate performance.

A new theoretical foundation for MLLL is presented in Section 3.2. In Section 3.2 we also present an improved version of MLLL, which is not only a lot more efficient but at the same time performs more accurately. This is followed by two practical solutions for implementation problems that arise due to the size of the training data. First an Approximate Recursive Singular Value Decomposition (ARSVD) algorithm is presented as a solution for computational limitations, regarding computer memory and processing time, using subspaces. Secondly, the MLLL is implemented in the spectral domain. Finally, in Section 3.5 the conclusions are presented.

3.1.1 Importance of registration for face recognition

Accurate registration is of crucial importance for good automatic face recognition. Although face recognition performance has improved greatly over the last decade [54], better registration will still lead to better recognition performance.

Many, but not all, registration systems use landmarks for the registration. A landmark can be any point in a face that can be found with sufficient accuracy and certainty, such as the location of an eye, nose and mouth. Some examples of landmarks are shown in Figure 3.3. The markers denote the landmarks as included in the BioID [43, 35] database (left) or FRGC [56] database (right). Riopka and Boult [57], Cristinacce and Cootes [23], Wang *et al.* [70], Campadelli *et al.* [17] and Beumer *et al.* [7, 8], and others have shown that precise landmarks are essential for a good face-recognition performance. In [7], for example, it was shown that more accurate landmarking brings a higher recognition performance and that using more landmarks results in higher recognition performance.

Besides face recognition there are other applications, such as positioning or measurement in an industrial setting, for which the detection of a landmark in an image with high accuracy is desirable.

3.1.2 Related work

Currently a popular approach is to use adaptations of the Viola-Jones [68] face finder for landmarking [23, 19, 18]. We use a version of that method in this paper as a reference algorithm. The original Viola-Jones method uses weak Haar classifiers and a boosted training method known as Adaboost.

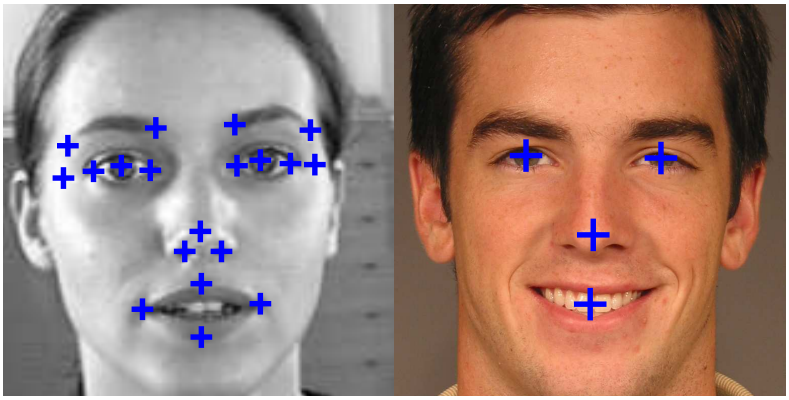


Figure 3.3: Landmarks as provided by the BioID database (left) and the FRGC database (right).

Multiple variations to this have been proposed. For example, Wang *et al.* [70] use this method in combination with different classifiers for eye detection. Because the Haar classifiers only represent rectangular shapes they propose to use multiple weak Bayesian classifiers assuming Gaussian distributions.

Campadelli *et al.* [17] made a different variation on to the Viola-Jones classifier. They used a combination of Haar classifiers and Support Vector Machines to create an eye detector. The Haar classifiers do not work on the image texture but on their wavelet decomposition.

Cristinacce and Cootes [22] present a landmarking method called Shape Optimized Search where probability of the constellation of landmarks is used to predict where the landmarks are to be expected. Then, they use one of three different landmark detectors to refine the search. Active Shape Models (ASM) [20] and Active Appearance Models (AAM) [21] can also be used for finding landmarks but both methods need good initialization for accurate results. These initialization must be provided by another method.

Everingham and Zisserman [28] use three statistical landmarking methods, namely a regression method, a Bayesian approach and discriminative approach. The second method calculates a log likelihood ratio between landmark and background samples i.e. samples not containing a landmark. Everingham concludes that the Bayesian approach performs best compared with much more complicated algorithms. The Bayesian implementation is essentially the same as earlier work by Bazen *et al.* [5]. Moghaddam and Pentland [51] used PCA to find landmarks.

3.1.3 Our work

In this chapter we continue earlier work by Bazen *et al.* [5] and Beumer *et al.* [8]. A new theoretical foundation for the Most Likely Landmark Locator (MLLL) [8] is presented in Section 3.2. This is followed by two practical solutions for implementation problems that arise due to the size of the training data. First, an Approximate Recursive Singular Value Decomposition (ARSVD) algorithm is presented as a solution for computational limitations, regarding computer memory and processing time, which occur if the training data grows in volume. The ARSVD tackles this problem using subspaces. Second, a spectral implementation of MLLL will be derived, allowing for a more than tenfold speed-up of MLLL. These new modifications render MLLL a practical and accurate method for landmarking.

The application MLLL was designed for, is frontal face recognition with limited variation of pose and illumination. This implies that the landmarks will not be occluded, that they will be in predictable locations and that there will be no projective deformations. In more advanced versions of the proposed method, however, these constraints could be relaxed or dropped.

Two additions to MLLL are proposed. The first is a subspace-based outlier detection and correction method named BILBO [8] that is capable of detecting and correcting erroneous landmarks. The second addition is a repetitive implementation of landmarking, The Repetition Of Landmark Locating (TROLL), which will improve accuracy. Both BILBO and TROLL can be used in combination with MLLL but can also work with any other landmarking algorithm. BILBO will be discussed in Section 3.3 and TROLL in Section 3.4.

3.2 Most Likely Landmark Locator

In this section we will present the Most Likely Landmarks Locator. First, a theoretical framework for landmarking will be presented. After that some implementation issues will be addressed. In order to speed up the computations we introduce a frequency domain implementation. Also the Approximate Recursive Singular Value Decomposition (ARSVD) is presented as a solution for computing large volume databases using subspaces.

3.2.1 Theory

Let the shape $\vec{s}$ of a face be defined as the collection of landmark coordinates, arranged into a column vector. The texture samples of the face are within a region of interest (ROI) and also arranged into a column vector, $\vec{x}$. The

maximum a posteriori estimate (MAP) [65] of the location of the landmarks, $\vec{s^*}$, given a certain texture $\vec{x}$, can be written as

$$\vec{s^*} = \underset{\vec{s}}{\operatorname{argmax}}\, q(\vec{s}|\vec{x}), \tag{3.1}$$

where $q(\vec{s}|\vec{x})$ denotes the probability density of the shape given image $\vec{x}$. According to Bayes rule, Equation 3.1 can be rewritten as

$$\vec{s^*} = \underset{\vec{s}}{\operatorname{argmax}}\, \frac{p(\vec{x}|\vec{s})}{p(\vec{x})} q(\vec{s}), \tag{3.2}$$

where $p(\vec{x}|\vec{s})$ can be recognized as the probability density of the texture $\vec{x}$ given a shape $\vec{s}$. Furthermore, $p(\vec{x})$ denotes the probability density if the landmark locations are unknown. Finally, $q(\vec{s})$ is the probability density of the shape. The quotient in Equation 3.2 is the likelihood-ratio of the texture belonging to shape $\vec{s}$.

Ideally, one would like to compute $\vec{s^*}$ from Equation 3.2, including the prior probability density $q(\vec{s})$ of $\vec{s}$. In order to reduce the computational complexity we assume $q(\vec{s})$ to be uniform over the region of interest. Therefore $q(\vec{s})$ can be removed from Equation 3.2. Let $\vec{x}_i$ be the texture surrounding the i-th landmark and $\vec{s}_i$ its location. We assume, for practical reasons, that $\vec{x}_i$ only depends on $\vec{s}_i$ and that $\vec{x}_i$ and $\vec{x}_j$, $i \neq j$, are independent. Therefore,

$$\frac{p(\vec{x}|\vec{s})}{p(\vec{x})} = \prod_{i=1}^{l} \frac{p(\vec{x}_i|\vec{s}_i)}{p(\vec{x}_i)}. \tag{3.3}$$

With this simplification the optimization problem in Equation 3.2 can be reformulated as

$$\begin{aligned} \vec{s_i^*} \quad = \quad & \underset{\vec{s}}{\operatorname{argmax}} \sum_{i=1}^{l} \big(\log(p(\vec{x}_i|\vec{s}_i)) - \\ & \log(p(\vec{x}_i)) \big) \end{aligned} \tag{3.4}$$

We assume that the probability density of the landmark texture $p(\vec{x}_i|\vec{s}_i)$ is Gaussian with mean $\vec{\mu}_{l,i}$ and covariance matrix $\Sigma_{l,i}$. Likewise $p(\vec{x}_i)$, which we will denote as the background density, thus emphasizing that x_i may come from an arbitrary location, is Gaussian with mean $\vec{\mu}_{b,i}$ and covariance $\Sigma_{b,i}$. These assumptions have been made for practical reasons, but are mildly supported by the fact that especially after dimensionality reduction, the texture probability density tends towards Gaussian. A more accurate model might be a Gaussian mixture model, but that would be much more complex. Because of the assumed mutual independence of the landmarks, the terms in Equation 3.4 can be maximized independently. This makes that

the estimation of the shape is now simplified to

$$\vec{s^*}_i = \underset{\vec{s}}{\operatorname{argmax}} \left\{ \left(\vec{x}_i(\vec{s}) - \vec{\mu}_{b,i}\right)^T \Sigma_{b,i}^{-1} \left(\vec{x}_i(\vec{s}) - \vec{\mu}_{b,i}\right) - \left(\vec{x}_i(\vec{s}) - \vec{\mu}_{l,i}\right)^T \Sigma_{l,i}^{-1} \left(\vec{x}_i(\vec{s}) - \vec{\mu}_{l,i}\right) \right\} \quad (3.5)$$

for all landmarks $i = 1 \ldots d$. This is identical to the optimization criterion used in MLLL presented in previous work [8]. Equation 3.5 is intuitively pleasing as each term of the summation benefits the similarity to a landmark and penalizes the similarity to the background.

Dimensionality reduction

The covariance matrices, Σ_l and Σ_b in Equation 3.5, need to be estimated from training data. Because landmark templates can be as large as $96 \times 64 = 6144$ pixels, direct evaluation of Equation 3.5 would be a too high a computational burden. Due to the limited number of training samples available in practice, the estimates of the covariance matrices could be rank-deficient. Even if not, they would be too inaccurate to obtain a reliable inverse, which is needed in Equation 3.5.

Therefore, prior to the evaluation of Equation 3.5, the vector $\vec{x}$ will be projected onto a lower dimensional subspace. This subspace should have several properties. First of all, its basis should contain the significant modes of variation of the landmark data. Secondly, it should contain the significant modes of variation of the background data. Finally, it should contain the difference vector between the landmark and the background means, for a good discrimination between landmark and background data. The modes of variation are found by principal component analysis (PCA) on landmark and background training data. After this first dimensionality reduction the landmark and background densities are simultaneously whitened [31], such that the landmark covariance matrix becomes a diagonal and the background covariance matrix an identity matrix in the reduced feature space. The latter whitening step is done for computational reasons. See Appendix A.1 for details of the procedure of the dimensionality reduction.

The previous feature dimensionality reduction steps aimed at creating a good representation of the landmark and background data. In the next feature reduction step we want to select the features that have the highest discriminative power. In this feature selection step, a fixed number of features are kept. The standard Linear Discriminant Approach as proposed by Fisher [30] is not applicable because the covariance matrices $\Sigma_{b,i}$ and $\Sigma_{l,i}$ are different. Instead, our approach is to keep those features for which the mean divided by their standard deviation is maximal. Informal experiments

in which this method was compared with alternatives have shown that this method gave the best results.

Feature extraction and classification

The total process of feature reduction and simultaneous whitening can be combined into one linear transformation by a matrix $T \in \mathbb{R}^{m \times n}$, with n the dimensionality of the training samples and m the final number of features after reduction. The detailed calculation of the feature reduction transformation T is given in Appendix A with the final result in Equation A.12. We aim to reduce the dimensionality while trying to optimize the discriminability between the landmark and non-landmark distributions. The method applied is variation of Approximate Maximum Discrimination Analysis [4].

With T we project the means, covariance matrices and feature vectors onto the subspace, ideally:

$$\vec{\mu}_l' \stackrel{\text{def}}{=} T\vec{\mu}_l, \qquad \vec{\mu}_b' \stackrel{\text{def}}{=} T\vec{\mu}_b. \tag{3.6}$$

$$\Lambda_l \stackrel{\text{def}}{=} T\Sigma_l T^T, \qquad I_b \stackrel{\text{def}}{=} T\Sigma_b T^T. \tag{3.7}$$

$$\vec{y}(\vec{s}) \stackrel{\text{def}}{=} T\vec{x}(\vec{s}). \tag{3.8}$$

where Λ_l is diagonal, I_b is identity, $\vec{y}(\vec{s})$ is the feature vector and $\vec{x}(\vec{s})$ denotes sample values from the ROI at location $\vec{s}$. Please note that Σ and T are estimates obtained from data and, therefore, not exact. Consequently, the resulting covariance matrices after the transformation are only approximately diagonal. After this transformation Equation 3.5 becomes

$$\begin{aligned} \vec{s^*} \quad = \quad & \underset{\vec{s}}{\text{argmax}} \Big\{ \left(\vec{y}(\vec{s}) - \vec{\mu}_b'\right)^T \left(\vec{y}(\vec{s}) - \vec{\mu}_b'\right) \\ & - \left(\vec{y}(\vec{s}) - \vec{\mu}_l'\right)^T \Lambda_l^{-1} \left(\vec{y}(\vec{s}) - \vec{\mu}_l'\right) \Big\}. \end{aligned} \tag{3.9}$$

Note that although Equation 3.9 resembles Equation 3.5, the result will be different due to the dimensionality reduction. Solving Equation 3.9 is, however, computationally far more efficient than solving Equation 3.5.

3.2.2 Approximate Recursive Singular Value Decomposition

Training on large data sets should make MLLL accurate and robust. However, as the amount of training data grows, the calculation of T quickly becomes computationally prohibitive, either because of time or, more

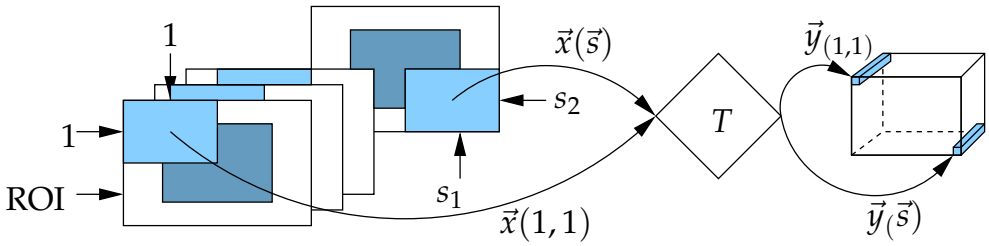


Figure 3.4: Feature extraction in the spatial domain. The pixel values surrounding the location of interest, $\vec{x}(\vec{s}) \in \mathbb{R}^m$ are multiplied with $T \in \mathbb{R}^{m \times n}$. The resulting feature vector $\vec{y}(\vec{s}) \in \mathbb{R}^n$ is of lower dimensionality than $\vec{x}(\vec{s})$.

likely, memory constraints. Especially the Singular Value Decompositions (SVDs) in Equations A.1 and A.9 in Appendix A.1 are troublesome. In order to overcome these problems an Approximate Recursive SVD (ARSVD) algorithm is introduced. Proper application relies on two conditions. The first is that the estimates of the covariance matrix improve when more data are processed. Second, the amount of explained variance kept in each recursion step must be sufficient. As the SVD is part of the feature reduction process, finally only a certain amount of the explained variance is to be kept and the amount of variance kept by the ARSVD should be higher than that. If these two conditions are met, there should be no significant loss of information. ARSVD is fairly straightforward. Let X be a matrix with all feature vectors as columns, split up into a number of submatrices, called blocks, with a fixed number of columns, called the blocksize b:

$$X = [X_1, X_2 \dots X_o] \tag{3.10}$$

Let U_j, W_j and V_j represent the ARSVD after j blocks, i.e.

$$[X_1 \dots X_j] \approx U_j W_j V_j^T \tag{3.11}$$

with $U_j \in \mathbb{R}^{n \times n}$, $W_j \in \mathbb{R}^{n \times b}$ and $V_j \in \mathbb{R}^{b \times b}$. Note that the number of pixels in the samples, n is larger than the blocksize b. The space of $[X_1 \dots X_j]$ is spanned by $U_j W_j$. Adding the next block of data of X and calculating the SVD gives

$$\begin{aligned} [U_j W_j | X_{j+1}] &= \tilde{U}_{j+1} \tilde{W}_{j+1} \tilde{V}_{j+1}^T \\ &\approx U_{j+1} W_{j+1} V_{j+1}^T \end{aligned} \tag{3.12}$$

where $U_{j+1} \in \mathbb{R}^{n \times b}$ and $W_{j+1} \in \mathbb{R}^{b \times b}$ are submatrices of $\tilde{U}_{j+1} \in \mathbb{R}^{n \times n}$ and $\tilde{W}_{j+1} \in \mathbb{R}^{n \times 2b}$ of reduced sizes. Each run the dimensionality retained is

reduced from twice the blocksize to the blocksize. Repeating this until all sub matrices of X are processed will give an estimate of the matrix U and matrix W after a standard SVD. The blocksize is a parameter that has an impact on the accuracy and the speed of the ARSVD.

3.2.3 Frequency domain implementation

Even in the reduced feature space, evaluating Equation 3.9 is still computationally demanding. This is because Equation 3.8 is evaluated at each possible location within a region of interest. A schematic overview of how the spatial algorithm operates is given in Figure 3.4. It can be observed that the calculation of each element of $y(\vec{s})$ is analogous to a filter operation or equivalently a cross-correlation operation. Hence we can make use of the fact that a cross-correlation operation in the spatial domain can be written as, a much less demanding, element wise multiplication in the spectral domain. The conversion to the spectral domain and back again can be done by an efficient implementation of a discrete Fourier transform, thus resulting in a net gain in processing time. As a result the processing time of an implementation in Matlab on a desktop PC was reduced more than tenfold.

Only considering the k-th element of vector $\vec{y}(\vec{s})$ from Equation 3.8 we have

$$y_k(\vec{s}) = \vec{t}_k \vec{x}(\vec{s}) \tag{3.13}$$

with $\vec{t}_k \in \mathbb{R}^{1\times n}$ the k-th row of $T \in \mathbb{R}^{m\times n}$. If $\vec{t}_k$ is reshaped to $\hat{t}_k \in \mathbb{R}^{v\times u}$ it can be seen as a correlation kernel, as seen in Figure 3.5, which is shifted over the ROI.

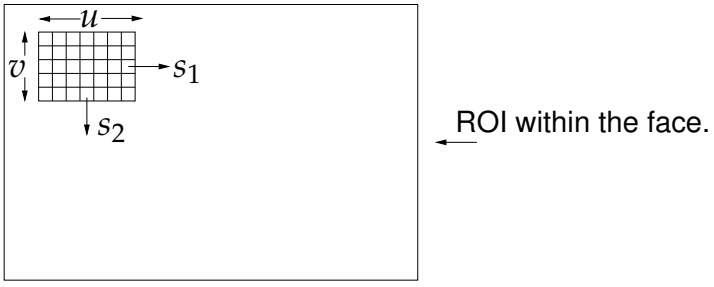


Figure 3.5: Applying the kernel $\hat{t}_k$ to the image. The similarity between the kernel and the image is calculated at all locations (s_1, s_2). Each row in T can be considered to be a single kernel.

At each location $\vec{s}$ this can thus be written as:

$$y_k(\vec{s}) = \sum_u \sum_v \hat{t}_k(u,v) x(s_1+u, s_2+v). \tag{3.14}$$

Because correlation in the spatial domain corresponds to an element wise multiplication of the signal with the complex conjugate of the correlation kernel in the spectral domain [32], we get:

$$\mathcal{F}(y_k(\vec{s})) = \mathcal{F}(\hat{t}_k(\vec{s}))\mathcal{F}(x(\vec{s}))^* \quad (3.15)$$

$$\mathbf{Y}_k = \hat{\mathbf{T}}_k \mathbf{X}^* \quad (3.16)$$

where $*$ denotes the complex conjugate and boldface printing denotes the representation in the spectral domain. The k-th elements of all feature vectors $y_k(\vec{s})$ at all locations $\vec{s}$ are given by the inverse Fourier transform of $\mathbf{Y}_k$. After calculating all $\vec{y}_k$ planes in the region of interest all the feature vectors are known at all locations in this region of interest. In Figure 3.6 this is graphically illustrated. Note the difference with Figure 3.4. All the elements

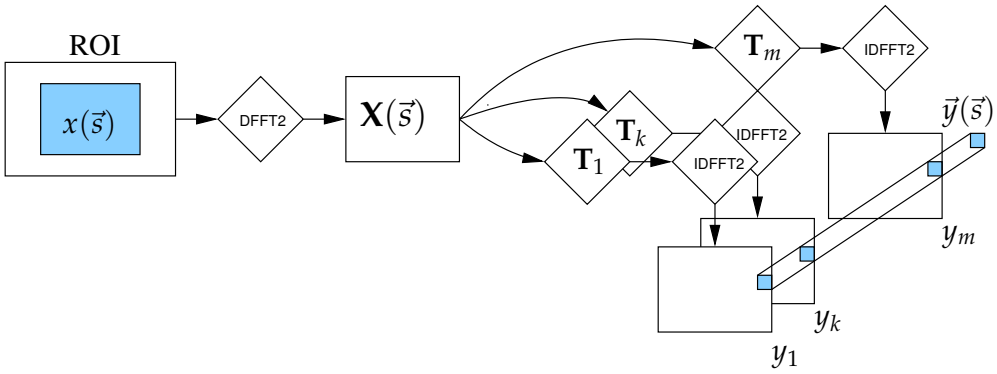


Figure 3.6: Feature extraction in the spectral domain.

of $\vec{y}(\vec{s})$ are calculated for all locations with one multiplication per element.

The spectral correlation kernels, $\hat{\mathbf{T}}_k$, can be pre-calculated during training thus keeping the number of calculations minimal.

In Appendix C the computational complexity of the frequency domain implementation is compared to the Viola and Jones implementation, which is known for its efficiency and speed. The complexities of MLLL and VJ are not essentially different.

3.3 BILBO

The landmarks are disturbed by two types of errors: noise and outliers. The noise refers smaller errors and will be present in every estimate. If a sufficient number of landmarks is used, the effect of noise on the registration will be limited [7]. The outliers are the larger errors, which will seriously distort the registration. In order to reduce these larger errors, we present an outlier

detection and correction method named BILBO. Although we assumed the landmarks to be independent for the derivation of MLLL in Section 3.2, we will now explicitly use the dependence of the locations of the landmarks to correct outliers.

In related research fields subspace methods are used as an effective tool for removing noise from images. This has been done by, amongst others, Muresan and Parks [52], Goossens *et al.* [33] and Osowski *et al.* [55]. By keeping only the dominant features in the subspace and subsequently projecting back to the image space, the noise is reduced. Here we apply the same principle onto the shape. We define a subspace and BILBO projects the shape *there and back again* [63]

3.3.1 Theory

Correct shapes are assumed to lie in a subspace of $\mathbb{R}^{2d}$ with d the number of landmarks. Incorrect shapes, containing one or more erroneous landmarks, are assumed to be outside this subspace. Consider a measured shape $\vec{s}\,'$ that consists of a part $\vec{s}$ which fits the subspace $\mathbb{R}^n$ with $n < 2d$ and an error $\vec{\epsilon}$ which cannot be represented in this subspace.

$$\vec{s}\,' = \vec{s} + \vec{\epsilon} \tag{3.17}$$

Erroneous landmarks correspond to a pair of large elements, ϵ_i, of $\vec{\epsilon}$. BILBO aims to find those landmarks and correct them. We can estimate the error on the measured shape $\vec{s}\,'$ by

$$\vec{\epsilon} = \vec{s}\,' - \left(BB^T(\vec{s}\,' - \vec{\mu}_s) + \vec{\mu}_s \right) \tag{3.18}$$

Large elements of $\vec{\epsilon}$ indicates an outlier. If for a certain landmark the error is above a threshold, τ, its location is replaced with the location after projection.

$$\vec{s}\,'_i = \vec{s}_i \; \forall i \big| |\vec{\epsilon}_i| > \tau \tag{3.19}$$

This procedure is repeated until convergence has been reached, which is usually already after one iteration.

Training BILBO is done by finding the largest variations for all normalized training shapes. Normalised means that the shapes are aligned to a reference shape. The reference shape, which is the average shape when the found face coordinates have been scaled between 0 and 1. Our implementation is explained in Appendix B.1.

Applying BILBO is schematically shown in Figure 3.7. It shows how the error, $\vec{\epsilon}$, is calculated. The error is used to determine which landmarks seem to be wrong and need to be corrected. This is done repetitively until all $|\vec{\epsilon}_i|$

are below the adaptive threshold τ. In Appendix B.2 this will be discussed in more detail.

Though simpler, BILBO shows a resemblance to the Ransac algorithm [29] where also a distinction between "inliers" and "outliers" is made. Also, if too few landmarks are used, BILBO could fail. In this case, a restoration method based on minimizing an expected restoration error. e.g. [66] could provide an alternative.

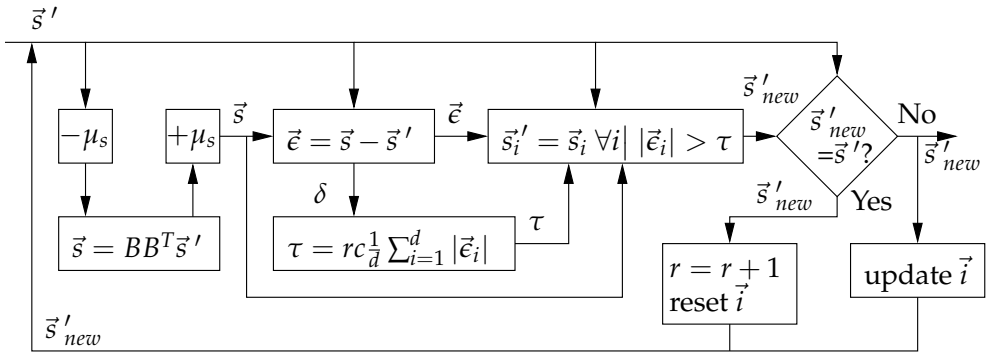


Figure 3.7: A schematic overview of BILBO. The vector $\vec{i}$ keeps track of the landmarks to be updated. A detailed description can be found in Appendix B.2.

3.4 The Repetition Of Landmark Locating

The training images have been registered to a standard scale and pose before extracting the transformation matrix T and the parameters of Equation 3.9. Therefore, these do not fully model the orientation variations that occur in the images when landmarking. Because of this, MLLL would perform best on registered faces. This is, of course, normally impossible as landmarking is one of the steps of registration. We therefore propose to iterate the landmarking procedure. This procedure will be called: The Repetition Of Landmark Locating (TROLL). Once landmark candidates have been found, the image is registered and the landmarking is repeated on the registered image. We use MLLL, in combination with BILBO as the landmarking method, but other landmarking methods could also be used iteratively in the same manner. The processing time is linear with the number of iterations. We will choose the number of iteration such that further iterations yield no significant improvement.

3.5 Conclusions

The landmarking method presented in this section, MLLL, is based on Bayesian classifiers and is presented with a new theoretical framework based on maximum a posteriori. Two important extensions are proposed. BILBO is an outlier correction method and TROLL an iterative implementation of the combination of MLLL with BILBO. Although the setting of this paper is landmarking on facial images the algorithms can be applied to many landmark versus background classification problems in images.

Two solutions to implementation issues are presented, namely the ARSVD and a spectral template matcher. The first makes it possible to do a singular value decomposition on large data with sufficient accuracy.

In Section 3.2 we assumed the landmarks to be independent. This assumption is known to be a simplification of the truth. Dropping this assumption very likely will improve the accuracy and robustness further, because using this dependence in hindsight, as BILBO does, has already shown to improve the results.

In Chapter 4 we will discuss the tuning and training of MLLL, BILBO and TROLL. An extensive parameter optimization will be used to tune the algorithms and test the proposed methods.

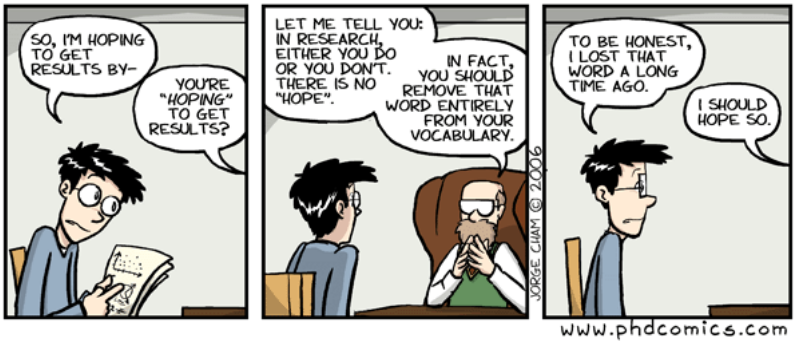
Chapter 4

Landmarker optimization by parameter tuning

4.1 Introduction

In my dictionary [1] the second meaning for the verb *tuning* reads: *to make an engine work as well as possible*. How to tune is completely dependent on the task ahead. The settings on a racing bicycle can be completely different for a curvy and hilly track than for a track with predominantly straight stretches of road. Tuning your bicycle before a ride can make all the difference, between a winning combination and a so called Did Not Finish (DNF) guarantee. Virtually every one has done it: tune something. Some people tune their bicycle, car, computer, clothing, phone or any other thing. We tune a landmarker.

MLLL, BILBO and TROLL all have parameters that have a strong influence on their own and each other's performance. In Section 4.2 we



"Piled Higher and Deeper" by Jorge Cham. www.phdcomics.com

outline the experiments and we will empirically analyse the relation of the parameters of MLLL to the performance of the algorithms.

An evaluation of the proposed methods and a comparison to other methods are presented in Section 4.3, showing that MLLL, especially with the extensions BILBO and TROLL, has a good performance. TROLL yields an error of 3.3% of the interocular distance. This error is obtained with a landmarker of which some of the parameters have not been optimized for specific landmarks, but for the entire set of landmarks. Tuning MLLL for each landmark individually is likely to improve the recognition performance further.

Finally, in Section 4.4 the conclusions are presented which show that the MLLL could be improved significantly compared to the initial implementation.

4.2 Training and tuning

In this section we will discuss the training and tuning of the parameters of MLLL, BILBO and TROLL. The performance of these algorithms has a strong relation to the choice of the parameters.

First, we start with discussing the databases used. Second, this section will focus on tuning of the various parameters and their influence on the algorithms. An overview of these parameters and their final values is given in Table 4.1. Repeatedly one parameter was optimized while all others were kept fixed until a stable solution was reached. We present only the results of the final parameter settings.

In order to evaluate the performance of the methods used we used the same error measure as Cristinacce and Cootes [24]. The error measure, m_e is the mean euclidean distance between the landmarks and the manually labelled groundtruth coordinates as a percentage of the interocular distance Δ_{ocl}.

$$m_e = \frac{1}{n\Delta_{\text{ocl}}} \sum_{i=1}^{n} \sqrt{\delta_{i,x}^2 + \delta_{i,y}^2} \tag{4.1}$$

All results in this section are obtained by landmarking images in the training set. The final results obtained with the fully tuned algorithm on the testing sets are given in Section 4.3.

Sometimes the full parameter space was not explored but only the part where an optimum could be expected because exploration of the full parameter space is not feasible due to time constraints. Although the authors made an effort in finding a good solution it may, therefore, be a local optimum.

Table 4.1: Overview of the tuning parameters and chosen values.

Parameter	final value
MLLL	
Face size	384 [px]
Template size Nose ($n = v \times u$)	48x64 [px]
Template size Eye ($n = v \times u$)	64x96 [px]
Template size Mouth ($n = v \times u$)	64x96 [px]
Relative distance to the landmark	25 [%]
ARSVD blocksize (b)	500
Number of features (m)	219
Explained variance Landmark	81 [%]
Explained variance Background	100 [%]
Explained variance Total	98 [%]
BILBO	
Maximum number of iterations (r)	3
Minimal threshold (τ_{min})	0.055
Error weight (c)	1.15
Number of features in subspace	1
TROLL	
Number of repetitions	3

4.2.1 Databases used

We used two databases from which we drew several datasets for the experiments. Both the FRGC 2.0 [56] and the BioID [43, 35] are publicly available. For testing we only used images in which the face was found by an unsupervised face finder, in this case the Viola and Jones [69] classifier from the OpenCV library with the "frontalface_alt2" cascade [40].

The BioID database consists of 1521 images, taken from 22 persons, which vary in pose, scale and illumination conditions, but which are mainly frontal. All images have been landmarked manually. The Viola-Jones face detector found a face in 1459 of the images (95.9%).

In total, the FRGC 2.0 database contains 39328 images, roughly one third of which are low quality images (LQ) and two third are high quality images (HQ). The FRGC 2.0 comes with hand labelled ground truth locations for four landmarks: the eyes, nose and mouth. We split the FRGC into a training set and a testing set: a training set containing 19674 images with subject ID number 4519 or lower and a testing set containing 19427 (98.8%) found faces in the 19654 images with subject ID numbers 4520 or higher. Both sets contain images from HQ and LQ.

4.2.2 Tuning MLLL

The MLLL has many parameters to tune. In Table 4.1 an overview of these parameters is given. For all parameters we started with an educated guess. Repetitively one parameter was optimized while the others were kept fixed. This was done until for all parameters a final setting was found, based on the landmarking performance in terms of either speed of accuracy.

It was possible, by reusing intermediate results, to keep the training of the algorithm sufficiently fast. Testing the algorithm was however slow because it had to be redone for each new parameter choice. In order to limit the tuning time, the parameters were tuned by landmarking the first 2000 images of the FRGC training set. This limitation implies the risk of overtraining on the first 2000 images of the training set. Verification on the larger dataset showed that this did not happen. Finally, after all parameters have been optimized the error measure, m_e calculated over the first 2000 images of the FRGC training set is 4.06 and over the full set it is 3.89. The fact that over the full set the error is lower suggests that there has been no significant overtraining in the tuning of the parameters.

Image size and landmark region of interest size

Since larger images imply larger areas to scan, the predetermined upper bound was an image size of 384×384 pixels. Experiments showed that smaller images resulted in larger errors. Therefore, the image size was set to 384×384. Note that for computational reasons we chose not to use images larger than 384×384. Improvement might be possible here.

Experiments with the template sizes showed that landscape shaped templates yielded lower errors than square or portrait shaped templates. For the eyes a template size of 64×96 gave best results. For the nose and the mouth the maximum performance was reached with templates of 48×64.

Selection of landmark and background training samples

In order to create a good separation between the landmark samples and the background samples, the background training samples should not include landmark templates. In Figure 4.1 we illustrate how the centre of the background training sample must have a minimal distance to the centre of the landmark. The minimal distance is relative to the width and height of the image, resulting in elliptical regions from which the centres of the background samples are taken. Experiments showed that a distance larger than 0.2 gave significantly better results than smaller distances. To be on the safe side this parameter was set to 0.25. The ellipse denoting the maximum distance had the same radius as half the template size, resulting in an

elliptical doughnut where the centres of the background training samples are taken from.

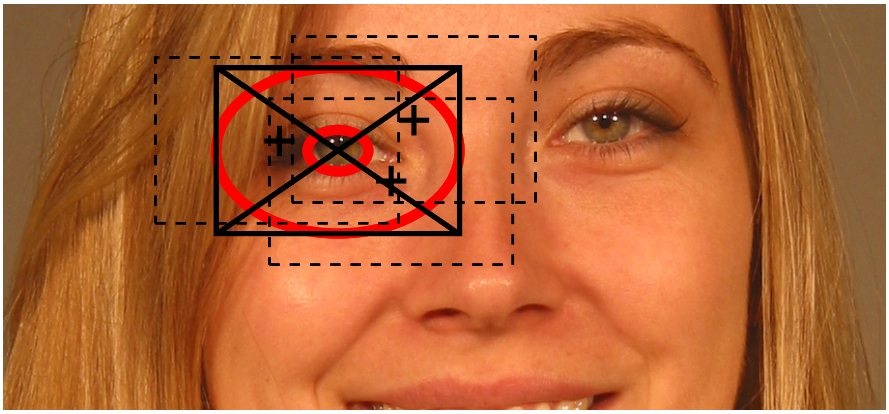


Figure 4.1: Training sample selection. The landmark training sample is a rectangular region around the landmark, denoted with the solid rectangle and cross. Within this region a subregion is defined. This elliptic doughnut shaped area is the region where the centres of the background training samples, denoted by the pluses, are chosen from. Three examples are given as rectangles with a dashed border.

Block size

The block size in the ARSVD algorithm must be large enough to capture all the variation. It turned out that it is not a parameter with a very large influence on the final result as long as it is larger than 300. To be on the safe side we chose 500, as illustrated in Figure 4.2. For the HQ smaller block sizes would be allowed than for the LQ. In Table 4.2 the amount of kept variance for a block size is given for both Landmark and Background samples. In Figure 4.3 the amount of kept variance is illustrated for a blocksize of 500. It shows clearly that each time a block is added the variance within the blocks is modelled better. Finally near 100% of the variance in the new block is already modelled by the data.

Dimensionality reduction

MLLL has four parameters that determine the dimensionality reduction of the feature vector. The first two are the dimensionalities of the subspaces of the landmark and background data, cf. Equations A.1 and A.2 in Appendix A. The third parameter is the dimensionality of the joint subspace

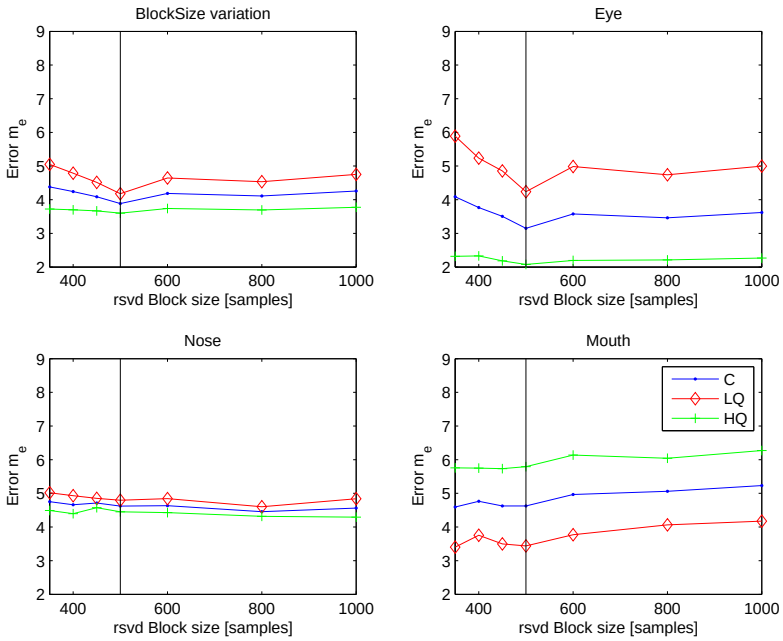


Figure 4.2: The error m_e as function of the blocksize. Block sizes smaller then 300 will not result in enough features. It is clear that below 500 features the error grows when the number of features is reduced. More features do no improve the performance. The black line indicates the chosen value.

of background and landmark data, cf. Equations A.3 to A.7 in Appendix A. Instead of these dimensionalities, we will take the amount of variance retained in the, respective, subspaces as tuning parameters. The fourth parameter is the number of most discriminating features that is selected in the final feature reduction step. For every parameter is a trade-off between speed and accuracy. The chosen setting for each of these parameters has an impact on the others. Fewer features will give faster performance but too few will make the error m_e too large. Too many features will lead to overfitting, again resulting in poor performance. The choice of these parameters are discussed in the following subsubsections. In that procedure we start with an educated guess and after that optimise the parameters one at a time, converging to a hopefully global optimum.

Explained variance landmark templates

Figure 4.4 shows that there is an optimum around 81% of kept variance, which is mainly due to a local minimum in the landmarking errors for the eyes. Errors for the eyes are the same for kept variances above 88% because

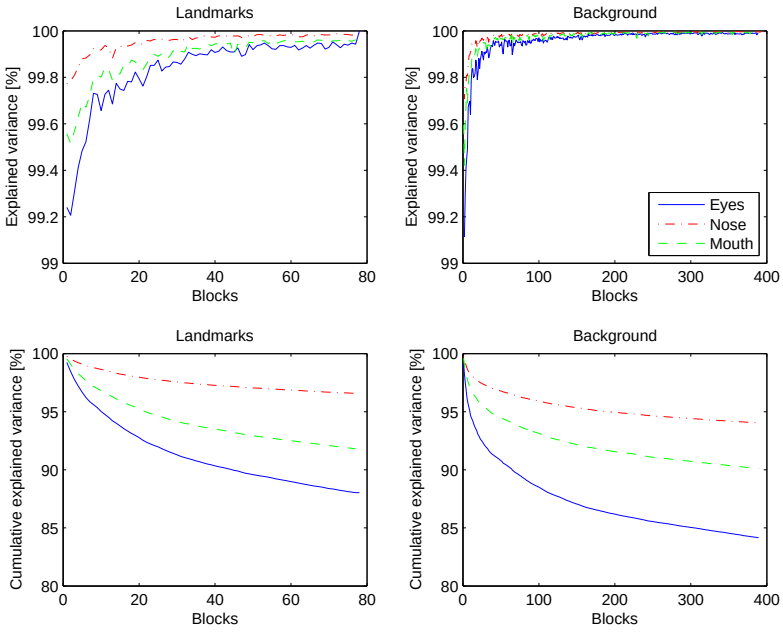


Figure 4.3: The upper two graphs show the amount of variance which is kept after each feature reduction step. This goes to 100% when the data are modelled better and better. The lower graphs show the cumulative kept variance of the total data as a function of the number of processed blocks.

the amount of kept variance due to the ARSVD is 88%.

Explained variance background templates

There is little room to vary this parameter. The total amount of kept variance after the ARSVD is 94% for the eyes and even less for nose and mouth. Keeping 94% or more of all features means de facto keeping all features. The drop off is very steep because at 94% all 500 features are kept while going below 93.5% only few features are kept. Therefore this parameters is set to 100%, keeping all features in order not to limit the choice for the number of features m in Section 4.2.2.

Combined explained variance

As we can see in Figure 4.5 the influence of the overall explained variance is a rather limited. It is, apart from noiselike fluctuations, almost flat throughout its range. Important considerations for this parameter are computational speed during training and the fact that we want to keep enough features for

Table 4.2: Amount of kept variance using a blocksize of 500 and training on all the data of the FRGC training set.

	Landmark	Background
Eye	88.0 [%]	84.2[%]
Nose	96.6 [%]	94.0[%]
Mouth	91.8 [%]	90.1[%]

the next phase to be effective. Nonetheless, we choose to tune our system to 98%, the local optimum.

Number of features during feature selection

The last feature selection step selects the number of features to be kept. As was explained in 3.2.1 the criterion here is the maximum of the quotient of the mean and the standard deviation. Figure 4.6 shows how the final selection of features enables one to find a local optimum. Not all landmarks have a clear optimum. For the eyes it is clear that around 150 features is best. For both the nose and the mouth, above a certain value the error becomes more or less constant. The value of 219 was the overall best.

Discussion

Interestingly, the m_e of 3.1 for the mouth on the LQ images is lower than the m_e of 5.8 for the HQ images. This is against the intuition that the error on HQ images should be lower. If we however calculate the errors for the full data set this effect disappears, as we would expect. The HQ error is 3.7 and the LQ error is 4.3. We, therefore, consider this to be a data anomaly.

4.2.3 BILBO

The BILBO outlier correction algorithm has four parameters to tune. The number of iterations, the minimal threshold, the weight factor and the number of features that are kept. Since the FRGC database has ground truth coordinates for four landmarks BILBO uses eight input features. In Figure 4.7 the first three modes of variation in the subspace are visualised in shape space. Experiments showed that by keeping only the first feature in the subspace the best results were obtained. The number of iterations was set to 3 because convergence was reached at that value for all the shapes in the training data. The final two parameters, the minimal threshold and the weight factor, were both optimized. The results are shown in Figure 4.8. The minimum is found for a minimal threshold, τ_{min} of 0.055 and an error weight

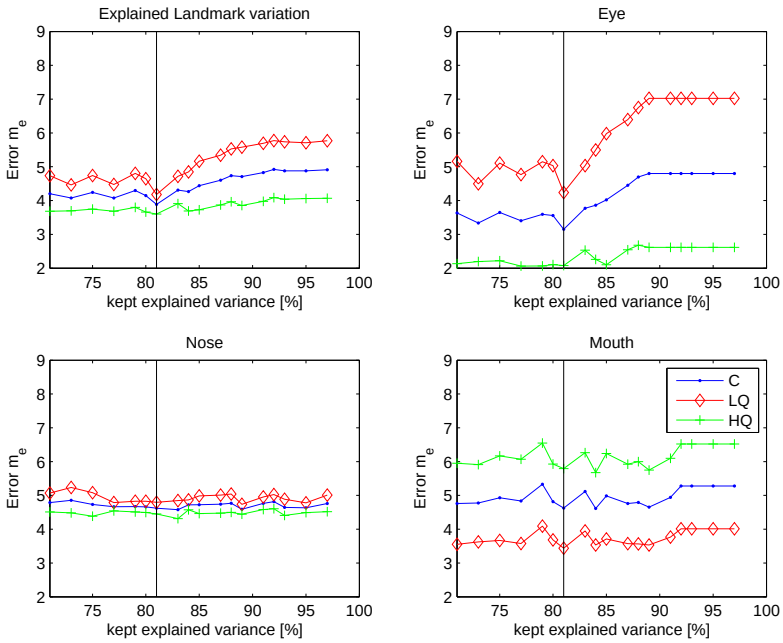


Figure 4.4: The error, m_e, as function of the amount of explained landmark variance. The black line indicates the chosen value.

c of 1.15. The mesh denotes the m_e without any outlier correction of 4.1% for reference purposes.

Examples of both correct and erroneous outlier corrections are given in Figure 4.9.

4.2.4 The Repetition Of Landmark Locating

The number of iterations determines how often we rerun the landmarker. Here that is MLLL in combination with BILBO. The choice of the number of iterations will be based on a trade off between accuracy, landmarking error and processing time. Since this parameter is linear with the total time needed we want to keep it as low as possible. In Table 4.3 it can be seen that with each iteration the error reduces, but not significantly after the 2nd iteration.

4.3 Final results

In this section the results of the landmarking experiments are presented and discussed. All tuning parameters are set to values as found in Section 4.2.2 and given in Table 4.1. In all experiments we distinguish between the high

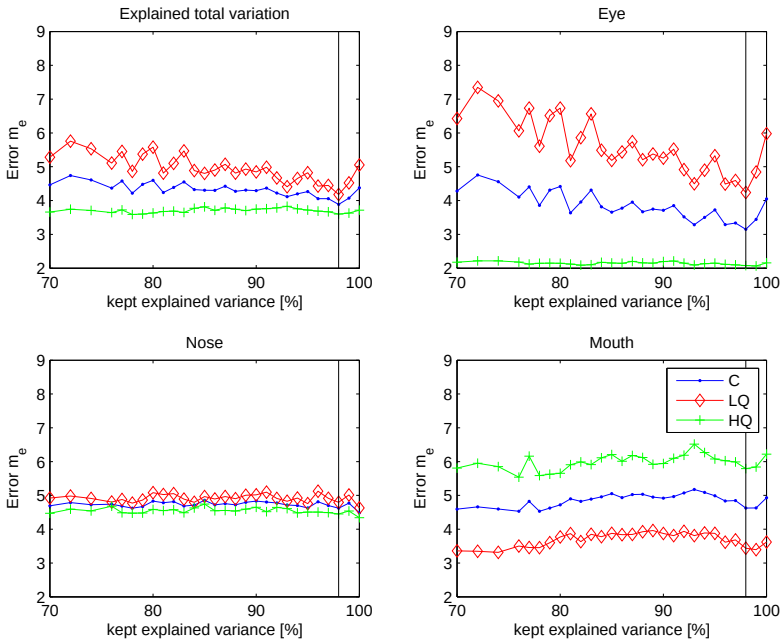


Figure 4.5: The error, m_e, as function of the amount of total or overall explained variance. The black line indicates the chosen value.

quality images (HQ), the low quality images (LQ) and the combined results (C). More information on the datasets has been given in Section 4.2.

We present the results for three combinations: MLLL, MLLL+BILBO, and TROLL, which iterates MLLL+BILBO. Also we provide the results of two reference algorithms.

4.3.1 Reference algorithms

For reference purposes we provide two basic algorithms. The first returns the a priori landmarks given the face location and size as found by the Viola and Jones face detector. It will be denoted as the a priori landmark locator. The second algorithm is the OpenCV [40] implementation of the Viola and Jones face finder, but now trained for finding landmarks on the same datasets as MLLL [62].

4.3.2 Results

The results of all experiments are given in Table 4.4. With a few exceptions it can be said that both BILBO and TROLL improve the performance of MLLL. On the eyes the Viola and Jones landmark locator performs better on the LQ

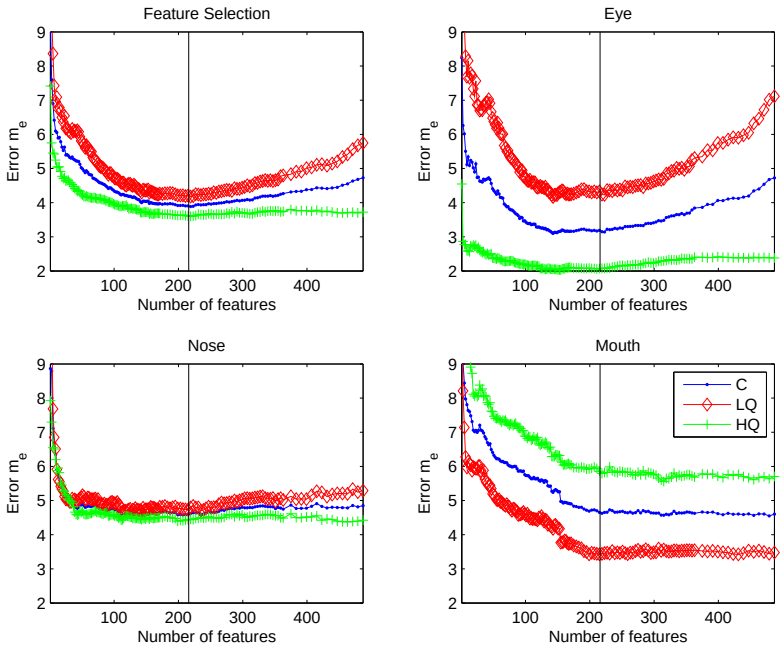


Figure 4.6: The error, m_e, as function of the amount of total or overall explained variance. The black line indicated the chosen value.

images and MLLL run on the HQ images. In general all methods perform better on the HQ images than on the LQ images. Virtually all methods perform better than the a priori landmark locator. Cumulative error plots for both the HQ and the LQ are given in Figures 4.10 and 4.11. In the latter case it can clearly be seen that for the eyes the Viola and Jones implementation outperforms all other methods, while on the mouth it lacks performance. Comparing the results for HQ and LQ shows that for the eyes the difference is large but at the same time for the nose and the mouth it is a lot smaller.

4.3.3 Discussion

MLLL

It is remarkable that for both nose and mouth there is a rather small difference between the HQ and the LQ. For the nose the LQ error is 1.2 times larger than the HQ error. For the mouth this is 1.4 times. On the contrary the eyes show a big difference with a 2.8 times larger error for the LQ data.

The weakest performance of MLLL is on the LQ eyes when trained on the FRGC training set. We suspect several causes of this. First of all, the illumination conditions which severely darken the eyes. Also the camera

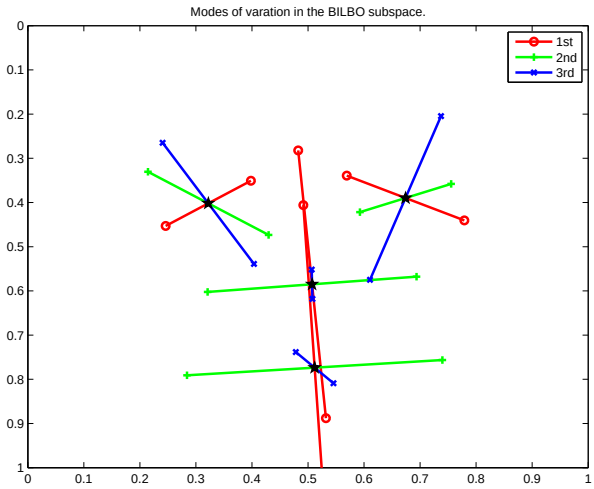


Figure 4.7: The three modes with the highest variation in the BILBO subspace.

Table 4.3: The m_e for all landmarks for five iterations. Changes beyond the second iteration are not significant. Boldface denotes the minimal value.

Landmark	1st	2nd	3rd	4th	5th
Combined	3.8	3.5	3.5	**3.4**	3.5
Eyes	3.2	3.2	**3.1**	**3.1**	**3.1**
Nose	4.5	**4.1**	**4.1**	**4.1**	**4.1**
Mouth	4.4	3.6	3.6	**3.5**	**3.5**

is sometimes out of focus. In the LQ images some people wear glasses, sometimes with a glare on it. Finally, people sometimes turn their eyes aside or close their eyes at the moment the image is taken. In Figure 4.12 some examples are shown. From these it can be seen that these causes affect the nose and mouth to a lesser degree than the eyes. This is supported by the fact that MLLL performs much better on the LQ data when trained on the BioID database, which does not contain such deteriorated samples. It is also true that for images in the testing set with the imperfections as shown in Figure 4.12, MLLL makes the worst errors. Having poor quality images in the training set apparently does not make MLLL more robust.

BILBO

The effect of BILBO can be analysed in more detail than just as the reduction of the error m_e after MLLL. In Figure 4.13 the change of the error per image are shown as the blue solid line. For illustrative purposes the errors are

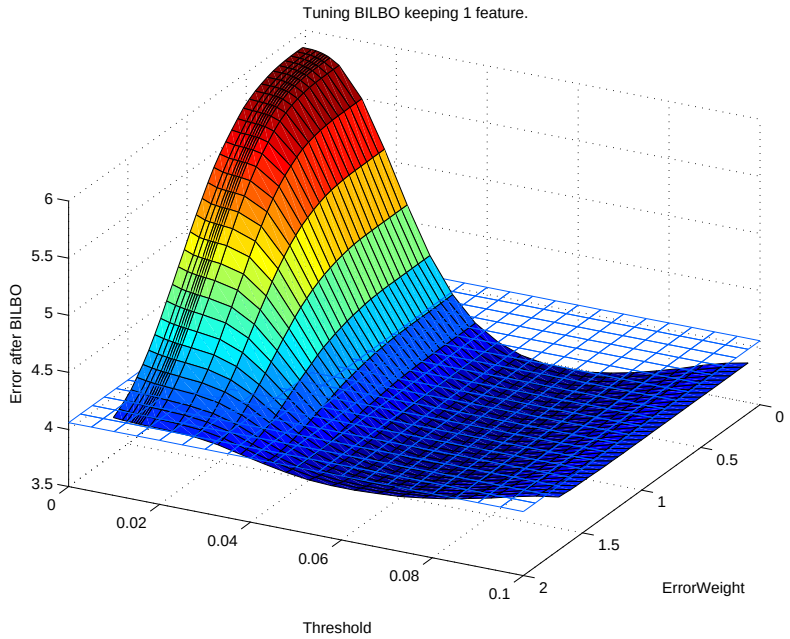


Figure 4.8: The error, m_e as function of both the minimal threshold τ_{min} and the error weight c. The surface indicates the error when using BILBO. The mesh denoted the error without applying BILBO for reference purposes.

sorted by the improvement by BILBO. On the left negative improvements represent the images where the estimates of the landmark coordinates had been deteriorated. Moving to the right it is clear that most of the images are not changed at all. Finally on the right the improvements are shown. The area between the blue solid line and the null-line is a measure for the total improvement. For the low quality images the positive improvement by BILBO is eleven times the deterioration. For the high quality images the effect is only just positive (1.3 times). The more detailed information in Table 4.4 shows that BILBO improves the results for all landmarks and datasets with the exceptions of the HQ images of the eyes when training on the FRGC training set and testing on the FRGC testing set. This is however only a very small effect.

TROLL

For the nose and the mouth TROLL yields the best results. The improvement caused by TROLL is analysed in the same way as the improvement of BILBO. This is also illustrated in Figure 4.13. Analogous to BILBO the gain is highest on the low quality images, namely 6 times. For the high quality images

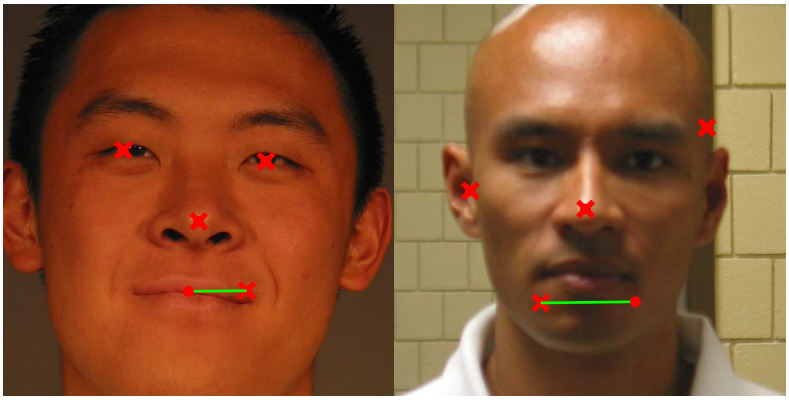


Figure 4.9: Landmark outlier correction. The crosses denote the landmark location by MLLL while the dots denote the corrected location. In the left image the successful detection and correction of an outlier is shown. The right image shows an example where the input data are so bad that BILBO is unable to do anything meaningful.

the improvement is a factor of 1.7. In contrast to BILBO there is a smooth transition from deterioration to improvement without a dead zone where the coordinates are not adjusted.

It proved that TROLL was not able to produce any intelligible results if the initial face bounding box had dimensions so that some landmarks fall outside the search areas. This would cause MLLL in the first run to give just any random position, and thus TROLL can drift away. An example is given in Figure 4.14. Because the face finder found the face on the wrong scale, the nose and mouth are not within the search regions, denoted by the red rectangles. The results of MLLL, BILBO and TROLL are thus not meaningful. In the FRGC testing set there are 810 images for which one of the landmarks is not in the search area. The impact on the overall performance is limited: it increases the error measure roughly 0.1%.

Comparison to other work

Several papers report results on eye-finders. Unfortunately the authors were not able to find any work for nose and mouth localization that could be compared on the FRGC database. Here we only focus on the ones that report results on the eyes and the FRGC for ease of comparison.

There is a difference between the shape Shape Optimised Search (SOS) by Cristinacce *et al.* and our proposed methods BILBO: SOS is an integral part of the approach and BILBO is performed as an outlier correction method after

Table 4.4: The m_e for all methods. The results for MLLL, MLLL+BILBO and TROLL are shown. As well as two reference methods.

	Combined			Eyes			Nose			Mouth		
	Training set: FRGC training set, Testing set: FRGC testing set											
	C	HQ	LQ	C	HQ	LQ	C	HQ	LQ	C	HQ	LQ
A priori	7.3	7.2	7.6	6.2	5.9	7.0	8.2	8.5	7.5	8.5	8.4	8.8
Viola Jones	4.2	3.5	5.6	**2.9**	2.4	**3.9**	4.4	3.5	6.1	6.6	5.8	8.3
MLLL	3.9	2.7	6.3	3.8	**1.9**	7.5	4.3	3.6	5.6	3.8	3.4	4.5
BILBO	3.5	2.7	5.0	3.2	**1.9**	5.4	4.1	3.6	5.0	3.6	3.3	4.3
TROLL	**3.3**	**2.5**	**4.9**	3.1	**1.9**	5.4	**3.9**	**3.4**	**4.8**	**3.3**	**2.9**	**4.0**
	Training set: FRGC training set, Testing BioID											
A priori		10.6			8.6			13.3			11.9	
Viola Jones		9.0			6.7			11.8			11.3	
MLLL		7.5			5.7			10.4			8.3	
BILBO		6.6			**5.3**			9.0			6.9	
TROLL		**6.3**			**5.3**			**8.1**			**6.6**	
	Training set: BioID, Testing set: FRGC testing set											
	C	HQ	LQ	C	HQ	LQ	C	HQ	LQ	C	HQ	LQ
A priori	8.3	8.4	8.2	7.7	7.6	7.9	8.4	8.7	**7.9**	9.3	9.6	8.9
Viola Jones	7.7	6.4	10.3	3.8	3.4	**4.7**	13.3	10.1	19.9	9.1	8.2	10.9
MLLL	6.9	6.0	8.6	3.3	2.5	4.8	12.5	13.1	11.5	8.5	6.0	13.4
BILBO	5.6	4.9	**6.9**	3.4	2.6	4.9	8.5	8.2	9.2	7.0	6.1	8.5
TROLL	**5.3**	**4.4**	7.0	**3.3**	**2.4**	4.9	**8.0**	**7.1**	9.6	**6.8**	**5.8**	**8.7**

landmarking.

Wang *et al.* [70] used Adaboost in combination with multiple weak probabilistic classifiers. Using non FRGC training data from multiple sources they report a mean Euclidian distance error on the eyes of 2.67% of the interocular distance on the FRGC 1.0 database, which is a subset of the FRGC 2.0 database. Their results can be compared to ours because they tested on the FRGC 1.0. The FRGC 2.0 database is larger but includes the FRGC 1.0 database. Wang *et al.* seem to have a similar, but slightly better result on the eyes than the Viola and Jones algorithm which has an m_e of 2.9 for the Viola and Jones method and a 3.1 for TROLL.

Campadelli *et al.* [17] used a combination of Haar classifiers and Support Vector Machines. They report a 2.65% error on the HQ data and a 3.88% error on the LQ data of the FRGC 1.0 database. These results are also similar to the ones we obtained with a Viola and Jones detector. The MLLL performs significantly better on the HQ data while on the LQ data it is worse. These results are summarised in Table 4.5.

In previous work by the authors [9] results for earlier versions of MLLL, which were not tuned nor optimized, and BILBO were given. See Table 4.6. These versions were trained on the BioID database and tested on the FRGC 1.0 database. The new results are significantly better for MLLL. For newly trained BILBO the results on the mouth and the nose yield slightly higher errors. This can be explained by the fact that BILBO used 4 landmarks

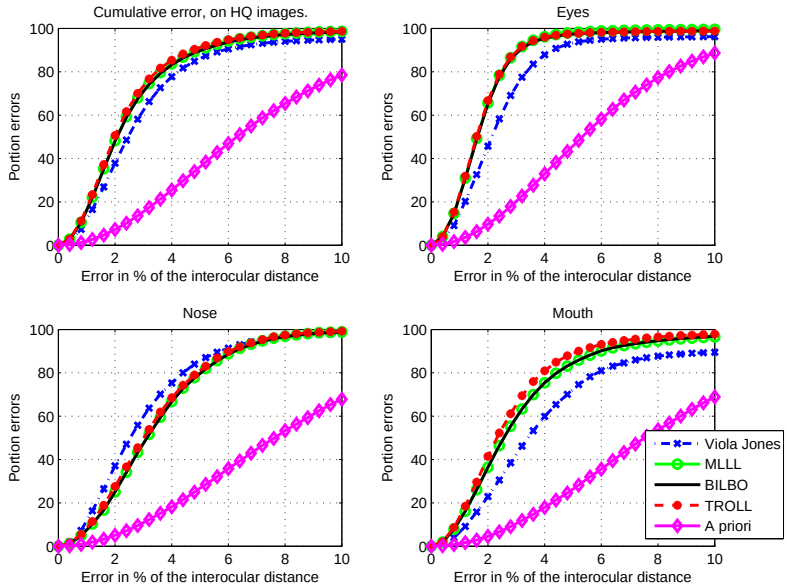


Figure 4.10: Cumulative error distribution. Landmarkers trained on the FRGC training set. Testing on HQ of the FRGC testing set.

while the 'old BILBO' in [9] used 17 and therefore could make better use of the dependency of the landmarks. Note that MLLL and BILBO were tuned using the FRGC 2.0 database. The tuned parameters were not changed when training on the BioID database. Therefore we do not have optimal performance when training on the BioID database. The numbers are given in Table 4.6. This shows that tuning can lead to significantly better result for MLLL. Also it shows that BILBO using more landmarks is useful for BILBO.

The MLLL method presented here used one set of parameters to find eyes, nose and mouth. These parameters have not been optimized for finding the eyes as was the case with the methods we used for comparison. Seeing

Table 4.5: Comparing other work on the eyes. Boldface denotes the minimum. Italics denotes an estimate not provided by the authors.

	Combined	HQ	LQ
Wang *et al.* [70]	**2.67**		
Campadelli *et al.* [17]	*2.7*	2.65	**2.88**
Viola and Jones [68]	2.9	2.4	3.9
Troll	3.1	**1.9**	5.4

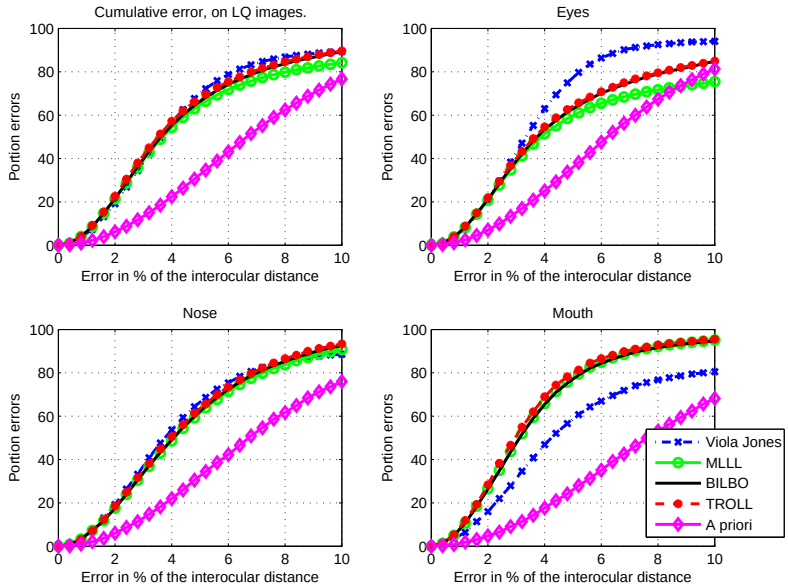


Figure 4.11: Cumulative error distribution. Landmarkers trained on the FRGC training set. Testing on LQ of the FRGC testing set.

that these specifically-for-the-eyes-trained locators perform similarly we are confident to say that our results have a good probability of performing better when tuned separately for each landmark. Finally, all methods are coming close to the accuracy of the manual landmarks. The manual groundtruth landmarks are sometimes, according to the authors, with larger error then the proposed methods. Figure 4.15 provides some examples. Here we see that the manual landmarks of the nose are not placed consistently, at least for these examples. Unfortunately the accuracy of the manual landmarks is unknown. The manual landmarks are given as natural, rounded, numbers. Locally assuming a uniform distribution for the real locations the quantisation error can be calculated to be in the order of 0.4 pixels. This corresponds to a m_e in the order of 0.2%. This is less than one tenth of the mean error and therefore not likely to significantly enlarge the errors.

Recommendations

For both training databases MLLL, BILBO and TROLL are trained using the same tuning parameters. Optimising for each landmark will surely improve the results because the current setting is probably a local optimum for minimizing for all landmarks at once. In the same fashion we treated the

Figure 4.12: Examples of LQ training samples that, for the eyes, deteriorate the landmarkers. Clockwise from the upper left we have illumination, illumination in combination with focusing on the background, looking sideways and finally glasses with glare on them. Having these in the training set does not improve the performance.

HQ and the LQ data equally. If we would have optimized MLLL for HQ and LQ and each landmark separately, the results are likely to improve.

In Section 3.2 we assumed the landmarks to be independent. This assumption is known to give a simplification of the truth. Not doing this very likely will improve the accuracy and robustness further because using this dependence in hindsight, as BILBO does, already improves the results.

4.4 Conclusions

The landmarking method presented in Chapter 3, MLLL, has been optimized. In this chapter we showed that all methods perform comparable to methods proposed by published state-of-the-art methods, even though we present

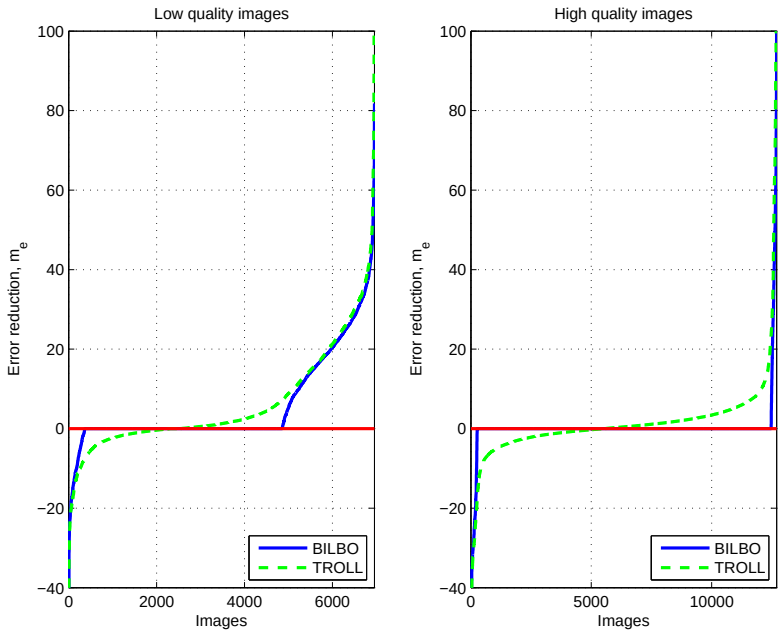


Figure 4.13: The error reduction by BILBO and TROLL, sorted by the improvement. The blue line denotes the error reduction by BILBO. The green dashed line denotes TROLL. Negative values show a deterioration of the results and positive values an improvement.

a more general implementation whereas others present a landmarker specifically for the eyes. TROLL has an overall error m_e of 3.3% of the interocular distance, which is far better than results obtained with earlier versions of MLLL. This shows that training on more data, as well as tuning the parameters, is worthwhile. BILBO also proved to be a useful tool, even if operated on only 4 landmarks. Iterative implementation of MLLL and BILBO proved to be a further improvement of the results significantly. TROLL shows the best overall performance of the presented algorithms.

It is to be expected that the results for the individual landmarks can be further improved by parameter tuning for each landmark individually. The same is true for training separately on the HQ of LQ data.

The spectral template matcher speeds up the execution of MLLL tenfold. Both the spectral template matcher and ARSVD were essential for final performance in terms of speed, accuracy and the possibility to investigate the parameter space while tuning.

Finally, because the accuracy of the manual groundtruth data the quality of current state of the art landmarkers is difficult to calculate reliably and

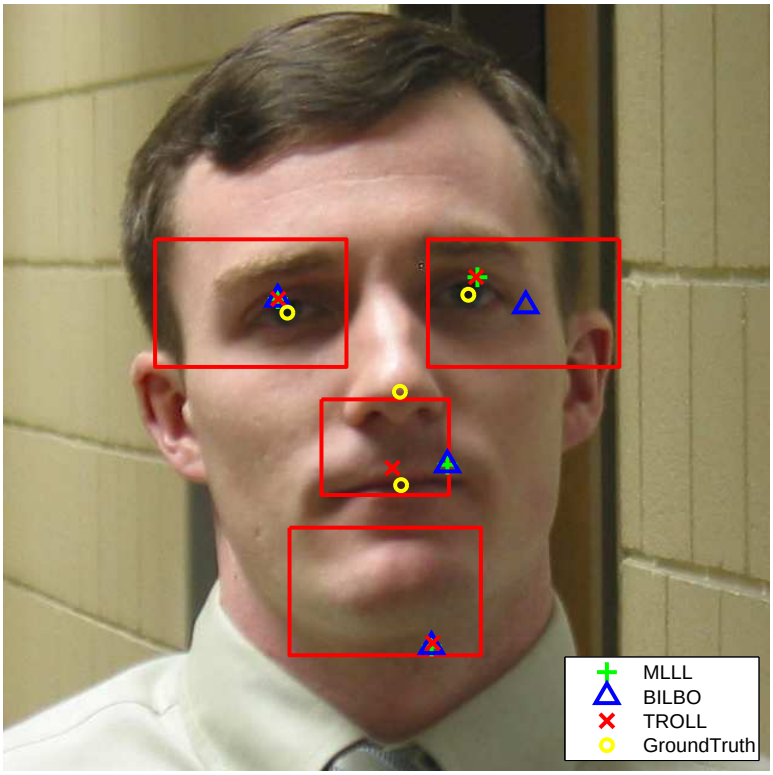


Figure 4.14: Poor performance of all algorithms because the face finder found the face on the wrong scale. The landmarks lie outside the search areas denoted by the red rectangles.

difficult to compare. Even though this might pose a problem in evaluating the quality of the landmarkers this should not limit the ambition to improve them.

Table 4.6: Comparing MLLL and BILBO to older work. Boldface denotes the minimum. Trained on the BioID database, tested on the FRGC. It should be noted that the old versions were tested on the FRGC version 1 database while the new ones were tested on our testing set of the FRGC version 2.

	Combined	Eyes	Nose	Mouth
old MLLL	10.3	6.2	17.1	7.7
new MLLL	6.9	3.3	12.5	8.5
old BILBO	6.2	5.4	8.0	**5.6**
new BILBO	5.6	3.4	8.5	7.0
TROLL	**5.3**	**3.3**	**8.0**	6.8

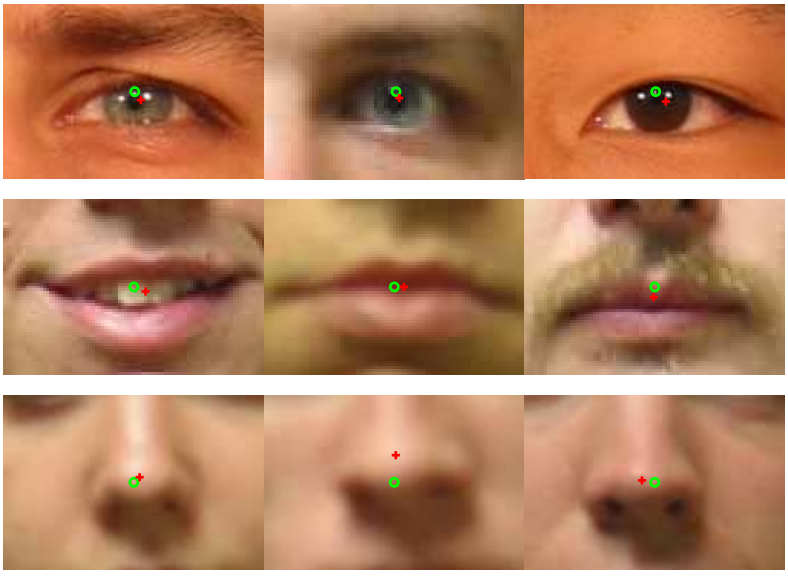


Figure 4.15: This figure provides some examples where the landmarkers MLLL, BILBO and TROLL give an equal or better estimates than the manual landmarks. The green circle denotes the manual position and the red cross denotes the position found by TROLL.

Chapter 5

Assumptions and the use of prior knowledge

This chapter is a combination of work presented in 2007 at the 28th Symposium on Information Theory in the Benelux [9] and the Biosignals 2009 conference in Porto [10].

5.1 Introduction

5.1.1 The benefits and risks of assumptions

Making assumptions in one's social interaction can lead to strange or uncomfortable situations. When one wants to give someone a gift most people try to give something of which they assume that the recipient will like it. Such assumptions will make the process of choosing a gift more efficient. However, when this assumption proves to be faulty the results can be disappointing, though not catastrophic. An example with a greater

"Piled Higher and Deeper" by Jorge Cham. www.phdcomics.com

risk can easily be thought of: a faulty assumption about a traffic light. If it is always red, until one has stopped, and always switches to green immediately, because no other traffic is near, then one might be tempted to run the red light under the assumption that no one else will be crossing. One can imagine the outcome when the assumption proves to be untrue. Assumptions, therefore, are not to be made lightly but with great care and after proper consideration. In the example of the gift it would be very useful to give something of which there is a-priori knowledge that the gift will be appreciated, for example, because it was on a wish-list.

That assumptions are not without risk applies to both real life and science. Scientists make assumptions, even ones they know to be untrue, for various reasons. Good reasons could be simplification of the problems or the statistical probability of the assumption to be true. Assumptions often enable the scientist to make the right decision with minimal effort.

Going back to Chapter 3 and the analogy between a classifier and Mount improbable, we saw that the closer a species ended up near the wolf the more probable it was a wolf. However the Tasmanian tiger also ended up very high up the mountain, even though it came through a completely different path. Just using height thus carried the danger of accidentally choosing the tiger instead of the wolf. If we would however not only look at the result but also the path the tiger took up Mount Improbable, if we used prior knowledge, then a mistake would be far less likely.

When the MLLL was designed it was mainly based on intuition. Working with the MLLL our insight grew, especially because of the good results by BILBO. We learned that the probability of shape, the collection of landmark locations, was also a factor in the equation. This led to the realisation that we implicitly had made important assumptions. A first experiment [10] confirmed that there was room for improvement. After that we improved the initial MLLL algorithm from [8] to a better one in Chapter 3 and 4. However in this chapter we will address the underlying assumption and further improvement.

5.1.2 Assumptions in MLLL

In Chapter 3 two important assumptions were made when implementing the MLLL that appear not to be fully correct. In this section we will discuss them and propose better ones. The first assumption was that the possible locations for the landmarks were assumed to be restricted to a ROI. Within this ROI the probability of the landmark location was assumed to be uniform. The second assumption was that no correlation existed between the landmark locations, ie. they were assumed independent. These two assumption, in fact, would allow facial shapes that would not be realistic, almost like a Picasso painting. Some valid faces, according to these assumptions, can be seen in Figure 5.1.

Figure 5.1: Assumption that the landmark coordinates are independent and uniform distributed over a ROI, these faces which would be equally probable.

Figure 5.2: Assumption that the landmark coordinates are dependent and Gaussian distributed, these faces seem possible.

It is obvious that these assumption could be improved because any person would think that only the lady on the left has a real face. The faces of her two nieces do not look real.

It is easy to see that the x coordinates of both eyes should have an inverted correlation, while at the same time the y coordinates would have a proportional correlation. This means that if one eye goes up, the other one also goes up. If one eye goes in or out, the other eye does the opposite, leaving the symmetry intact. In Figure 5.2 an artistic impression is shown to illustrate how the same face would look when the shape is varied in a more probable way. The image on the left is the original photograph. Her two sisters look completely different but not as weird or impossible as her nieces in Figure 5.1 do.

The faces of the nice lady, her sisters and nieces as shown in both Figure 5.1 and Figure 5.2 illustrate the differences in the assumptions between MLLL and MAP (Maximum Aposteriori Probability estimator). MAP does not assume all possible shapes to be equally probable but uses the a-priori knowledge of where the landmarks should be and which variations can be expected. We argue that this will reduce the number of false positives because they are too unlikely. Therefore MAP has a higher probability of finding the proper landmark location than MLLL. The assumptions of uniformity and independence of the shape are replaced by a priori knowledge that the landmark coordinates are mutually dependant and the assumption that the landmarks are Gaussian distributed with a certain mean and variance. It should be noted that for good face recognition it is imperative to have proper knowledge of the landmarks and the locations. At the same time it is equally important for landmarking to have a good knowledge of the face. Duin [27] already stated that the use of a priori knowledge is useful as long as the a priori features are different than the ones used for the classifier.

5.1.3 MAP

In Chapter 3 we discussed a landmarker based on a log likelihood ratio landmarker by [5], namely the Most Likely Landmark Locator (MLLL) [8]. Aiming to improve landmarking on frontal images we argue that the maximization of the likelihood ratio is a heuristic approach, lacking a formal proof that this is indeed the best position for the landmark.

We will formalize the likelihood-ratio based method as a MAP approach [65], thus giving it a solid theoretical foundation and taking the a priori probability of a landmark location into account. This will prove to render the method robust against outliers. In order to validate this approach we performed a simple experiment. We extend the MLLL to a MAP. We then evaluate all the images in the test set and see if the results are better then with only MLLL. In Section 5.4 we discuss the experiment in more detail. Results show that the new method performs significantly better than when only using the likelihood-ratio, in particular on low quality images.

In this chapter we propose two improvements on the MLLL framework from Chapter 3. As noted in Chapter 3, two assumptions on the probability distribution of the shape were made. The first was that the distributions are uniform. The second assumption is that the probabilities are independent. In this chapter we drop both. In [10] and [9] we showed that a significant improvement is made by dropping the first assumption but keeping the independence between the landmark locations. Here the last assumption, independence, also is dropped, leading to a significant improvement of landmarking. The landmark positions are not longer assumed independent. These, previously unpublished results, are presented and discussed in

Section 5.5. We will show that MAP performs better than MLLL and MLLL+BILBO. MAP performs with with equal accuracy but is more efficient than the iterative implementation, TROLL from Section 3.4. This makes MAP a clear improvement to MLLL, MLLL+BILBO and TROLL.

5.2 Theory

In this section we will present a theoretical framework for statistical landmark finding. In particular we will extend a likelihood-ratio based approach such as MLLL [8] from Chapter 3 to a MAP based approach. For clarity most of the theory is shortly repeated here. For more details on the MLLL algorithm see Chapter 3.

The shape $\vec{s}$ of a face is defined as the collection of landmark coordinates, arranged into one column vector. They belong to a face with texture $\vec{x}$, measured in a certain region of interest and also arranged into a column vector. In Equation 3.1 we calculated the maximum a posteriori estimate (MAP) [65] of the location of the landmarks, $\vec{s^*}$, given a certain texture $\vec{x}$:

$$\vec{s^*} = \underset{s}{\operatorname{argmax}}\, q(\vec{s}|\vec{x}) \tag{5.1}$$

and through the Bayes equality the probability density in Equation 5.1 becomes

$$q(\vec{s}|\vec{x}) = \frac{p(\vec{x}|\vec{s})}{p(\vec{x})} q(\vec{s}) \tag{5.2}$$

where $p(\vec{x}|\vec{s})$ is the probability density of the texture $\vec{x}$ given a shape; $p(\vec{x})$ is the background probability density; and $q(\vec{s})$ is the probability density of the shape as function of the location $\vec{s}$. The most likely shape $\vec{s^*}$ is now given by

$$\vec{s^*} = \underset{\vec{s}}{\operatorname{argmax}} \frac{p(\vec{x}|\vec{s})}{p(\vec{x})} q(\vec{s}). \tag{5.3}$$

The first factor of Equation 5.2 is the likelihood-ratio of the texture belonging to shape $\vec{s}$ over the overall texture probability. The last factor takes the probability of the shape $\vec{s}$ into account. Ideally, one would like to compute $\vec{s}$ from Equation 5.3, given all probabilities and possible shapes. This, however would be prohibitively complex.

Let $\vec{s_i} \in \mathbb{R}^2$ denote the column vector containing the spatial coordinates of landmark $i = 1 \ldots d$ and $\vec{x_i} \in \mathbb{R}^n$ the column vector containing the n pixel values from the texture in a region of interest surrounding the assumed landmark i.

We still assume the n pixels surrounding the landmark to be independent between the landmarks,

$$\vec{s^*} = \underset{\vec{s}}{\operatorname{argmax}} \prod_{i=1}^{d} \frac{p_i(\vec{x_i}|\vec{s_i})}{p_i(\vec{x_i})} q(\vec{s}). \tag{5.4}$$

The locations of the landmarks are assumed to be dependent. Equation 5.4 is, due to the argmax_s, equivalent to using the log likelihood

$$\vec{s^*} = \underset{\vec{s}}{\text{argmax}} \sum_{i=1}^{d} \Big\{ -\log\big(p_i(\vec{x_i}|\vec{s_i})\big) + \log\big(p_i(\vec{x_i})\big) \Big\} + \log\big(q(\vec{s})\big). \tag{5.5}$$

Next we assume that the probability of the texture $p(\vec{x_i}|\vec{s_i})$ has a Gaussian probability density with mean $\mu_{l,i}$ and covariance $\Sigma_{l,i}$. Likewise, we assume that $p(\vec{x_i})$ has a Gaussian probability density with mean $\mu_{b,i}$ and covariance $\Sigma_{b,i}$. Finally we assume that the shape also has a Gaussian probability density with mean $\vec{\mu}_s$ and covariance Σ_s. This is justifiable because, especially after the dimensionality reduction, the data distribution shows a Gaussian character. Even though Gaussian mixture models might model the data better, they would be much more complex, therefore, we assume Gaussian probability densities, even before the dimensionality reduction that will be discussed later.

$$p_i(\vec{x_i}|\vec{s_i}) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma_{l,i}|^{\frac{1}{2}}} e^{-\frac{1}{2}\left(\vec{x_i}(\vec{s})-\vec{\mu}_{l,i}\right)^T \Sigma_{l,i}^{-1}\left(\vec{x_i}(\vec{s})-\vec{\mu}_{l,i}\right)}, \tag{5.6}$$

$$p_i(\vec{x_i}) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma_{b,i}|^{\frac{1}{2}}} e^{-\frac{1}{2}\left(\vec{x_i}(\vec{s})-\vec{\mu}_{b,i}\right)^T \Sigma_{b,i}^{-1}\left(\vec{x_i}(\vec{s})-\vec{\mu}_{b,i}\right)} \tag{5.7}$$

and

$$q(\vec{s}) = \frac{1}{(2\pi)^{d}|\Sigma_s|^{\frac{1}{2}}} e^{-\frac{1}{2}\left(\vec{s}-\vec{\mu}_s\right)^T \Sigma_s^{-1}\left(\vec{s}-\vec{\mu}_s\right)}, \tag{5.8}$$

where n is the number of pixels in a sample.

On substitution of Equation 5.8, Equation 5.7 and Equation 5.6 in Equation 5.5 we get, ignoring the constant terms and the factor $\frac{1}{2}$, a final expression for the MAP estimate of the shape:

$$\begin{aligned} \vec{s^*} \quad = \quad & \underset{\vec{s_i}}{\text{argmax}} \sum_{i=1}^{d} \Big\{ -\big(\vec{x_i}(\vec{s_i}) - \vec{\mu}_{l,i}\big)^T \Sigma_{l,i}^{-1} \big(\vec{x_i}(\vec{s_i}) - \vec{\mu}_{l,i}\big) \\ & + \big(\vec{x_i}(\vec{s_i}) - \vec{\mu}_{b,i}\big)^T \Sigma_{b,i}^{-1} \big(\vec{x_i}(\vec{s_i}) - \vec{\mu}_{b,i}\big) \Big\} \\ & - \big(\vec{s} - \vec{\mu}_s\big)^T \Sigma_s^{-1} \big(\vec{s} - \vec{\mu}_s\big). \end{aligned} \tag{5.9}$$

5.2.1 Dimensionality reduction

Because $\vec{x_i}$ consists of a large number, n, of statistically dependent pixels it is possible and useful to perform a feature reduction. The covariance matrices, Σ_l and Σ_b, need to be estimated from training data. Due to their size, direct evaluation of Equation 5.9 would be a high computational burden. Due to

the limited number of training samples available in practice, they would be rank-deficient or, if not, too inaccurate to obtain a reliable inverse, which is needed in Equation 5.9. For example, a typical training sample consists of 6144 pixels while there are only 19674 landmark samples. Therefore, prior to evaluation of Equation 5.9, the vector $\vec{x}$ will be projected onto a lower dimensional subspace. This subspace should have several properties. First of all, its basis should contain the significant modes of variation of the landmark data. Secondly, it should contain the significant modes of variation of the background data. Finally, it should contain the difference vector between the landmark and the background means. The latter is needed for good discrimination between landmark and background data. The modes of variation are found by principal component analysis (PCA). This procedure is the same as discussed in Chapter 4. See Appendix D for details.

Finally the landmark and background densities are simultaneously whitened such that the landmark covariance matrix becomes a diagonal matrix, Λ_L, and the background covariance matrix becomes an identity matrix.

5.2.2 Feature extraction and classification

The entire process of feature reduction and simultaneous whitening can be combined to one linear transformation with a matrix $T_i \in \mathbb{R}^{n \times m}$, with n the dimensionality of the training templates and m the final number of features after reduction.

$$\vec{\mu}'_{l,i} = T_i \mu_{l,i}, \qquad \vec{\mu}'_{b,i} = T_i \mu_{b,i} \tag{5.10}$$

$$\Lambda_{l,i} = T_i \Sigma_{l,i} T_i^T, \qquad I = T_i \Sigma_{b,i} T_i^T \tag{5.11}$$

$$\vec{y}_i(s) = T_i \vec{x}(s) \tag{5.12}$$

Substituting this in Equation 5.9 gives:

$$\begin{aligned} \vec{s^*} = \underset{\vec{s}}{\operatorname{argmax}} \sum_{i=1}^{d} \Big\{ & -\big(\vec{y}_i(\vec{s}_i) - \vec{\mu}'_{l,i}\big)^T \Lambda_{l,i}^{-1} \big(\vec{y}_i(\vec{s}_i) - \vec{\mu}'_{l,i}\big) \\ & + \big(\vec{y}_i(\vec{s}_i) - \vec{\mu}'_{b,i}\big)^T \big(\vec{y}_i(\vec{s}_i) - \vec{\mu}'_{b,i}\big) \Big\} \\ & - \big(\vec{s} - \vec{\mu}_s\big)^T \Sigma_s^{-1} \big(\vec{s} - \vec{\mu}_s\big). \end{aligned} \tag{5.13}$$

Note that although Equation 5.13 resembles Equation 5.9, the numerical result will be different due to the dimensionality reduction. This form is however computationally far more efficient than Equation 5.9.

5.3 Implementation

The aim of landmarking is to find the most likely shape. The total number of possible shapes is over 10^{14}. This makes that it is impossible to evaluate Equation 5.5 for the entire shape space. We thus need a optimization algorithm that converges to a good solution in a limited number of evaluations. For the optimization algorithm we define a cost function, $m(\vec{s}, \vec{x})$:

$$\begin{aligned} m(\vec{s}, \vec{x}) \quad = \quad & \sum_{i=1}^{d} \Big\{ - \big(\vec{y}_i(\vec{s}_i) - \vec{\mu}'_{l,i}\big)^T \Lambda_{l,i}^{-1} \big(\vec{y}_i(\vec{s}_i) - \vec{\mu}'_{l,i}\big) \\ & + \big(\vec{y}_i(\vec{s}_i) - \vec{\mu}'_{b,i}\big)^T \big(\vec{y}_i(\vec{s}_i) - \vec{\mu}'_{b,i}\big) \Big\} \\ & - \big(\vec{s} - \vec{\mu}_s\big)^T \Sigma_s^{-1} \big(\vec{s} - \vec{\mu}_s\big) \end{aligned} \tag{5.14}$$

which we aim to maximize. The first terms of the summation of Equation 5.14 are the objective functions that are maximized by MLLL. We calculate these values for the entire region of interest for each landmark. These landscapes are used as lookup-tables. The mean shape, $\vec{\mu}_s$, and covariance matrix, Σ_s, are calculated from the training data. Since it is not possible to evaluate the entire parameter space, an optimization algorithm is used to find the optimal value. In our implementation we used the Matlab implementation of the Nelder-Mead Simplex (direct search) method [53].

Because the Simplex algorithm tends to get stuck in local optima, and the landscapes are slightly noisy, with many local optima, the MLLL landscape is smoothed by a Gaussian filter. In Section 5.5 the size of the filter kernel is discussed.

5.4 Experiments

In order to evaluate the performance of MAP and compare it to the methods from Chapter 4 we perform experiments similar to those described in Section 4.2. We evaluate images and estimate the position of the landmarks. This is compared to the groundtruth data which is provided with the FRGC database. Two new methods will be evaluated. First we will evaluate MAP as discussed in Section 5.2. Secondly we investigate if TROLL, as discussed in Section 3.4, works with MAP as the engine instead of MLLL. Two datasets were used. One for training and one for testing. Half of the FRGC database was used as the training set and the other half as the testing set. The selection of the datasets from the FRGC database is described in detail in Section 4.2.1. For MLLL all the settings are used as found in Section 4.2.2. The FRGC contains high quality images (HQ) and low quality images (LQ). The union of both set of images we will denote as combined (C).

5.5 Results and discussion

In Figure 5.3 and Figure 5.4 the cumulative error plots of MLLL, BILBO, TROLL and MAP are shown. They are split into two sets. The top block of 4 plots shows the cumulative errors for the high quality images and the bottom block of 4 plots for the low quality images. The upper right plot of each block of 4 plots shows the overall error. The upper left plot shows the error for the eyes. The bottom left and right plots show the graphs for the nose and mouth, respectively. It can be seen that, although the differences are small, MAP performs best, or equal to TROLL. The performance gain is most significant for the LQ images, hence where noise and distortions are most severe.

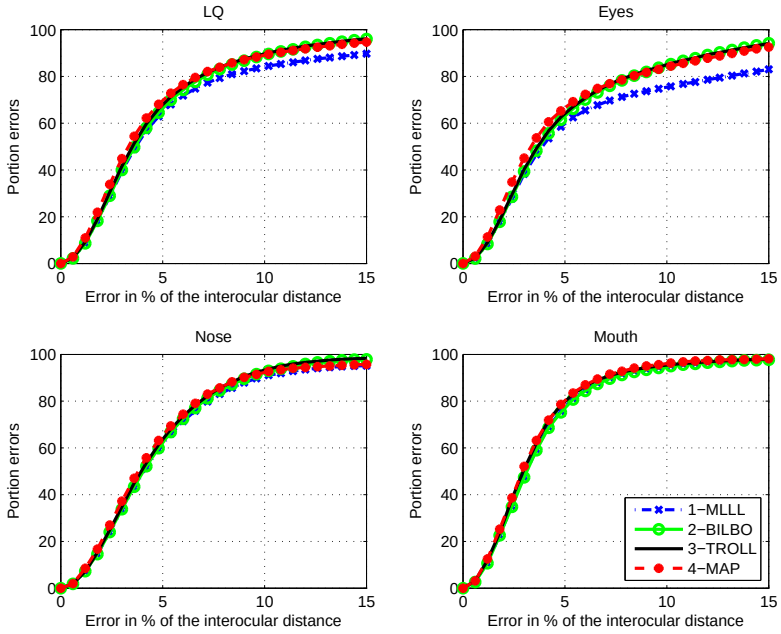


Figure 5.3: The cumulative error: high quality images.

The use of a posteriori knowledge that the landmarks are correlated and that their probability densities are better represented by a Gaussian than by a uniform probability density improve the results. Without the knowledge of relationship between the location of the landmarks, the landmarking becomes worse.

In Table 5.1 the results are given. This table shows the average RMS error per landmark. The results for MAP are compared to MLLL, BILBO and TROLL. Also the impact of the filtering on the MLLL without MAP is investigated. It can be seen that MAP performs better than MLLL and BILBO. Compared to TROLL there is no significant improvement in accuracy

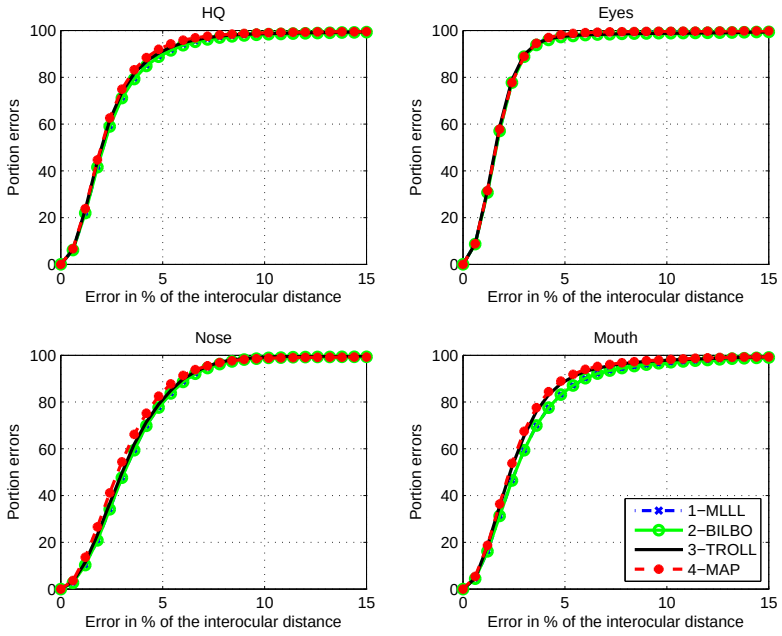


Figure 5.4: The cumulative error: low quality images.

by MAP, although there are small differences. We can also conclude that there may be a small improvement caused by the filtering but that it is smaller than in the case of BILBO, TROLL or MAP. Where BILBO and TROLL were ad-doc extensions of the MLLL algorithm, MAP is a theoretically sound implementation which performs with equal accuracy. Further improvements might be possible by modelling the data better since now it is assumed Gaussian.

A comparison between MLLL,BILBO,TROLL and the work of others is done in Section 4.3. Since MAP does not perform better than TROLL in terms of accuracy the results can be compared.

Filter As stated in Section 5.3 a low pass filter operation is needed to prevent MAP from finding local optima. We chose a Gaussian filter. Experiments showed that a kernel with a standard deviation of 3.2 pixels performs best. Small deviation from the kernel width have limited effect. This can be seen in Figure 5.5. Although there is little variation between 1.5 and 4, outside this range the error quickly explodes. For all landmarks the error without filtering is approximately 0.6 pixels higher, a big step compared to the variation as seen due to the kernel width. In Figure 5.6 the MLLL landscapes before and after filtering are shown. Many local optima in the unfiltered MLLL map can be clearly seen. After filtering the high frequency

Table 5.1: The average RMS error to the groundtruth data. Between brackets the relative improvement to MLLL is given. Boldface denotes the smallest error.

	Total	Eyes	Nose	Mouth
HQ				
MLLL	2.7	1.9	3.6	3.4
MLLL filtered	2.6 (3%)	**1.8 (4%)**	3.5 (2%)	3.3 (4%)
BILBO	2.7 (0%)	1.9 (-4%)	3.6 (1%)	3.3 (3%)
TROLL	2.5 (6%)	1.9 (-2%)	3.4 (6%)	2.9 (14%)
MAP	**2.5 (8%)**	1.8 (2%)	**3.4 (7%)**	**2.9 (15%)**
LQ				
MLLL	6.3	7.5	5.6	4.5
MLLL filtered	5.8 (8%)	6.6 (11%)	5.3 (7%)	4.5 (-1%)
BILBO	5.0 (20%)	5.4 (27%)	5.0 (11%)	4.3 (4%)
TROLL	**4.9 (22%)**	5.4 (28%)	**4.8 (14%)**	4.0 (10%)
MAP	**4.9 (22%)**	**5.2 (30%)**	5.1 (10%)	**4.0 (11%)**
C				
MLLL	3.9	3.8	4.3	3.8
MLLL filtered	3.7 (6%)	3.5 (9%)	4.1 (4%)	3.7 (2%)
BILBO	3.5 (11%)	3.2 (17%)	4.1 (6%)	3.6 (4%)
TROLL	3.3 (15%)	3.1 (19%)	**3.9 (10%)**	3.3 (13%)
MAP	**3.3 (16%)**	**3.0 (21%)**	4.0 (8%)	**3.3 (14%)**

local optima are gone, resulting in a smoother image.

TROLL with MAP TROLL in combination with MAP instead of MLLL did not yield any improvement. In Figure 5.7 is illustrated that already after the first repetition of MAP the results deteriorate. This is especially true for the LQ data, and even for the HQ data there is a small increase in error. The cause of this is twofold. First, for the majority of images TROLL does not improve the results but shows noisy behaviour around the ground truth. Secondly, for the remainder of the images the shape does not converge but at some point diverges and 'wanders off'. This is illustrated in Figure 5.8; the majority remains roughly the same while a few images grow in error. This leads to the conclusion that while with BILBO there was, on the whole, enough room for improvement to make TROLL useful. With MAP most results initially are already so good that there is no additional improvement by TROLL. At the same time, in some images the error will grow out of proportion when applying TROLL because there is no convergence. This effect was not seen in Chapter 4, where we only investigated until the fifth iteration while here the

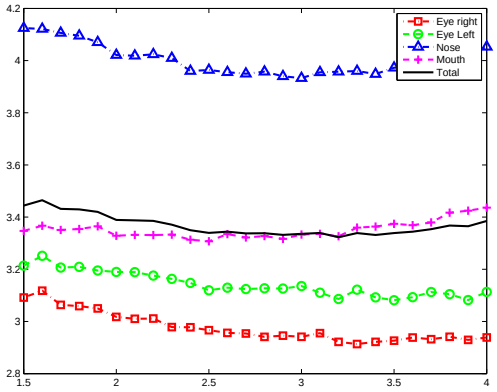


Figure 5.5: The average error calculated on 10% of the training image as function of the width of the Gaussian averaging filter kernel, in order to determine the best filter size.

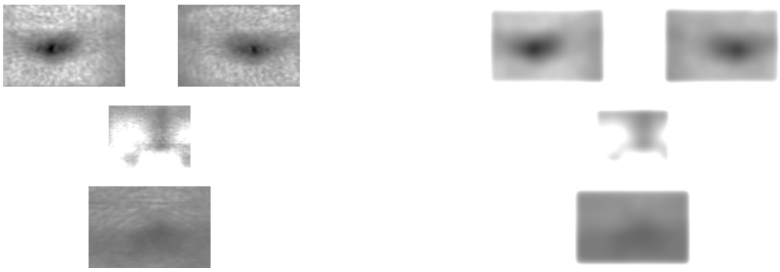


Figure 5.6: Example of an MLLL landscape (left) and the smoothed version of it (right). Black denoted high probability and white low.gmap:mlllmap

results deteriorate directly after the first repetition of MAP.

Speed For both algorithms, TROLL and MAP, calculating the MLLL landscape is by far the most time consuming step. TROLL calculates the landscape a number of times. MAP only calculates it once and uses it as a lookup table. This makes MAP faster than TROLL by almost a factor equal to the number of iterations, since there is only very little time lost to overhead. This increase is the number of iterations TROLL uses. In our implementation MLLL takes approximately 7.5 seconds to calculate and 8 including the optimization and overhead. TROLL, with three iterations, takes approximately 23 seconds to execute. Both implemented in Matlab. Threefold speed improvement without loss of accuracy is a very useful improvement.

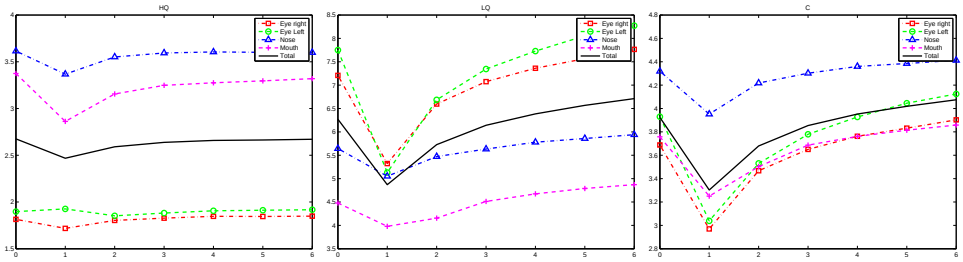


Figure 5.7: TROLL in combination with MAP instead of MLLL and BILBO. The zero-th iteration denotes MLLL, 1 the first time MAP is calculated, 2 the second time and so on. From left to right: HQ, LQ and C images.

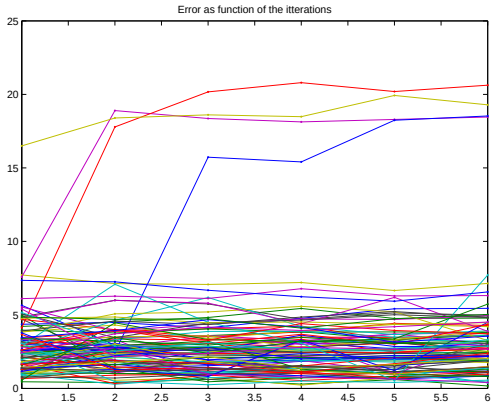


Figure 5.8: Errors for the first 100 individual images as function of the number of iterations.

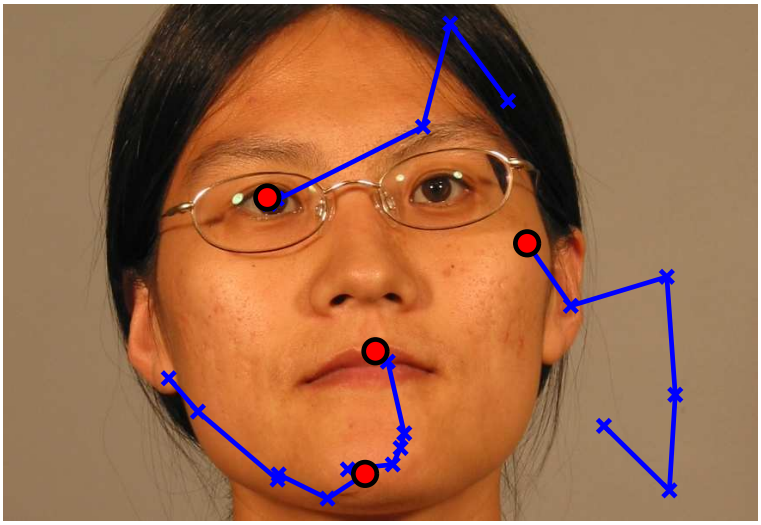


Figure 5.9: Example of TROLL exploding after a bad initial guess due to a face finder result on the wrong scale.

5.6 Conclusions

In this chapter we formulated a solid Maximum A Posteriori (MAP) framework for finding the landmarks in a facial image. Using the likelihood ratio of a location as well as the a priori probability of a landmark location the Maximum Likelihood Landmark Locater (MLLL) was expanded to MAP. We performed landmarking experiments on many images. We also applied The Repetition Of Landmark Locating (TROLL) in combination with MAP, assuming that better registration of the face prior to landmarking would lead to improvements as was the case with MLLL as shown in Chapter 4.

The results show that using MAP actually improves the performance of the MLLL algorithm on frontal still images. MAP has turned out to be more robust because the performance on the low quality images improved a lot, narrowing the performance gap with the high quality images. MAP seems to perform better than an iterative implementation of a maximum likelihood ration classifier, TROLL using MLLL. The difference is too small to be significant. Applying MAP in combination with an iterative method, TROLL using MAP, did not perform better than the MAP approach by itself. However, MAP is more efficient than TROLL, resulting in a faster landmarker.

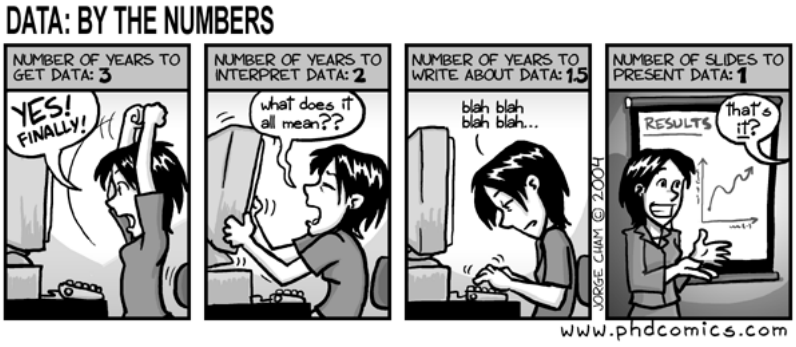
From this we conclude that, as we expected, the assumption we made that the landmark locations are independent and uncorrelated is incorrect. Replacing these assumptions with a priori knowledge of the distribution

of the landmarks, results in similar performance in term of accuracy but in better performance in terms of efficiency, compared to MLLL and TROLL.

Chapter 6

Conclusions and recommendations

The context of our research was face recognition in the home environment. The importance of registration for dealing with the variability encountered has been investigated. As a result of this study, the role of landmarking methods as a cornerstone of accurate registration methods and their underlying dependencies has been confirmed. We proposed the Most Likely Landmark Locator (MLLL) as a robust landmarker which distinguishes a landmark from a background. Two improvements are proposed: BILBO and The Repetition Of Landmark Location (TROLL). BILBO improves the landmark locations by using dependencies between landmark locations. TROLL improves the results by repeating landmarking on registered images and thus presenting face which fit the statistical models better. Finally all the work on landmarks converged to a Maximum A Posteriori landmarker (MAP). In Section 6.1 we discuss the conclusions and in Section 6.2 we give recommendations for improvements and future work.



"Piled Higher and Deeper" by Jorge Cham. www.phdcomics.com

6.1 Conclusions

6.1.1 Answers to the research questions

What is the relation between landmarking accuracy and face recognition performance?

The relationship between the quality of landmarks used for registration and the accuracy of recognition has been investigated in Chapter 2. Noise was added to the ground truth landmarks before the classifier was trained. We applied both rigid and deformable registration methods based on a set of landmarks. We used the equal error rate (EER) of a verification experiment as a measure for the quality of registration. A lower error corresponds with better registration. Two important relations were found. First, using a higher number of landmarks leads to better registration. Secondly, landmarks with higher accuracy also lead to better registration. This means that when using an automated face finder for an automatic face recognition system it is important to use as many landmarks as possible that are as accurate as possible, for good registration.

Can a statistical classifier approach be used for landmark detection?

MLLL is based upon a maximum a posteriori theoretical framework. MLLL determines the likelihood ratio of two probability densities. Namely, the probability density that a location is a landmark, divided by the probability density that a location is not a landmark. In Chapter 4 we show that the MLLL has a performance that is equivalent to methods proposed by others, even though we present a more general implementation whereas others mostly present landmarkers specific for the eyes. MLLL has shown to perform well on up to 17 different facial landmarks. This number was based upon availability of training data and there is no reason not to use more landmarks. We showed that by tuning the many parameters the performance can be boosted significantly compared to initial educated guesses and, that a classifier based on Simplified Bayesian theory is still competitive.

Can the underlying statistical relationship between landmark locations be used to improve landmarking?

BILBO In Chapter 3 we proposed BILBO, a post landmarking, subspace based, outlier correction algorithm, which uses the underlying relationship between multiple landmarks in the face. The results from a landmarker can be substantially improved by BILBO. We found in Chapter 4 that applying BILBO to the shape found by MLLL, improves the accuracy of the landmarks

significantly. Although BILBO works better on shapes with many landmarks, useful results have been obtained using shapes with only 4 landmarks.

MAP In Chapter 5 we formulated a solid MAP frame work for finding the landmarks in a facial image. By using the likelihood ratio of a location as well as the a priori probability of a landmark location the MLLL was expanded to MAP. The results show that using MAP improves the performance of the MLLL. MAP has turned out to be more robust, because the performance on the low quality images improved significantly more than on high quality images. This narrowed the performance gap between low quality and high quality images. MAP is more efficient than TROLL, thus faster, while at the same time it yields equal accuracy. We conclude that landmark locations show a strong dependence and correlation. Using a priori knowledge of the distribution of the landmarks, results in similar performance in terms of accuracy but in better performance in terms of efficiency, compared to MLLL, BILBO and TROLL. We expect that MAP will, just as BILBO, perform better when the number of landmarks is expanded. The fact that MAP is a robust and generic landmarker for cases with multiple landmark candidates present inside a Region Of Interest (ROI) makes it a valuable addition to the spectrum of landmarking tools.

Which methods can be used to reduce computational complexity and thus also overcome the computational problems which arise from very large training sets?

ARSVD We learned that using more training data improves the results. If computational limitations exist for using large quantities of training data, recursive algorithms, such as Approximate Recursive Singular Value Decomposition (ARSVD), can be useful.

Frequency domain implementation. Another implementation to speed-up landmarking is a spectral template matcher. In Chapter 3 this is explained in detail. It speeds up the execution of MLLL tenfold.

Both ARSVD and the frequency domain implementation, are essential for performance in terms of speed, accuracy and the possibility to investigate the parameter space while tuning.

6.1.2 Additional conclusions

TROLL TROLL is an iterative implementation of a landmarker. It uses the results of a landmarker to register the face and then again find landmarks in a registered image until convergence is reached. We applied TROLL in

combination with MLLL and BILBO. We show in Chapter 4 that TROLL improves the results significantly in combination with MLLL.

General Although the landmarking methods are general and regardless of the landmark it is trained to locate, the tuning is very specific. The focus of this thesis is landmarking on facial images. The algorithms can be applied to many landmark versus background classification problems in images. MAP as a method for landmarking and registration is not limited to the face.

Finally we can conclude that with our proposed landmarking methods and available methods for registration, feature reduction and classification, face recognition in the home environment is a significant step closer to reality. By improving landmarking methods we will be able to deal with much of the variability encountered in the home environment.

6.2 Recommendations

Distance from subspace MLLL uses feature reduction for computational reasons for creating maximum separation between classes. Every image tested is projected onto this subspace. It is however so that even images that are not part of the landmarks or background sets are projected onto the subspace. Those images can end up close to the target. In order to avoid misclassification, we can use the same fundamental trick as BILBO: determine the distance between the sample and its projection onto the subspace. The difference is a measure of how good the image fits the subspace, which can be incorporated into MLLL.

Tuning MAP and MLLL are tuned and optimized manually. Because of the large parameter space and limited time available this tuning was done rather with crude heuristics and educated guesses. We are convinced that, although we gave it our best effort, we are not at the optimal setting. A possible solution is to use an automated optimization algorithm to find the optimal settings. Also, as stated in Chapter 4, we tuned all MLLL parameters collectively for all landmarks. Landmarkers for the eyes, nose and mouth use the same settings for most parameters. Tuning each landmark individually will improve the performance of each landmarker and thus MAP.

Evaluation criteria In order to evaluate the algorithm we used the manually labelled ground truth data. As stated in Chapter 5 we suspect that we are getting close to accuracy of the manually labelled landmarks on the FRGC database. To confirm this it would be wise to estimate the amount of noise on the landmarks. If contemporary landmarking methods approach

manual landmarking quality, the distance between the found landmarks and the manual landmarks is not a valid error measure. Possibly, the outcome of a recognition experiment, can indicate the quality of landmarking reliably.

More landmarks The current implementation of MAP is trained on four landmarks, as found in the FRGC database. There are other databases offering other landmarks. We propose to combine these databases by more training landmarkers and combining them with MAP. Using bootstrapping methods we can refine these results and retrain all the landmarkers on the data of one or all databases combined. We expect that adding more landmarks to MAP will make MAP more robust and more accurate.

Appendices

Appendix A

MLLL

Here we briefly list the steps in the algorithms for the dimensionality reduction and the whitening of the data.

A.1 Dimensionality reduction

The subspace should contain a good representation of both the landmark data, X_l, and the background data, X_b.

i. Create the data matrices X_l and X_b where each column is a single training sample $\vec{x}(\vec{s})$.

ii. Calculate a basis of both landmark and background data:

$$U_{[l,b]} S_{[l,b]} V_{[l,b]}^T = (X_{[l,b]} - \mathrm{M}_{[l,b]}), \tag{A.1}$$

where $\mathrm{M}_{[l,b]} = \vec{\mu}_{[l,b]}[1 \ldots 1]$, i.e. a matrix whose columns are the column average of X. The subscript $[l, b]$ denotes that it applies to both the landmark and background data.

iii. For computational reasons only the first columns of U_b and U_l, which contain a fixed amount of the variance are kept.

$$\hat{U}_{[l,b]} = [\vec{u}_{[l,b],1} \vec{u}_{[l,b],2} \ldots \vec{u}_{[l,b],n_l}], \tag{A.2}$$

where n_l and n_b denote the number of columns kept. Note that $\hat{U}_l$ and $\hat{U}_b$ are not mutually orthogonal.

iv. The orthonormal basis should also contain the difference vector between both means. Therefore we estimate the normalised average landmark projection $\vec{u}_l$. This is the difference between the two landmark means, normalised to unity length.

$$\vec{u}_{lb} = \frac{\vec{\mu}_l - \vec{\mu}_b}{|\vec{\mu}_l - \vec{\mu}_b|}. \tag{A.3}$$

v. Transform the combined matrix $[\hat{U}_l\ \hat{U}_b]$ so that it is orthogonal to u_{lb}.

$$U_{lb} = (I - \vec{u}_{lb}\vec{u}_{lb}^T)[\hat{U}_l\ \hat{U}_b]. \tag{A.4}$$

vi. Make U'_{lb} an orthonormal basis of U_{lb}

$$U'_{lb}SV^T = U_{lb}. \tag{A.5}$$

vii. The final basis is given by

$$U = [\vec{u}_{lb}\ U'_{lb}]. \tag{A.6}$$

viii. For the third time reduce the number of features:

$$\hat{U} = [\vec{u}_1\vec{u}_2 \dots \vec{u}_j]. \tag{A.7}$$

ix. Project the data onto the subspace

$$X'_{[l,b]} = \hat{U}^T(X_{[l,b]} - M_b). \tag{A.8}$$

A.2 Whitening the data

Whitening the data is done so that both the covariance matrices are diagonal and the background data are unity in variance. This later enables straight forward computation of Equation 3.5 or its final implementation Equation 3.9.

i. It follows from Equation A.8 that the mean of X'_b, M'_b is zero. Perform an SVD on X'_b:

$$U_w S_w V_w^T = X'_b. \tag{A.9}$$

ii. Transform the data so that the background variance is unity:

$$X''_{[l,b]} = \frac{S_w^{-1}U_w^T}{\sqrt{n_b}}X'_{[l,b]}. \tag{A.10}$$

where S_w and U_w follow from the SVD in Equation A.9. After this tranform the backgound covariance matrix is (approximately) unity.

iii. Diagonalise the landmark covariance. The background covariance matrix remains unity. Perform an SVD on the transformed landmark data X''_l:

$$U_d S_d V_d^T = X''_l - \frac{S_w^{-1}U_w^T\hat{U}^T}{\sqrt{n_b}}(\mathrm{M}_l - \mathrm{M}_b). \tag{A.11}$$

iv. This results in a projection matrix U_d. The transformation from the original image space to the subspace, which renders the background covariance matrix (aproximately) unity and (approximately) diagonalizes the landmark covariance matrix, is now defined as:

$$T = \frac{U_d^T S_w^{-1} U_w^T \hat{U}^T}{\sqrt{n_b}}. \tag{A.12}$$

Appendix B

BILBO

B.1 Training

BILBO is trained on a set of shapes, taken from the groundtruth data, arranged as the columns of a matrix S. The training consists of the following steps:

i. All shapes normalised in scale so that the region where the VJ face finder found the face is between 0 and 1. Using this method we model the real distributions of the data. All coordinates in S are thus between 0 and 1.

ii. Perform a singular value decomposition $(S - \vec{\mu}_s) = BWV^T$, with $\vec{\mu}_s$ the mean shape.

iii. Reduce the dimensionality of the subspace by taking only the first $n < 2d$ columns of B.

B.2 Algorithm

To correct a shape the following algorithm is used:

i. Estimate the shape after transformation, $\vec{s} = BB^T\hat{s}$.

ii. Determine the Euclidean distance $|\vec{\epsilon}_i|$ per landmark between $\vec{s}$ and $\vec{s}\,'$.

iii. Determine the threshold

$$\tau = rc\frac{1}{d}\sum_{i=1}^{d}|\vec{\epsilon}_i|, \tag{B.1}$$

with c a constant and r the iteration number. Do not choose τ smaller then a predetermined threshold.

iv. For the landmarks of which $|\vec{\epsilon_i}| > \tau$, replace in $\vec{s}\,'$ by the corresponding coordinates from $\vec{s}$: $\vec{s_i}\,' = \vec{s_i} \forall i \big|\ |\vec{\epsilon_i}| > \tau$.

v. Repeat steps i to iv. Once for a landmark $|\vec{\epsilon_i}| < \tau$ stop updating it. Continue until all landmarks satisfy $|\vec{\epsilon_i}| < \tau$. Keep track of the coordinates which are allowed to change (update $\vec{i}$).

vi. Repeat step i to v changing all coordinates until stable or $r = 5$. Allow all landmark coordinates to update (reset $\vec{i}$).

vii. Transform the coordinates back to the original scale.

In Figure 3.7 a schematic overview of the shape correction algorithm is shown.

Appendix C

Complexity

C.1 MLLL

Consider a ROI containing n pixels. The number of operations per DFFT2 is then $O(n \log_2(n))$. After feature reduction the number of features is m. The number of DFFT2s to be computed is $m + 1$, as can be seen in Figure 3.6. Computing the likelihood ratio after feature computation, Equation 3.9, at every pixel location has a complexity of $O(5mn)$. Number of operations per ROI for finding the maximum value is $O(n)$. This makes the total number of operations per ROI:

$$(m + 1)O(n \log_2(n)) + nO(5m) + O(n). \tag{C.1}$$

Dividing by n gives the number of operations per pixel in ROI:

$$O(m(\log_2(n) + 6)). \tag{C.2}$$

The large ROIs used are 256×256 pixels, which means that $n = 25088$. We used $m = 219$ features. Equation C.2 results in a complexity of $O(5000)$ operations per pixel in the ROI.

C.2 Viola and Jones

The complexity of the Viola and Jones algortihm depends on the numbers of scales S, cascades C, and features K. Estimates for these numbers are taken from [62]; $S = 11$, $C = 15$, $K = 30$, on average. The total number of operations per pixel in the ROI are upperbounded by $O(S \times C \times K) \approx O(5000)$.

Appendix D

Dimensionality Reduction

The subspace should contain a good representation of both the landmark data, X_l, and the background data, X_b. Each column of data matrices X_l and X_b is a single training sample x_s. Therefore the two projections are

$$U_{(l,b)f} S_{l,b} V_{l,b}^T = (X_{l,b} - \mathrm{M}_{l,b}) \tag{D.1}$$

where $\mathrm{M}_{l,b} = \mu_{l,b}[1 \ldots 1]$, i.e. a matrix whose columns are the column average of X. For computational reasons we only keep a certain amount of data. Only the first columns which contain a fixed amount of the variance are kept. Here that is 90% of the landmark variance and 98% of the background variance. How many columns are kept varies per landmark. So U_l and U_b are not mutually orthogonal and may have possible overlap between the both of them.

The basis should also contain the difference vector between both means. Therefore estimate the normalised average landmark projection u_l. This is the difference between the two landmark means, normalised to unity length.

$$u_{lb} = \frac{\mu_l - \mu_b}{|\mu_l - \mu_b|}. \tag{D.2}$$

Next, we transform the combined matrix $[U_l \; U_b]$ so that it is orthogonal to u_{lb}:

$$U_{lb} = (I - u_{lb} u_{lb}^T)[U_l \; U_b] \tag{D.3}$$

and turn U_{lb} into an orthonormal basis of U_{lb}:

$$U'_{lb} S V^T = U_{lb}. \tag{D.4}$$

The final basis is given by

$$U = [u_{lb} \; U'_{lb}]. \tag{D.5}$$

Again we now reduce the number of features to n by keeping only the first n columns of U with the relevant information. Here that is 98% for the variance.

Bibliography

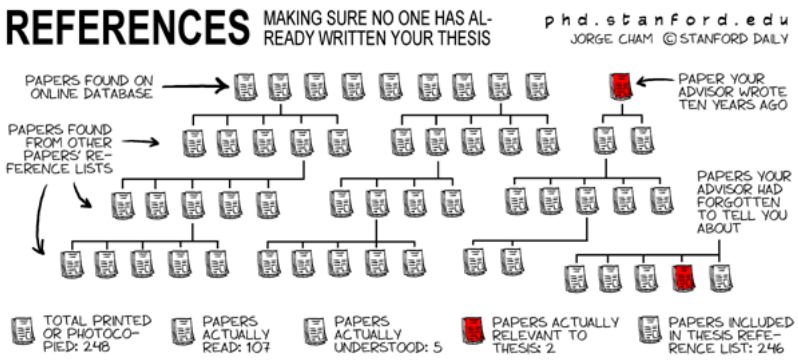
[1] *Longman dictionary of contemporary English*. Longman, Harlow, England, 3rd ed. edition, 1995.

[2] A. Albrecht, M. Behrens, T. Mansfield, W. Mcmeechan, M. Rejman-Greene, M. Savastano, P. Statham, C. Schmidt, B. Schouten, and M. Walsh. Biovision: Roadmap for biometrics in Europe to 2010, 2003.

[3] O. Ambekar, E. J. Pauwels, and B. A. M. Schouten. Binding Low Level Features To Support Opportunistic Person Identification. In *Proceedings of the 28th Symposium on Information Theory*, pages 1 – 10. Werkgemeenschap voor Informatie- en Communicatietheorie, May 2007.

[4] A. M. Bazen and R. N. J. Veldhuis. Detection of cores in fingerprints with improved dimension reduction. In *Proc. SPS 2004*, pages 41–44, Hilvarenbeek, The Netherlands, apr 2004.

[5] A. M. Bazen, R. N. J. Veldhuis, and G. H. Croonen. Likelihood ratio-based detection of facial features. In *Proc. ProRISC 2003, 14th*



"Piled Higher and Deeper" by Jorge Cham. www.phdcomics.com

Annual Workshop on Circuits, Systems and Signal Processing, pages 323–329, Veldhoven, The Netherlands, nov 2003.

[6] P. N. Belhumeur, J. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. In *ECCV 2*, 1996.

[7] G. M. Beumer, A. M. Bazen, and R. N. J. Veldhuis. On the accuracy of EERs in face recognition and the importance of reliable registration. In *5th IEEE Benelux Signal Processing Symposium (SPS-2005), Antwerp, Belgium*, pages 85–88, secretariaat in Delft, April 2005. IEEE Benelux Signal Processing Chapter.

[8] G. M. Beumer, Q. Tao, A. M. Bazen, and R. N. J. Veldhuis. A landmark paper in face recognition. In *Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on, Southampton, UK*, Los Alamitos, April 2006. IEEE Computer Society Press.

[9] G. M. Beumer and R. N. J. Veldhuis. A map approach to landmarking. In *Proceedings of the 28th Symposium on Information Theory in the Benelux*, pages 183–187, Enschede, The Netherlands, May 24/25 2007.

[10] G. M. Beumer and R. N. J. Veldhuis. Mapping landmarks onto the face. In *Biosignals 2009*. Springer, January 2009.

[11] G. M. Beumer and R. N. J. Veldhuis. A practical subspace approach to landmarking. *Journal of Multimedia*, To appear.

[12] G. M. Beumer, R. N. J. Veldhuis, and A. M. Bazen. Transparent face recognition in the home environment. In *15th Annual Workshop on Circuits, Systems and Signal Processing (ProRISC), Veldhoven, The Netherlands*, pages 225–229, Utrecht, The Netherlands, November 2004. STW Technology Foundation.

[13] F. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Trans. PAMI*, 11(6):567–585, jun 1989.

[14] B. J. Boom, G. M. Beumer, L. J. Spreeuwers, and R. N. J. Veldhuis. The effect of image resolution on the performance of a face recognition system. In *Proceedings of the Ninth International Conference on Control, Automation, Robotics and Vision (ICARCV), Singapore, Malaysia*, pages 409–414, December 2006.

[15] B. J. Boom, G. M. Beumer, L. J. Spreeuwers, and R. N. J. Veldhuis. Matching score based face recognition. In *Proceedings of ProRISC the 17th Annual Workshop on Circuits, Systems and Signal Processing, Veldhoven, The Netherlands*, pages 1–4, November 2006.

[16] A. E. Brehm. *Brehms Thierleben : allgemeine Kunde des Thierreichs / von A.E. Brehm*. Leipzig, Germany : Bibliographisches Institut, 1876-1879.

[17] P. Campadelli, R. Lanzarotti, and G. Lipori. Precise eye localization through a general-to-specific model definition. In *British Machine Vision Conference (BMVC), Edinburgh, UK, 2006.*, pages 187–196. BMVA, 2006.

[18] M. Castrillón-Santana, O. Déniz-Suárez, L. Antón-Canalís, and J. Lorenzo-Navarro. Face and facial feature detection evaluation. In *3rd International Conference on Computer Vision Theory and Applications (VISAPP)*, 2008.

[19] L. Chen, L. Zhang, L. Zhu, M. Li, and H. Zhang. A novel facial feature localization method using probabilistic-like output. In *Asian Conference on Computer Vision*, pages 1–10, 2004.

[20] T. Cootes, D. Cooper, C. Taylor, and J. Graham. Active shape models - their training and application. In *Computer Vision and Image Understanding 61*, volume 1, pages 38–59, 1995.

[21] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *IEEE Trans. Pattern Analysis Machine Intelligence*, 23(6):681–685, jun 2001.

[22] D. Cristinacce and T. Cootes. A comparison of shape constrained facial feature detectors. In 6^{th} *International Conference on Automatic Face and Gesture Recognition 2004, Seoul, Korea*, pages 375–380, 2004.

[23] D. Cristinacce, T. Cootes, and I. Scott. A multi-stage approach to facial feature detection. In 15^{th} *British Machine Vision Conference, London, England*, pages 277–286, 2004.

[24] D. Cristinacce and T. F. Cootes. Facial feature detection and tracking with automatic template selection. In *FGR '06: Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition (FGR06)*, pages 429–434, Washington, DC, USA, 2006. IEEE Computer Society.

[25] R. Dawkins. *Climbing mount improbable*. Norton, 1996.

[26] P. A. Devijver and J. Kittler. *Pattern Recognition: A Statistical Approach*. Prentice Hall, 1982.

[27] R. Duin. *On the accuracy of statistical pattern recognizers*. PhD thesis, Technische Hogeschool Delft, 1978.

[28] M. Everingham and A. Zisserman. Regression and classification approaches to eye localization in face images. In *FGR '06: Proceedings of*

the 7th International Conference on Automatic Face and Gesture Recognition (FGR06), pages 441–448, Washington, DC, USA, 2006. IEEE Computer Society.

[29] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[30] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936.

[31] K. Fukunaga. *Introduction to statistical pattern recognition (2nd ed.)*. Academic Press Professional, Inc., San Diego, CA, USA, 1990.

[32] R. C. Gonzales and P. Wintz. *Digital Image Processing*. Addison-Wesley, Reading, MA, 1977.

[33] B. Goossens, A. Pizurica, and W. Philips. Noise removal from images by projecting onto bases of principal components. In J. Blanc-Talon, W. Philips, D. Popescu, and P. Scheunders, editors, *ACIVS*, volume 4678 of *Lecture Notes in Computer Science*, pages 190–199. Springer, 2007.

[34] M. Grgic and K. Delac. Face recognition homepage. `http://www.face-rec.org/`.

[35] HumanScan. Bioid face db. `http://www.humanscan.de/`.

[36] T. Ignatenko, G. Schrijen, B. Skoric, P. Tuyls, and F. Willems. Estimating the secrecy rate of physical uncloneable functions with the context-tree weighting method. In *Proc. of 2006 IEEE Int. Symp. Information Theory, July 9-14 2006, Seattle, WA, USA*, pages 499–503, 2006.

[37] T. Ignatenko and F. Willems. On the security of the xor-method in biometric authentication systems. In *Proc. of 27th Symp. on Information Theory in the Benelux, June 8-9 2006, Noordwijk, The Netherlands*, pages 197–204, 2006.

[38] T. Ignatenko and F. Willems. On privacy in secure biometric authentication systems. In *Proc. of 2007 IEEE Int. Conf. on Acoustics, Speech and Signal Processing, April 15-20 2007, Honolulu, HI, USA*, volume 2, pages 121–124, 2007.

[39] T. Ignatenko and F. Willems. Privacy leakage in biometric secrecy systems. In *Proc. of Forty-Sixth Annual Allerton Conference on Communication, Control, and Computing, September 23-26 +2008,Monticello, IL, USA*, 2008.

[40] Intel. Open computer vision library. http://sourceforge.net/projects/opencvlibrary/.

[41] A. K. Jain, R. Bolle, and S. Pankanti, editors. *Biometrics: Personal Identification in Networked Society (The International Series in Engineering and Computer Science)*. Springer, 1st edition, January 1999.

[42] A. K. Jain, A. Ross, and S. Prabhakar. An introduction to biometric recognition. *Circuits and Systems for Video Technology, IEEE Transactions on*, 14(1):4–20, 2004.

[43] O. Jesorsky, K. J. Kirchberg, and R. W. Frischholz. Robust Face Detection Using the Hausdorff Distance. In J. Bigun and F. Smeraldi, editors, *Audio- and Video-Based Person Authentication - AVBPA 2001*, volume 2091 of *Lecture Notes in Computer Science*, pages 90–95, Halmstad, Sweden, 2001. Springer.

[44] K. Jia, S. Gong, and A. Leung. Coupling face registration and super-resolution. In *British Machine Vision Conference*, volume 2, pages 449–458, September 2006.

[45] K. Jonsson, J. Matas, J. Kittler, and S. Haberl. Saliency-based robust correlation for real-time face registration and verification. In *Proc British Machine Vision Conference BMVC98*, pages 44–53, 1998.

[46] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *International Joint Conference on Artificial Intelligence*, pages 1137–1145, 1995.

[47] S. Li and A. Jain, editors. *Handbook of Face Recognition*. Springer, Berlin, 1 edition, March 2005.

[48] A. Mahalanobis, B. V. K. V. Kumar, and D. Casasent. Minimum average correlation energy filters. *Applied Optics*, 26:3633–3640, 1987.

[49] J. Matas, K. Jonsson, and J. Kittler. Fast face localisation and verification. In *Image and Vision Computing*, volume 17, pages 575–581, 1999.

[50] Ministerie van Economische Zaken. Igc03003 biometric authentication supporting invisible security (basis). `http://www.senternovem.nl/iopgeneriekecommunicatie/projecten/IGC03003_Biometric_Authentication_Supporting_Invi.asp`.

[51] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):696–710, 1997.

[52] D. D. Muresan and T. W. Parks. Adaptive principal components and image denoising. In *ICIP (1)*, pages 101–104, 2003.

[53] J. A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, January 1965.

[54] NIST. Face recognition vendor test, 2006. `http://www.frvt.org/`.

[55] S. Osowski, A. Majkowski, and A. Cichocki. Robust PCA neural networks for random noise reduction of the data. In *ICASSP '97: Proceedings of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '97) -Volume 4*, page 3397, Washington, DC, USA, 1997. IEEE Computer Society.

[56] 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. In *In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[57] T. Riopka and T. Boult. The eyes have it. In *Proceedings of ACM SIGMM Multimedia Biometrics Methods and Applications Workshop.*, pages 9–16, Berkeley, CA, 2003.

[58] M. Savvides, R. Abiantun, J. Heo, S. Park, C. Xie, and B. Vijayakumar. Partial & holistic face recognition on frgc-ii data using support vector machine. *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW '06. Conference on*, pages 48–48, June 2006.

[59] M. Savvides and B. Vijaya Kumar. Efficient design of advanced correlation filters for robust distortion-tolerant face recognition. *Proceedings. IEEE Conference on Advanced Video and Signal Based Surveillance, 2003.*, pages 45–52, July 2003.

[60] A. Senior, R.-L. Hsu, M. A. Mottaleb, and A. K. Jain. Face detection in color images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):696–706, 2002.

[61] X. Shang. *Grip-pattern recognition: Applied to a smart gun*. PhD thesis, Univ. of Twente, Enschede, December 2008.

[62] Q. Tao. *Face Verification for Mobile Personal Devices*. PhD thesis, Univ. of Twente, February 2009.

[63] J. Tolkien. *The Hobbit – There and back again*. George Allen and Unwin, London, 1937.

[64] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of cognitive neuroscience*, pages 71–86, 1991.

[65] H. van Trees. *Detection, Estimation and Modulation Theory, Part I*. John Wiley and Sons, New York, 1968.

[66] R. N. J. Veldhuis. *Restoration of Lost Samples in Digital Signals*. Prentice Hall, New York, 1990.

[67] R. N. J. Veldhuis, A. M. Bazen, W. Booij, and A. Hendrikse. Hand-geometry recognition based on contour parameters. In *Proceedings of SPIE Biometric Technology for Human Identification II*, pages 344–353, Orlando, FL, USA, March 2005.

[68] P. Viola and M. Jones. Robust real-time object detection. *International Journal of Computer Vision*, 2002.

[69] P. A. Viola and M. J. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR (1)*, pages 511–518, 2001.

[70] P. Wang, M. B. Green, Q. Ji, and J. Wayman. Automatic eye detection and its validation. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, page 164, Washington, DC, USA, 2005. IEEE Computer Society.

[71] H. Wechsler. *Reliable Face Recognition Methods: System Design, Implementation and Evaluation (International Series on Biometrics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[72] M.-H. Yang, D. J. Kriegman, and N. Ahuja. Detecting faces in images: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(1):34–58, 2002.

[73] B. Zitova and J. Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21(11):977–1000, October 2003.

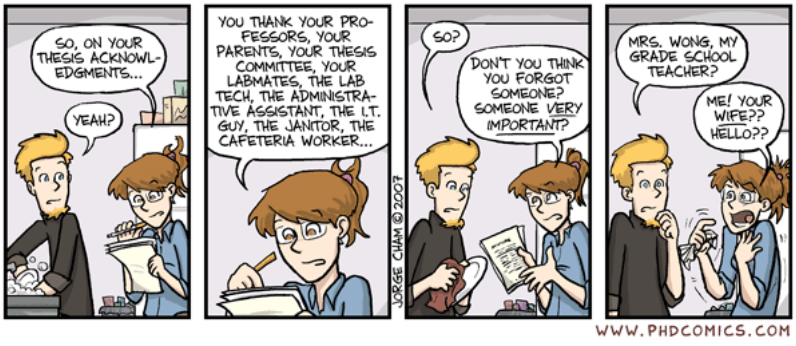
Acknowledgements

The last, almost, six years have been intense. After eight years as a student I signed up for another tour, as an 'Aio'. The ambition to pursue a Ph.D. and the decision to do so, grew long before I became aware of it. Driven by both hunger for knowledge and ambition to take on the challenge, I finally jumped to the opportunity given to my by Prof.dr.ir. Slump. For this I would like to thank him and offer my gratitude first.

Secondly, I would like to express my thanks to both Raymond Veldhuis and Asker Bazen for helping me start off during the first few months. Raymond is also credited for the freedom he gave me to choose my own path during the adventure I embarked on.

My colleagues at the group deserve an equal amount of thanks for all the lunches and drinks we had during and after work. It was an honour to work with you all. Xioaxin owns my gratitude for not only being the room-mate I could rely on, but also for becoming a personal friend. I too will remember our conversations. You were my window on China and it taught me a lot. Special thanks also go to Bas for both the work we did together and for the company he kept me in Singapore.

Also I would like to thank the University of Bologna, and especially Annalisa Franco, for my time in Italy. Although there was no direct scientific



"Piled Higher and Deeper" by Jorge Cham. www.phdcomics.com

output, the work we did influenced much of my later work. Italy made an impact on me which I will not soon forget.

Many friends supported me during the last few years. Most of all I would like to thank Geert for all the time he took to drink, sport or just talk with me. Special thanks to him and Pauline van Steijn-Buunk for their willingness to support me today as Paranymph. Special thanks go to Vincent Arkesteijn for his effort in proof-reading this thesis for mine and the readers benefit. Many people from underwaterhockey, the VriMiBo, Scintilla, #ligfiets and my dormitory at the campus always proved good friends through out the years. It would not have been fun without them.

For their willingness to facilitate and support the last and final hurdles of my thesis and the job opportunity they gave me I would like to that Gerard van den Eijkel and Robbert Evers. Without them I would have been a difficult and frustrating task.

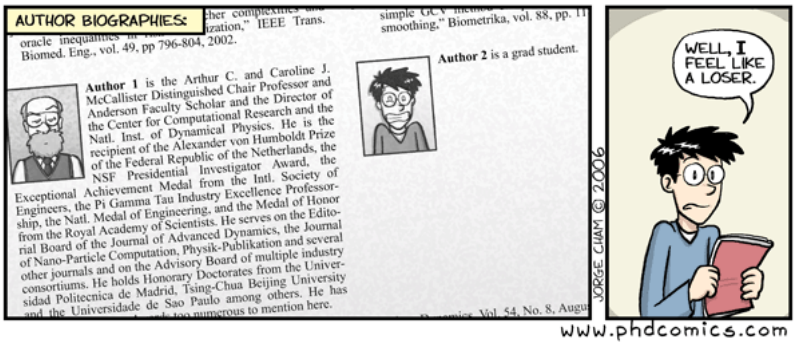
Furthermore, almost last but not least I would like to thank both my parents, for stimulating my love for knowledge and curiosity. I conclude with my thanks to Ingrid, for her loving support and faith in me. Did I mention her endless patience?

Gerrit Maarten Beumer
August 2009,
Enschede, the Netherlands

About the author

I was born as Gerrit Maarten Beumer on the 29th of August 1975 in Ede. After graduating from pre-university secondary education in Arnhem I started with Electrical engineering at the university of Twente in 1995. In 2003 I graduated after having found a, in my own opinion, perfect balance between study and extracurricular activities in the Integrated Circuit Design group. After that I continued as a PhD student at the same university but with a different group; The Signals and Systems group of Professor Slump. After little more than fourteen years my relationship with the university finally ends. Currently I am working for MECAL Focal as a technical specialist developing complex and special applications and technologies in vision and imaging.

Off duty I enjoy long distance cycling, reading graphic novels, underwaterhockey, freediving and watching films and the company of Ingrid.



"Piled Higher and Deeper" by Jorge Cham. www.phdcomics.com