BioMIS - Biometric Multi-factor Identification System

Martin Bornhøft

Kongens Lyngby 2011
IMM-M.Sc.-2011-28

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk

Summary

When a patient arrives at the Emergency Reception Ward at Frederiksberg Hospital the patient must be registered in their computer system, which can be done directly through a computer or using an electronic whiteboard, called "AMA-Tavlen" (the ER-Board). To access the ER-Board the staff uses a magnetic identification keycard.

As an extension of this authentication system the Biometric Multi-Factor Identification System (BioMIS) has been devised. An implementation of this extension made in C# programming language has been developed. Based on the results given in the test section, it is demonstrated that the use of multiple weak biometrics in connection with a primary biometric modality provides better performance than only using the primary biometric modality.

Resumé

Når en patient ankommer til Akut Modtage Afsnit på Frederiksberg Hospital så skal patienten registreres i deres edb-system, hvilket kan gøres direkte via en computer eller vha. et electronisk white board, kaldet "AMA-Tavlen". For at tilgå AMA-Tavlen bruger personalet et magnetisk nøglekort.

Som en udvidelse til dette autentifikationssystem er det Biometriske Multifaktor Identifikations-System (BioMIS) blevet udtænkt. En implementering af denne udvidelse, foretaget i programmeringssproget C#, er blevet udarbejdet. På baggrund af resultaterne givet i test afsnittet påvises det at brugen af flere svage biometriske modaliteter i sammenhæng med en primær biometrisk modalitet giver bedre performance end ved brug af kun den primære biometriske modalitet.

Preface

This thesis was prepared at the department of Informatics and Mathematical Modelling (DTU Informatics) at the Technical University of Denmark (DTU) as part of the requirement for obtaining a Masters degree in Science.

Additionally this thesis was created in collaboration with Pallas Informatik under the supervision of Svend Vitting Andersen and Rune Saaby Duschek. A collaboration between Pallas Informatik and Frederiksberg Hospital prompted the creation of this thesis.

The thesis deals with different forms of biometrics and the use of these in a multi-factor identification system called BioMIS. The main focus is on using a compination of soft biometrics (weight, height etc.) with a primery biometric (facial features, fingerprint etc.), thereby enhancing the performance of the authentication.

Lyngby, April 2011

Martin Bornhøft

Acknowledgements

I would like to thank my supervisor Christian Damsgaard Jensen at DTU Informatics for his guidance and encouragement during the project as well as my family for their support and understanding.

Furthermore I would like to thank Michael Højgaard Bojsen at Pallas Informatik and Peter René Bolvig Stentebjerg for helping with proof reading and support.

I would also like to thank my coworker at Pallas Informatik for technical support and a special thanks to Svend Vitting Andersen and Rune Saaby Duschek for supervision.

Contents

Summary **i**

Resumé **iii**

Preface **v**

Acknowledgements **vii**

1 Introduction **1**

2 Security **5**

3 Biometrics **7**

3.1 General Recognition Paradigm 11
3.2 Uni-modal Biometric Authentication 12
3.3 Soft Biometrics 12
3.4 Multi-modal Biometric Authentication 13
3.5 Fusion of Modalities 13
3.6 Score Normalisation 14

4 Face Recognition Overview **17**

4.1 Image Pre-processing 17
4.2 Face Localisation 18
4.3 Feature Extraction 19
4.4 Feature Matching 25
4.5 Discussion 27

5 Design **29**
5.1 Systems Description 29
5.2 Enrolment in BioMIS 30
5.3 Verification in BioMIS 31
5.4 Database Storage 32
5.5 System Structure 32

6 Implementation **39**
6.1 External Dependencies 39
6.2 Design Implementation 41
6.3 Programs Developed for Calculating Results 50

7 Tests **53**
7.1 Obtaining Data Samples 53
7.2 Test Scenario 56
7.3 Weight Controller Analysis 57
7.4 Simulated Height Controller Analysis 60
7.5 Face Controller Analysis 60
7.6 Combined Performance Analysis 65
7.7 Test Conclusions 68

8 Discussion & Further development **69**
8.1 Image Stream Acquisition 69
8.2 Training of the Data Set 70
8.3 Enrolment 71
8.4 Clustering 72
8.5 Larger Scale 72

9 Conclusion **73**

A Appendix **75**
A.1 Getting the AAM-API to work in a new project. 75
A.2 Compatibility Issues (.NET 4.0 & CLR/C++) 76

Chapter 1

Introduction

When a person receives an injury that requires immediate medical attention that person is rushed to the hospital. Arriving at the hospital the person will be registered for treatment, for instance if the person has broken his arm a doctor will need to have taken an x-ray image of the arm. All this procedural information (patient X needs to have right arm x-rayed, which was performed by Doctor Who etc.) are often just written down on a big whiteboard using whiteboard markers, but lately some hospitals are modernising these procedures and are now using a digital whiteboard to eliminate double registration (old school whiteboard and computer system). The injured person is quickly registered on the digital whiteboard and this information is automatical syncronized across the digital computer systems. However when dealing with personal information you only desire the relevant staff of the hospital to have access to this information. At Frederiksberg Hospital a digital whiteboard solution called "AMA-Tavle" (**A**kut **M**odtage **A**fsnit - Emergency Reception Ward - ER-Board) has been implemented in collaboration with Pallas Informatik. To gain access to the digital whiteboard at Frederiksberg Hospital a doctor or nurse will swipe an ID card to authenticate themselves to the system. This method was chosen because it is fast, you only need to swipe the card and then you are logged in. When dealing with an emergency reception department speed is a great factor in the daily workings of the department, as injured people's lives might be at stake.

This thesis seeks to improve on the digital whiteboard system implemented at Frederiksberg Hospital by examining the addition of a verification process. This is done based on the paradigm of multifactor authentication, where at least two different types of authentication are implemented. We will outline an identification system, which uses an ID card to establish the identity of the person seeking access to the system, and the BioMIS (Biometric Multifactor Identification System) verification process for a second factor authentication. An important factor in the verification process is that it is fast, which means that the process of verifying the person's identity should be no greater than a few seconds. The process of accessing the "AMA-Tavle" system using the BioMIS authentication is shown in Figure 1.1 The BioMIS authentication system consists

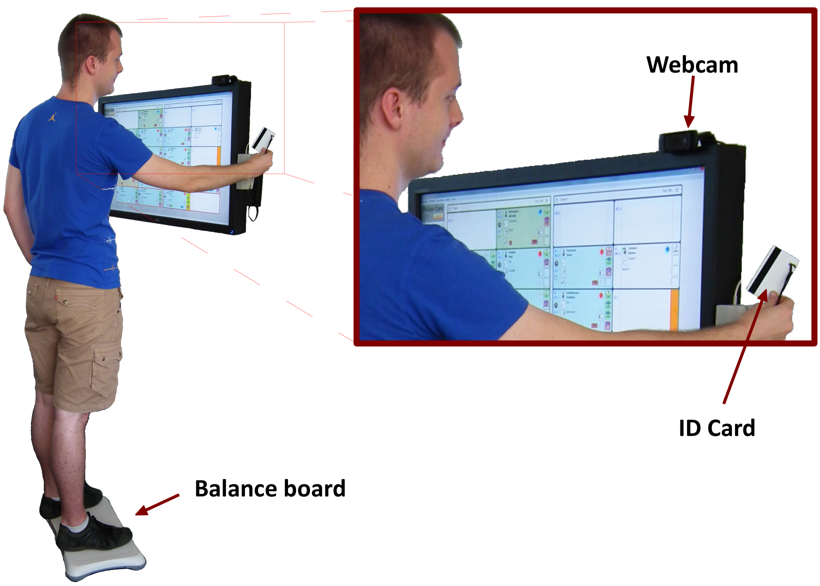

Figure 1.1: "AMA-Tavle" setup using the BioMIS authenticaton.

of two processes, identification and verification. The identification process will not be discussed in detail in this thesis, and focus will primarily be on the verification process, as an identification process using an magnetic strip ID card already in place at Frederiksberg Hospital. The verification process was chosen to consist of a face verification using Active Appearance Models (AAM), as I have worked on a face recognition system using AAMs before. In addition to the face verification, we have also chosen to use a Wii Balance Board as a weight and center of gravity verification. We look at how this impacts the performance of the overall verification process using multiple modalities during verification.

Tests to evaluate the performance of BioMIS are performed using the Equal Error Rate (EER), also known as the Crossover Error Rate (CER), given the False Acceptance Rate (FAR) and False Rejection Rate (FRR) of the BioMIS verification process. We also look at how using multi-modal verification affects overall performance of the BioMIS verification process.

Conclusions are drawn to indicate that using a multi-modal verification process gives a better performance than only using uni-modal verification, even using biometric modalities that are considered of lower quality than most biometric modalities.

The following gives a short overview of what is contained in the thesis project and a short description of these sections.

Security - This chapter contains an overview of uni- and multi-factor authentication in security systems.

Biometrics - This chapter describes what biometrics are and how these can be used in authentication systems. Uni-modal and multi-modal verification and the general recognition process used in biometric systems.

Face Recognition Overview - an overview of a typical face recognition system. What are the key components and what do these consist of. It will also contain a discussion, focusing on benefist and drawbacks using the different methods.

Design - This chapter describes the overall design of the Biometric Multi-factor Identification System (BioMIS). This incorporates the interaction between the different system components that are used in the verification process.

Implementation - This chapter contains an overview of the BioMIS software implementation, in .NET using C#, mimicking the functionality described in the Design chapter.

Tests - This chapter contains a description of the different test scenarios and the results of the implemented BioMIS verification process.

Discussion & Further Development - This chapter contains a discussion about the implication of the result presented in the Tests chapter. It also contains a short description of different ways of furthering the development of BioMIS.

Conclusion - This chapter will present master thesis project findings and summary conclusions.

Chapter 2

Security

This chapter describes some of the different types of security that are used in security systems today. Normally when you use a computer system you usually use a username and a password to identify yourself. In security systems we look at three types or categories of security that can be used to authenticate by

1. Something you have or posses such as an identification card or chip. The drawback of possessions are that they can easily be lost, illegally forged or duplicated.
2. Something you know, which is usually a password or a pin number. However a pin number can be forgotten or it can be shared, guessed or obtained without your knowledge (stolen).
3. Something you are, which is refered to as biometrics. Biometrics can be your DNA, fingerprint, handprint, or your face etc. Of cause this can also be obtained, for example a fingerprint can be obtained from a surface you have touched or someone can take your picture. Your face can also be mimicked to some degree.

The scenario mentioned above, where you use your username and password to authenticate yourself is known on the computer system as uni-factor authentication as only one of the before mentioned type of factor (the password as *something you know*) is used to authenticate. Today some laptop (like Lenovo's Thinkpad) computers also have a fingerprint sensor and uses this to authenticate individuals by. But again this is only a uni-factor authentication system, as only the fingerprint (fingerprint as a biometric is *something you are*) is used to authenticate [9].

In systems that require a higher security authentication, multi-factor authentication is often used. A multi-factor authentication is defined as an authentication system that uses more than one of the three types of authentication factors listed above. A system, which uses multiple authenticators from the same type (fx biometrics) is *not* considered a multi-factor authentication system. This means that an authentication system using both facial features and fingerprints, which are both of the type *something you are*, to authenticate an individual does not constitute a multi-factor authentication system. However a system that uses an ID card (something you have) and facial features (something you are) to authenticate an individual, would constitute a multi-factor authentication system, as these are from two different categories [9].

Today there are many multi-factor authentication systems in use, in which money transaction systems are probably the most common, where you use a credit card (*something you have*) and the pin number (*something you know*) to access the transaction system. This type of authentication system uses a binary method of authentication as you either possess the credit card or not, and the same for the pin number [4].

Other systems might use a probabilistic authentication paradigm, where the outcome of the different authentication mechanisms is assigned a value, that indicates the probability, that a user will be authenticated by the mechanism. Here it is not required, that all authenticating mechanisms succeed. It is the accumulated probability measure of all mechanisms that decide if an individual is authenticated or not. In case of probabilistic authentication a predefined threshold is used to determine the decision. An approach like this could be used in a multi-factor authentication system, which depend on a biometric authenticator, that uses multiple sources or devices to authenticate by [9].

Chapter 3

Biometrics

This section gives an overview of what biometrics are and for what purpose it can be used to identify a person. In forensics, biometrics are used to identify or survey criminals, who are under investigation of crime. In security systems biometrics are often used as a form of access control, where the goal is to identify a person seeking access to a secure system. It consists of methods for telling persons apart using either physical or behavioural traits. The physiological or physical traits are passed on by genetic inheritance from ancestors. Types of physiological characteristics are among others fingerprints, facial features, DNA, palm print, hand geometry, iris, retina and vocal cords [9] [18]. An overview of different biometrics is shown in Figure 3.1.

For a more indepth overview of the different biometric modalities please consult [18] *Chapter 2 - Biometrics: Overview.*

The behavioural characteristics are those traits that are not inherited but rather developed through routines. These characteristics relate to typing rhythm, walking pattern (gait) and voice or speech pattern. Some development of behavioural characteristics are influenced by social and cultural placement such as speech patterns.

For a human characteristic to be used as a valid biometric trait there are some parameters that need to be observed.

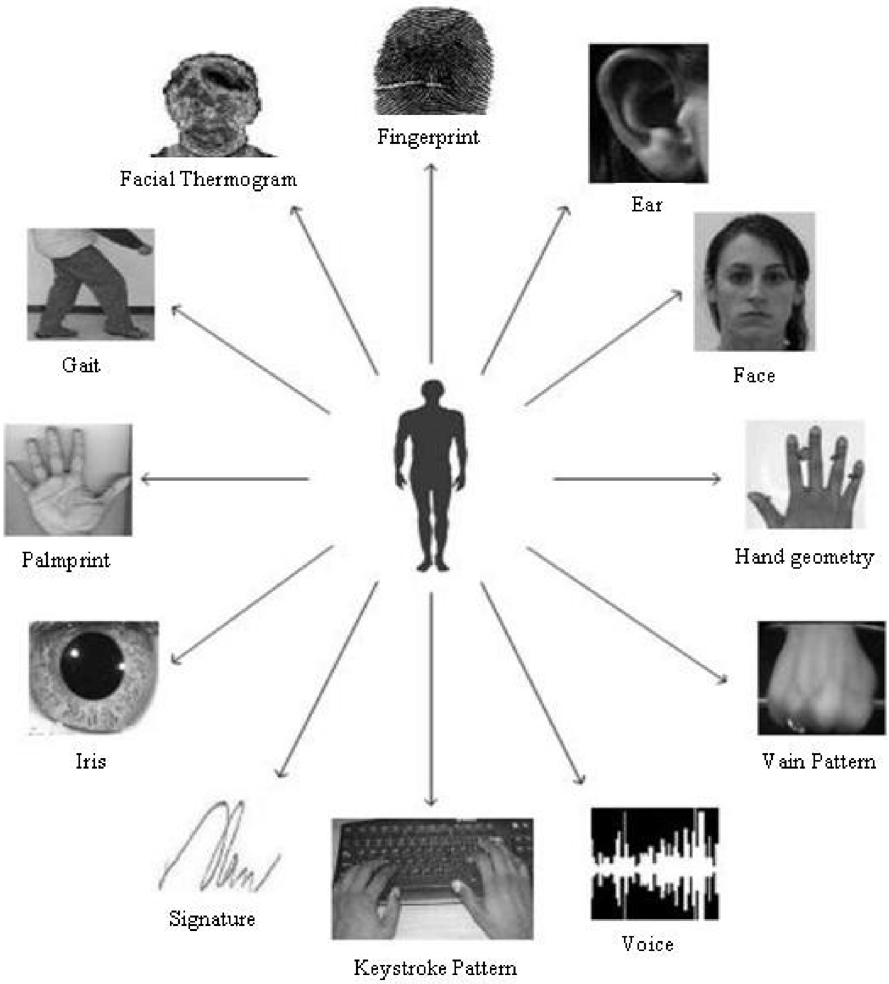

Figure 3.1: Different biometric features [15].

- Universality - means that the biometric trait should be some that all persons have.
- Uniqueness - defines how well a biometric trait separates individual persons from one another.
- Permanence - is how well the trait upholds to ageing or other some other variance over time.
- Collectability - determines how easy it is to acquire a sample for measurement.
- Performance - specifies how fast, accurate and robust the biometric trait is.
- Acceptability - how well the technology is accepted in general public.
- Circumvention - determines how easy it is for an imposter to circumvent or substitute the biometric trait.

Looking at a biometric system, there are two different states, verification and identification.

Verification concerns the one-to-one comparison of an acquired biometric sample with a print of a person stored in a database (recorded during enrolment). This means that the identity of the person using the system is already known, and the system only needs to verify, that the person is the person who he or she claims to be.

In identification, the person using the system is unknown, so the claim of identity needs to be established. Because of this, identification is a one to many comparison between the acquired biometric sample and all of the enrolled prints of the database. The identity of the person using the system is established based on the enrolled print with the best matching result. This being said the criteria for a successful match between a biometric sample and the database print is based on set threshold.

Before a person can use a biometric system, the individual needs to be *enrolled* in the system, which is called *enrolment*. The chosen biometric characteristics of the system (fingerprint, facial features, etc.) are recorded during enrolment and stored in a database. An overview of how enrolment is performed can be seen in Figure 3.2. Firstly a biometric sample is acquired from the sensor, which is then pre-processed (e.g. normalized) removing artefacts and enhancing the sample, so that it is easier to extract features from the sample. After normalisation, the

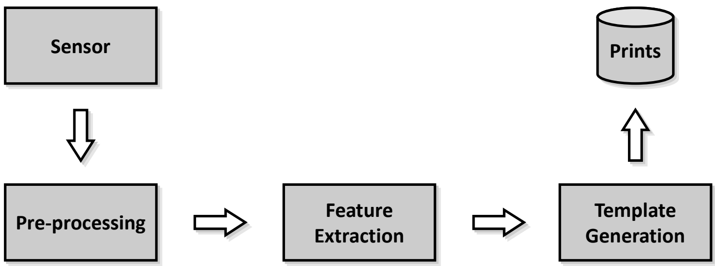

Figure 3.2: Biometric Enrolment.

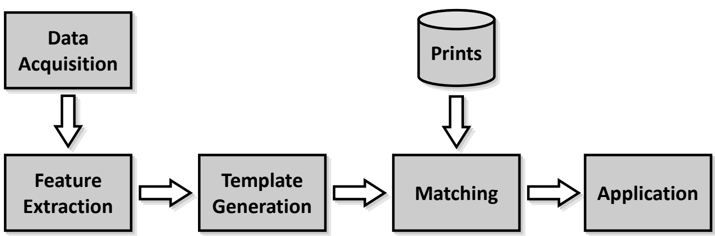

Figure 3.3: Biometric Verification.

biometric features are extracted, so that a template or print can be generated and stored in a database for later comparison.

During identification or verification a biometric sample is obtained just as when enrolment is done. Figure 3.3 shows the process of biometric verification, although here the sensor and pre-processing have been combined into data acquisition for convenience. After acquiring the sample and extracting the biometric features, these are compared with the enrolled print of a person (verification) or with the entire print database (identification). If the matching result is satisfactory this is passed on to the biometric application. The matching is usually calculated as a measure of distance (example Hamming distance), where the best fit is where the distance is lowest. The following section shows an overview of how a general recognition process proceeds.

3.1 General Recognition Paradigm

The process of any recognition system is usually finding the features of interest in sample data, retrieving this relevant information and then using it in some sort of matching between the information and some pre-observed or learned information. The process of pattern or subject recogntion can be broken into

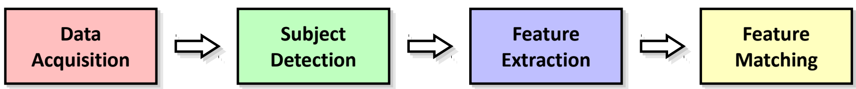

Figure 3.4: General recognition process.

the four steps as shown in Figure 3.4. Here a short description of the different steps or phases follows, where

Data acquisition concerns acquiring data from a source. For example obtaining an image from a webcam.

Detection or **localisation** concerns obtaining information about the presence of the desired subject within the data and locating this. This could be locating the face of a person in an image.

Feature extraction is about extracting relevant information about the detected subject from the data acquired. Here an example could be extracting the facial features of a person in an image, given some knowledge about the location of the face.

Feature matching involve measuring differences either known (supervised learning) or unknown (unsupervied learning), which could be matching minutiae points of fingerprints.

Between each step we can perform a pre- or post-processing of the data in order to get better results. An example is performing a bias correction before supplying the localisation phase with the raw data obtained in the acquisition phase.

Different classification schemes exist such as neural networks (where a special case is perceptrons), decision trees, support vector machines and statistical models (often referred to as regression analysis) just to name a few [2]. In this thesis we only look at matching where supervised learning has been used, using a training set of data samples.

BioMIS uses a classification scheme based on statistical models, therefore the following chapter describes different statistical models used in face recognition. The reason for choosing a statistical model classification method, is that the author of this thesis has previously worked with these types of models. An indeph analysis has not been performed on weight or height biometrics and should be considered a subject for future development. It is decided based on these biometrics being used as support for the face recognition to heighten the performance of the overall authentication system.

3.2 Uni-modal Biometric Authentication

Systems that use a single biometric trait such as fingerprint or face are called unimodal biometric systems, where authentication only uses one type of modality.

3.3 Soft Biometrics

Biometrics such as height, weight, gender, eye colour, ethnicity etc. have been coined *soft biometrics* by Jain, Dass & Nandakumar in [14] [16]. These biometrics are characterised as not having the distinctiveness and permanence to uniquely identify a person over a period of time.

> "Soft biometric traits are those characteristics that provide some

information about an individual, but lack the distinctiveness and permanence to sufficiently differentiate any two individuals."[14]

3.4 Multi-modal Biometric Authentication

Looking at multi-modal biometric authentication we will first have to define different types of multi-modalities. Lets us define the different modalities as classes, where a class is a biometric like fingerprint or iris. Biometric modalities can then occur *within* a given class or *across* different classes, and we will refer to these types of modalities as *within-class* modality and *across-class* modality respectively. If we look at face recognition, within-class modalities describes different methods of face recognition e.g. Fisherfaces and Eigenfaces. Across-class modalities are the different classes, e.g. fingerprint, face, palm print or vein pattern etc.

A system can be multi-modal in many different ways. One is where the category of the modality (e.g. fingerprint, facial, iris, etc.) is different, but you can also have different modalities within the same category, but using a different algorithm to extract the pattern that is later matched.

Another feature of multi-modal systems is the fusion of the results from the modalities. The fusion usually takes place in one of three states.

3.5 Fusion of Modalities

The different steps (data acquisition, object detection, feature extraction, feature matching) described previously can be thought of as modules. Fusion of the information can occur in each of these modules, like

- Fusion in the data acquisition module where the data itself, coming from multiple sensors or sources, is fused.
- Fusion at in the matching module, which fuses the match scores from different modalities.
- Fusion in decision where each of the decisions are fused together for example using majority voting.

It is believed that if fusion occurs at an early stage in the authentication process it will have a greater impact on the effectiveness of the performance than in later stages due to the fact that richer information is handled at the early stages. However it is also more problematic to fuse the information at earlier stages due to incompatibilities in the obtained data (for example fusion of eigen-coefficients of face and fingerprint minutiae) [19] [17], because of problems like this fusion is usually done in the later stages such as matching or decision making.

When dealing with fusion of modalities, then at some point the data needs to be comparable in some way, which brings us to score normalisation.

3.6 Score Normalisation

In multi-modal biometric system score normalisation deals with normalising the data to enable fusion of the different modalities. Here we look at some of the different types of normalisation.

Given a set of scores $\{s_k\}, k = 1, 2, ..., n$ observe the following normalisation schemes:

Min-max normalisation. If the minimum and maximum value of the data output are known then min-max normalisation can easily be shifted to scores between 0 and 1, respectively.

$$s'_k = \frac{s_k - \min}{\max - \min},$$

where s_k is the score value, and max and min are the maximum and minimum score value of the set of score values, respectively. Even if the minimum and maximum values are not known they can still be estimated from the known set of samples. However if this is the case, it renders the method unrobust, as the sample values might not be contained in the interval between min and max. Min-max normalisation still uses the same distribution as before normalisation except for a scaling factor, and the scores are transformed into a range from 0 to 1.

In *Decimal scaling* normalisation scores of one modality is scaled to a scale used by another modality. If the primary modality is in the range [0, 1] and the other in [0, 1000], we can use this normalisation method

$$s'_k = \frac{s_k}{10^n},$$

where $n = \log_{10}\max(s_i)$. However this methodology is based on the assumption that different modalities vary on a logarithmic factor, and still lacks robustness.

The *Z-score* normalisation technique is calculated using the arithmetic mean and standard deviation of the score values. If prior knowledge of the average and variation of the matching stage is known it performs well. If this information about the matching algorithm is not known the mean and standard deviation are estimated based on the set of scores. Z-score is given by

$$s'_k = \frac{s_k - \mu}{\sigma},$$

where μ is the arithmetic mean and σ is the standard deviation of the set of scores. If there are outliers in the distribution the z-score normalisation cannot retain robustness.

Using the score normalisation, which suits the data best can result in better performance as shown in [17].

In BioMIS, fusion occurs at both the matching and the decision stage in some cases. Fusion in the decision stage is done via a master controller, which makes the final decision of accepting or rejecting a person trying to gain access to the system. Fusion is also performed at the matching stage in a weight controller, by fusing the probabilities of the normal distribution of weight and center-of-gravity coordinates, and in a face controller, which fuses different statistical properties to obtain a higher matching score.

Chapter 4

Face Recognition Overview

Looking at face recognition, it is the task of recognizing a person given an image or a video stream of that persons face. In most face recognition systems an image pre-processing step is often coupled with the standard recognition scheme described in figure 4.1.

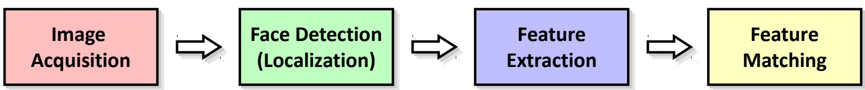

Figure 4.1: Simple overview of the face recognition process

4.1 Image Pre-processing

When constructing a face recognition system, you will often have a pre-processing step of the image, primarily to normalize the input for the localization and feature extraction steps. In case of first locating the image, a pre-processing step could be filtering of some kind to either highlight features for localisation or decreasing redundant background static.

4.2 Face Localisation

Before facial features for matching can be extracted, the overall placement of the face must be found - this done on a video stream (frames) or on an individual image. One of the more common methods for obtaining this localisation is the Viola-Jones face detection algorithm [20].

A comparative analysis of face localisation algorithms is not in the scope of this thesis. The object detection algorithm, which can be trained to detect faces, developed by Viola and Jones is chosen because an implementation is available in the OpenCV API [3].

4.2.1 Viola-Jones Face Detection

The Viola-Jones face detection system uses three main components for fast face detection, the *integral image*, using multiple weak classifier to make strong classifiers (using *AdaBoost*), and the *attentional cascade*, which is described shortly below.

4.2.1.1 Integral Image

The use of the *integral image*, which given at location x, y contains the sum of the pixels above and to the left of x, y

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y')$$

Here the integral image is denoted by $ii(x, y)$ and $i(x, y)$ is the original image. Using the recursive pair-wise structure, this can be shown as

$$\begin{aligned} s(x, y) &= s(x, y-1) + i(x, y) \\ ii(x, y) &= ii(x-1, y) + s(x, y) \end{aligned}$$

where $s(x, y)$ denotes the cumullative row sum, and it is assumed $s(x, -1) = 0$ and $ii(-1, y) = 0$. Using this information structure the integral image can be computed in one pass over the original image.

4.2.1.2 Classification

The AdaBoost learning algorithm is used by classification functions. It uses a lot of weak classifiers and combines them into a few strong classifiers, and uses, the attentional cascade to obtain fast classification. The purpose of the attentional cascade is to quickly sort through the classifiers, and as such, "attempts to reject as many negatives as possible at the earliest stage possible" - [20].

4.3 Feature Extraction

After locating the face in the input image, it is wise to perform an additional pre-processing step of the face image, so only the face will be analysed by the feature extraction algorithm. The face image pre-processing step, in connection to the feature extraction, usually consists of normalizing the face image, which often include an affine transformation of the image, so it is centered, rotated and scaled to some predefined orientation and size.

Looking at the feature extraction process, there are a lot of methods dealing with this problem. For the most part extraction algorithms fall into two categories, *appearence-based* and *feature-based*. The appearence-based approach utilizes holistic texture features and considers either the whole face or regions of the face for feature extraction. Feature-based face recognition use structures of facial features, such as the mouth, chin, nose and eyes, to find geometric relations between these. If you look at Figure 4.2 the overall process of feature extraction

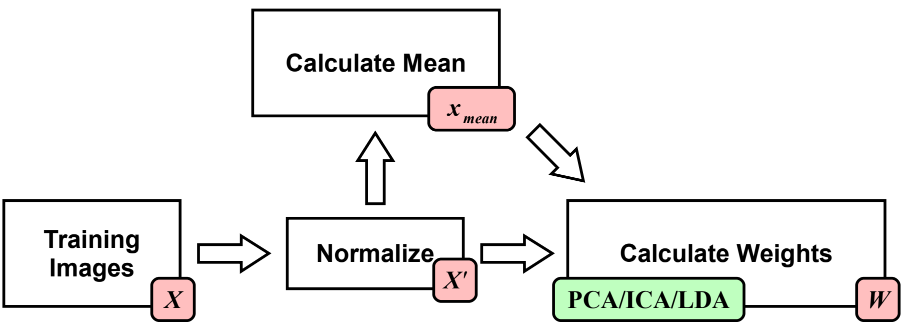

Figure 4.2: The training process - calculating the weight matrix.

training is shown. Here X are the images to be trained, each image expressed as a column in the matrix X. These are then normalized as expressed before, and are denoted X'. The mean face image x_{mean} of all the training faces is calculated. Given the mean face and the normalized training images matrix, and using the desired sub-space projection method, e.g., PCA, ICA or LDA, the projection matrix W^T is calculated [12]. When projecting the gallery images (training images) into the desired face sub-space, shown in Figure 4.3, each gallery image x_g is mean subtracted and projected into the face sub-space using the projection matrix. The galley image projections P are then stored in a database for use when matching is done.

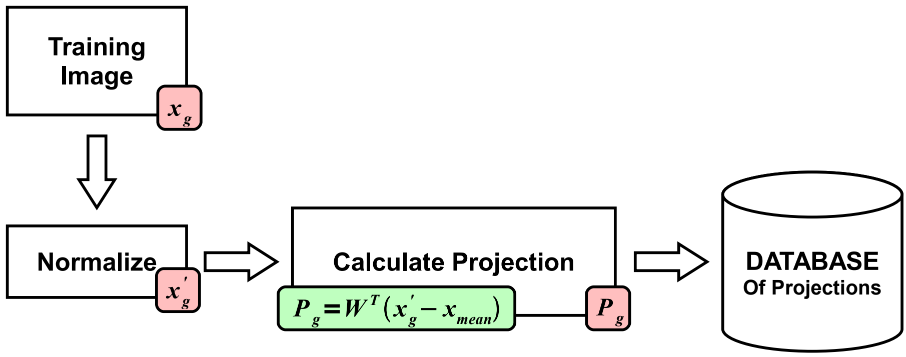

Figure 4.3: Each training image projection is stored in the database.

4.3.1 Feature extraction algorithms

The following sections contain an overview of different statistical face models and a discussion of the pros and cons there are among them.

4.3.1.1 Principal Component Analysis

Given a training set of m observations where each observation is represented as an s-dimensional vector, PCA is used to find a t-dimensional sub-space, in which the basis vectors correspond to the maximum variance in direction in the original space. In this sense PCA can be used to reduce the dimensionality of s to the sub-space t ($t \ll s$).

To compute the principal components of a system of s-dimensional data, you calculate the emperical mean, as the sum along each dimension $i = 1, ..., m$ in your data set

$$\mu[i] = \frac{1}{s}\sum_{j=1}^{s} X[i,j]$$

where the data set consists of m s-dimensional vectors concatenated into a matrix, denoted $X_{m\times s}$. Now subtract this from each column vector in the data set, so you have zero emperical mean, like so $\widetilde{X} = X - \mu$.

Now compute the covariance matrix using the mean centered data $\widetilde{X}$

$$C_{\widetilde{X}} = \frac{1}{s}\widetilde{X}\widetilde{X}^T$$

Having done this, we find the eigenvector decomposition of the covariance matrix as follows

$$V^{-1}C_{\widetilde{X}}V = D$$

Where D is a diagonal matrix, which consists of the eigenvalues for $C_{\widetilde{X}}$ and V consists of the corresponding eigenvectors. The eigenvector corresponding to the largest eigenvalue is then called the first principal component, the second largest the second principal component and so forth. This is also where the dimensionality reduction comes into play; if you think that a 95% estimation of your data covers enough, you can use the eigenvectors to project your data into a smaller sub-space by only using the principal components that cover 95% of the eigenvectors corresponding to the largest eigenvalue covering 95% of the total sum of the eigenvalues.

In image analysis the method of computing PCA on the face image data, where each face (observation) is represented as a vector (usually each column vector in the image concatenated into one long vector), is refered to as *Eigenfaces* [8]. If a computed eigenvector of a face image is restructured into a matrix, of the same dimension as an input image, and then displayed it would look something like figure 4.4 [8].

4.3.1.2 Independent Component Analysis

Independant component analysis is used to determine source signals, called the independent components, given a signal of mixed signals. To find these source signals we assume that these are hidden in the mixed signal. If the signals are represented as vectors, the task at hand is then to filter the mixed signal, x, using a transformation matrix W

$$s = Wx$$

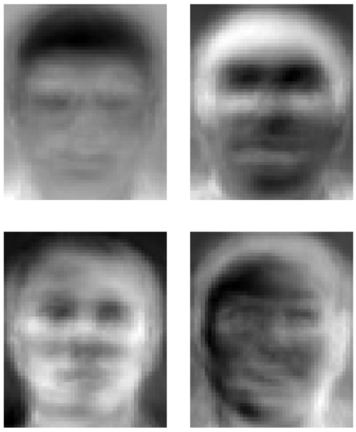

Figure 4.4: Eigenfaces.

into maximally independent components s measured by some function of independence. As the source signals (independent components) s are assumed to be hidden in x, the mixed signal is formatted in a special way, as a combination of mixing weights denoted by a, and the source signals.

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_i \\ \vdots \\ x_m \end{bmatrix} \qquad x_i = a_{i,1}s_1 + ... + a_{i,k}s_k + ... + a_{i,n}s_n$$

where s_k are the independent components and $a_{i,k}$ are the mixing weights, for $k = 1, ..., n$. More compact this can be written as

$$x = \sum_{k=1}^{n} s_k a_k \quad a_k = \begin{bmatrix} a_{1,k} \\ \vdots \\ a_{m,k} \end{bmatrix}$$

If all the k mixing vectors are concatenated into a matrix, we denote this matrix A and call it the *mixing matrix*:

$$A = \begin{bmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & \ddots & \vdots \\ a_{m,1} & \cdots & a_{m,n} \end{bmatrix}.$$

This gives rise to the defintion of x as $x = As$. Now W is defined as the inverse matrix of A, $W = A^{-1}$, and is known as the unmixing matrix.

4.3.1.3 Fisher's Linear Discriminant (FLD) and Fisherfaces

FLD is used to find the vectors that best descriminate among selected classes in an underlying space.

$$S_B = \sum_{i=1}^{c} N_i \cdot (x_i - \mu) \cdot (x_i - \mu)^T$$
$$S_W = \sum_{i=1}^{c} \sum_{x_k \epsilon X_i} (x_k - \mu_i) \cdot (x_k - \mu_i)^T,$$

where S_B and S_W denotes the *between class* scatter matrix and the *within class* scatter matrix respectively. μ_i is the mean image of class X_i, and N_i is the number of samples in the class X_i. In the case where S_W is non-singular, the optimal projection W_{opt} is given by maximizing the difference between the determinant of the between class scatter matrix and the determinant of the within class scatter matrix

$$\begin{aligned} W_{opt} &= \arg\max_W \frac{\mid W^T S_B W \mid}{\mid W^T S_W W \mid} \\ &= [\mathbf{w}_1 \mathbf{w}_2 \cdots \mathbf{w}_m] \end{aligned}$$

The problem is, that S_W is always singular in face recognition problems [8]. To remedy this, one can use a PCA to reduce the dimensionality of the subject data, which will result in the within scatter being non-singular, and then use the FLD. The procedure would look like this

$$\begin{aligned} W_{opt} &= W_{fld} W_{pca} \\ W_{pca} &= \arg\max_W \mid W^T S_T W \mid \\ W_{fld} &= \arg\max_W \frac{\mid W^T W_{pca}^T S_B W_{pca} W \mid}{\mid W^T W_{pca}^T S_W W_{pca} W \mid} \end{aligned}$$

4.3.1.4 Directional Independant Component Analysis

Using a directional image (an image schewed in its horizontal or vertical axis) it is possible to maintain some of the structural information of the image, which might otherwise be lost when using the *Independent Component Analisys* (ICA). The test results displayed in the article [21] show great improvement over unsupervised methods like Eigenfaces (PCA), ICA and 2DPCA, and a slight increase in recognition accuracy against supervised methods like Fisherfaces (FLD), Tensor-FLD and 2DFLD. The reason for using a directional multilinear ICA approach is to alleviate the problem, caused in normal ICA, of small sample size that leads to the dimensionality dilemma.

From the test results in [21] it seems a viable solution to the problem of feature extraction, although it still unclear how the face is located, as the research in the article is only done on already sampled and cropped data from a palmprint database, the UMIST face database and the AR face database.

4.3.1.5 Similarity-based Fisherfaces

This method uses geometrical data of faces (landmarks) and textural (visible and infrared spectral). The specific method for obtaining the geometrical landmarks is not discussed in [13]. After obtaining geometrical and textural feature representations, they are transformed into a similarity-based face representation. The method of the article differentiates itself from other known methods, fx Fisherfaces, as representing each enrolled person in the database as a set of n local 1-dimensional projections (one projection for each enrolled member in the database), as opposed to projecting data into an n-dimensional space. Test results show good accuracy and better or similar performance as competing algorithms.

If this approach was to be used in the face verification process of BioMIS, a pre-step to using this algorithm would be to run an *active appearence model* (AAM) or similar method to obtain the geometrical data.[13]

4.3.1.6 Active Appearance Models

Active Appearance Models (AAM) is a further development *Active Shape Models* (ASM), which basically are advanced active contour models (also known as snakes) guided using a mean shape. One of the drawbacks of ASM is that it only looks at locally around the points in the shape model, which can give erroneous results [10].

AAM takes the ASMs a step further, by adding the texture information to the model. Cootes *et al* describes AAM as manipulating a model that can synthesize new images of the object of interest. The goal of the AAM algorithm is to find the model parameters that create a synthetic image as close as possible match to the target image. It is further stated that "in each case the parameters of the best fitting model instance can be used for further processing, such as for measurement or classification" - [10]. This means using the shape and texture of an object (for instance a human face) to build a model that can describe the variation (textual and geometrical) of the training samples. Using this difference or error measure, try to improve the model parameters to get a better, lower, error measure. As mentioned in [11] [10], a bad starting point can result in failiure to recognize or convergance of the model. Using the mean texture and shape of the training samples, project the model onto the test sample (face of an unknown person) and calculate a difference measure. A basic formulation of the appearance model parameters, **c**, for controlling the shape and texture can be viewed as

$$\begin{aligned} \mathbf{x} &= \overline{\mathbf{x}} + \mathbf{Q}_s\mathbf{c} \\ \mathbf{g} &= \overline{\mathbf{g}} + \mathbf{Q}_g\mathbf{c} \end{aligned}$$

where $\overline{\mathbf{x}}$ is the mean shape of the shapes contained in the training set, $\overline{\mathbf{g}}$ is the mean texture in a mean shaped patch, and $\mathbf{Q}_s$, $\mathbf{Q}_g$ are matrices describing the modes of variation derived from the training set.

4.4 Feature Matching

When features have been extracted from a probe image, you need to determine if these features correspond to a person stored in the database of enrolled people. This is done by calculating the distance between the feature vector of the probe image and each of the stored feature vectors. The stored feature with the least distance will, to a certain degree of accuracy, be the closest match.

A graphic representation of the matching process can be seen in Figure 4.5 - where x_p denotes the test or probe image, P_p is the projection of the probe image, and d is the distance measure. In [12] four types of distance measures are used, and in the article they test if there is any difference in using distance measures with different feature extraction methods. The distance measures are: $L1$, $L2$, cosine angle (COS), and the Mahalanobis distance (MAH), which are given as follows

$$\begin{aligned} d_{L1}(x,y) &= \mid x - y \mid \\ d_{L2}(x,y) &= \|x - y\|^2 \\ d_{cos}(x,y) &= -\frac{x \cdot y}{\|x\| \cdot \|y\|} \\ d_{MAH}(x,y) &= \sqrt{(x-y)V^{-1}(x-y)^T} \end{aligned}$$

where V is the covariance matrix.

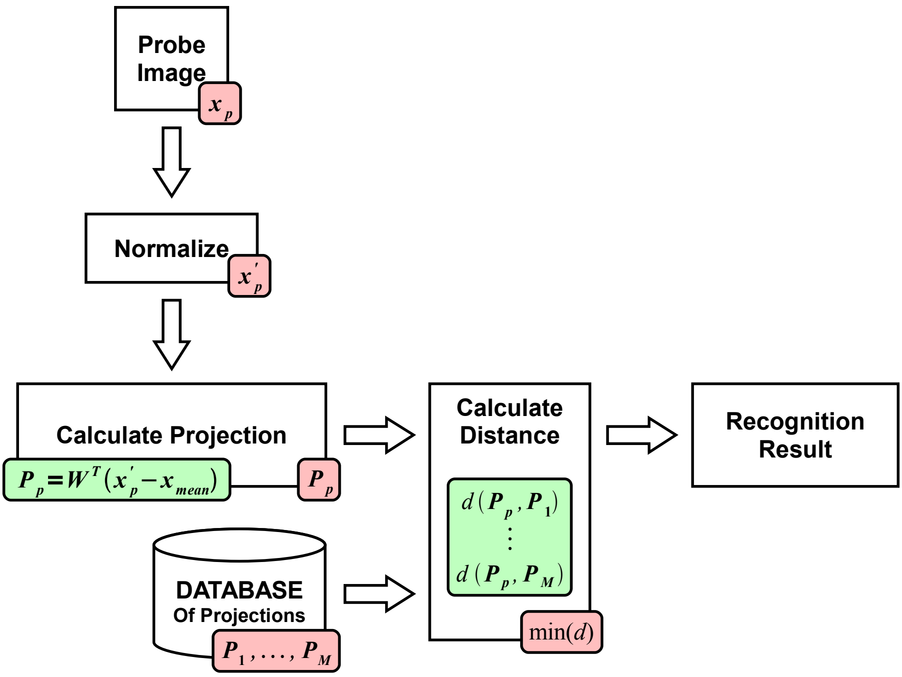

Figure 4.5: Feature Matching

4.5 Discussion

Looking at these different methods, we will discuss the advantedges and disadvantedges of these methods. Looking at the PCA method of dimensionallity redcution and projection (Eigenfaces) it has the advantedge of being simple, but this is also its drawback. Eigenfaces is highly affected by changes in posture and lighting conditions, according to [8]. Belhumeur *et al* constructed an algorithm, which uses a combination of PCA and Fisher's LDA (FLD) to improve the classification, but this algorithm still does not take the geometrical information into account. But this information could be added to the model if needed, as done in [13] using AAM. In [21] Zhang *et al* tries to incorporate structural information into the model, by schewing the images horizontally and vertically. Active Appearence Models uses both the structural- or geometrical information and the textural information with an adaptive approach to the facial model. This model uses the error between the model and the test subject to guide the next iteration of the model parameters.

Chapter 5

Design

This chapter will outline the design of the Biometric Multi-factor Identification System and give an overall view of the components contained in the system, and how these interact.

Furthermore this thesis will only look at the verification part of the person identification and therefore assumes that the identity of the person logging into the system is known. Preliminary identification of the person would be done, as suggested previously, using a colour-coded card, an RF ID card or a magnetic keycard as is in use at present.

5.1 Systems Description

For a doctor or a nurse to gain access to the "AMA-Tavle"-system, they need to be authenticated via the BioMIS system (hereafter refered to just as *the system*) and for this we use a webcam and other peripherals (used to capture biometric user information) and a piece of software. The process of authentication is overall a two step procedure - an identification process, secondly a multi-modal verification process.

Users will identify themselves by swiping an ID card[1] as shown in Figure 1.1 in the Introduction. It will then look-up the corresponding user reference in a database of enrollees of the system. Using this user reference the system will attempt verify the identity of the user using a multi-modal verification process. A successful verification will grant access the "AMA-Tavle" system.

However to be authenticated by BioMIS, the users need to be enrolled in the system.

For ease in the design of the computer system, as to which components should be used, it is a good idea to follow the setup of the recognition setup (reference to recog_setup).

5.2 Enrolment in BioMIS

When a person is enrolled into the system, a frontal portrait picture is supplied along with samples of the enrollee's height, weight and center of gravity. Ideally the picture should be taken using a standard setup with uniform lighting and positioning, this way all images will have the same scale and position, and light will not affect the captured images. After capturing the images of an enrollee, these will be annotated either manually or automatically. The manual approach requires the use of a piece of software like the AAMLab developed by Mikkel B. Stegmann. In case of an automatic approach a model with a sufficient face distribution will need to be used. If the face distribution is inadequate the AAMLab can be used as a semi-automatic annotation tool, where the best fit using the automatic method is used first and later the outlying landmarks that do not align with the face can be manually moved to the correct position.

Sample of weight and center of gravity are acquired by having the enrollee stand on the balance board. The enrollee should stand still while the sample is taken, to avoid distortions in the sampled distribution. The general approach to enrolment can be seen in Figure 5.1.

[1] A system using an identification card with a colour code on it, which was tracked using a webcam was also proposed. Once this card was found, the software would extract the colour code and analyze it. However this identification scheme was not implemented

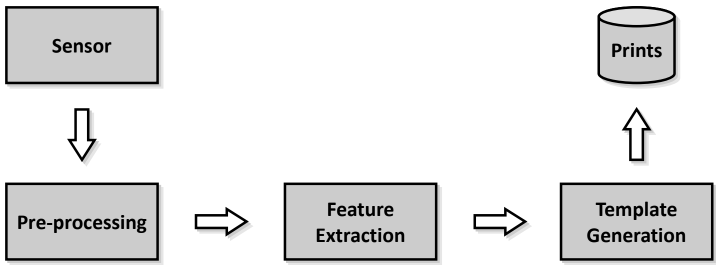

Figure 5.1: Biometric Enrolment.

5.3 Verification in BioMIS

When a user wants to gain access to the ER-Board, they will need to be verified using the previously obtained samples contained in the enrollee database. The general approach to verification can be viewed in Figure 5.2

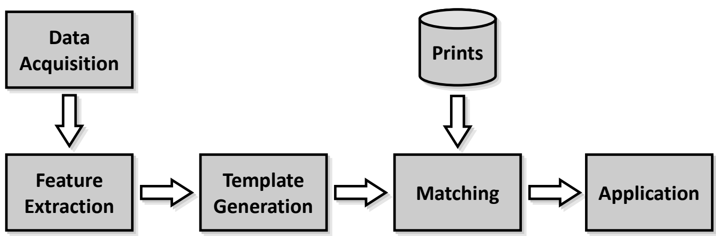

Figure 5.2: Biometric Verification.

The user will stand on the balance board and face the webcam, where the log in screen will indicate if the presence of a person is located. Swiping the magnetic idenfication keycard will identify the user to the system and will query the master controller for a verification. The master controller will then delegate this verification on to each of the slave controllers that will perform a recognition using the recognition process as outlined in Figure 5.3.

For easy reference each controller should have a method called *Enroll*, which is used to enroll a user into the system storing their respective prints or templates in the database, and a method called *Verify*, used to perform the verification procedure. Both of these methods should take as argument a *User* or *UserID* to identify the user by.

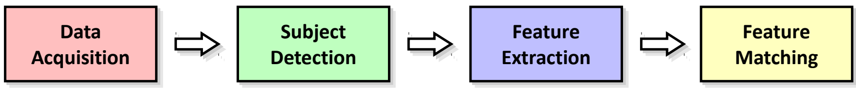

Figure 5.3: General recognition process.

When using soft biometrics, as they suffer from permanence issues, a method for continuous update of the distribution values should be present. The method should be called `Update`.

5.4 Database Storage

Using a central database or multiple databases (one for each controller). We use a distributed database model where each controller has their own database. This was chosen based on vendors of biometric systems and APIs each use a different way of managing their own database structure. If a mobile database paradigm like a smartcard is used, where the authentication data belonging to the enrollee is stored on, one database with all the different biometric information of the enrollee might be prefered.

5.5 System Structure

The Biometric Multi-factor Identification System (BioMIS) has two types of controllers. The first type is the more basic of the two, a slave controller, and the second type of controller is a master controller. The slave controller interacts with the biometric sensor, be it a camera for face recognition or a pressure gauge (Wii Balance Board) for weight and center of gravity data. The master controller interacts with the slave controllers and receives the interpreted sensor data from the slaves. A diagram of the relationship between the master and slave controllers can be seen in Figure 5.4. When initialising the system the master controller will delegate a command to the slave controllers to have them establish a connection to the sensor or device that they are controlling. A method for disconnecting a device should also be present.

When a user is be to verified the master controller sends a request to the each of the slave controllers to acquire a sample from their connected sensor. This sample is then normalized and compared to the templates of the person stored

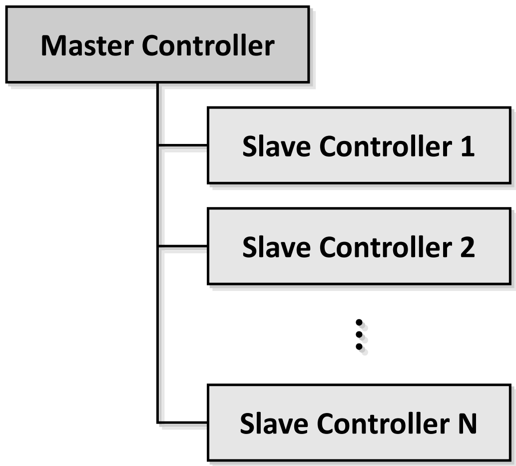

Figure 5.4: Overview of the structure between Master controller and Slave Controllers.

in the database. The best match result is then passed back to the master controller, which after collecting the match results from the slave controllers, fuses the match score, and based on a predetermined threshold value, accepts and grants further access to the "AMA-Tavle"-system or reject the claim of identity, and is denied access.

If the same individual tries and fails the verification process a certain number of times, the system should freeze any further attempts from the with the enrollee ID for a pre-determined length of time.

5.5.1 Image controller (slave controller)

The image controller has been designed as a module based system, divided into the principal parts of a general recognition system in the context of face recognition, as shown in Figure 3.4 on page 11. These parts consist of

Image Acquisition This is where capture or acquisition of an image from a device, for example a webcam, is performed. Furthermore if there is a need for normalization of the acquired image and the use or any other form of image processing for which the localization part of the design can benefit. For this system the image is scaled and transformed into a grayscale copy for better localization results.

Detection / Localization In this part of the system the localization of the face is performed. Any form of image preprocessing which the feature extraction part of the system can benefit from is also done in this part of the system. In this system a preprocessing of the image is done in for of a cropping of the image to narrow the search for the feature extraction part of the system.

Feature Extraction The extraction of features in this project is done via an active appearance model. In this case the features extracted are not really face feature but more features describing the face features.

Feature Matching After extracting the features, these need to be matched using an appropriate measure. In section [Face Recognition Overview] some distance measures are presented. In case of the active appearance model parameters, which are the extracted features, these are presented in a multi-dimensional space. It is assumed that the model parameters derived by the model of each person in the training set can be presented in this multi-dimensional space. Using the training set (images and annotations) the active appearance model can approximate the manifold or class a person or a group of people (in case of gender and race) belongs to. To match or classify a person to belong to a specific class in this space you can use the mahalanobis distance of measure between the sample person (i.e. the person who needs to be identified) and the persons in the training set, who belong to that class.

The benefit of designing the system in this way is that the modules can be replaced by a different one that uses a different algorithm. This type of module-based design also benefits from being easier to make unit tests for, as the different modules can be unit tested separately in closed environments. The technique of using modules in building such a system is refered to as a modular programming [1].

5.5.1.1 Verification

The face controller obtains an image using the webcam and uses the face localisation algorithm developed by Viola and Jones described on page 18 to obtain the location of the face. This is also the method used for detecting and tracking the user before authentication is attempted. The obtained face coordinates are then passed on to an active appearance model that uses the face coordinates as a starting point for the AAM search to obtain the model parameters. The model parameters are used to measure the Mahalanobis distance between the obtained model parameters and the model parameters stored in the database. Let $\mathbf{X}$ denote all the enrollees in a database

$$\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_m\}, \tag{5.1}$$

where $\mathbf{x}_i$ is an enrolled person of the database for $1 \leq i \leq m$, where m is the total number of enrollees.

Now let $\mathbf{P}$ denote the set of parameter vectors sampled of a person

$$\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_n\}, \tag{5.2}$$

where $\mathbf{p}_i$ is a parameter vector of a person and n is the number of samples for a person. Using the parameter vectors of a person, compute the mean vectors of each person

$$\overline{\mathbf{p}} = \frac{\sum_{i=1}^{n} \mathbf{p}_i}{n}. \tag{5.3}$$

Next compute the covariance matrices (or here the unbiased variance/covariance matrices) of the parameter vectors and the corresponding mean for each person

$$\mathbf{S}_p = \frac{\sum_{i=1}^{n} (\mathbf{p}_i - \overline{\mathbf{p}})(\mathbf{p}_i - \overline{\mathbf{p}})^T}{n-1}. \tag{5.4}$$

Using the mean and the covariance matrix of the set of parameter vectors belonging to a person enrolled in the database, we can now compute the mahalanobis distance between a sample (extracted parameter vector of a sample image) and the enrolled person

$$D_{MAHA}(\mathbf{p}_s) = \sqrt{(\mathbf{p}_s - \overline{\mathbf{p}})^T \mathbf{S}_p^{-1} (\mathbf{p}_s - \overline{\mathbf{p}})}). \tag{5.5}$$

An simplifyed example of how this could look is shown in Figure 5.5 below for a sample person belonging to the enrolled person with the green distribution. After obtaining the match score, a score normalisation is performed to transform the score values into a predetermined interval, for example [0, 100].

5.5.2 Weight controller (slave controller)

The weight controller handles the acquisition of weight samples. As a weight sample gained from a weight scale usually has multiple sensors for obtaining a weight measure, x and y coordinates of the center of gravity on the weight scale should also be obtained through the weight controller.

Fusion of these values can either occur at the matching step, using match score normalisation or later at the decision step in the master controller.

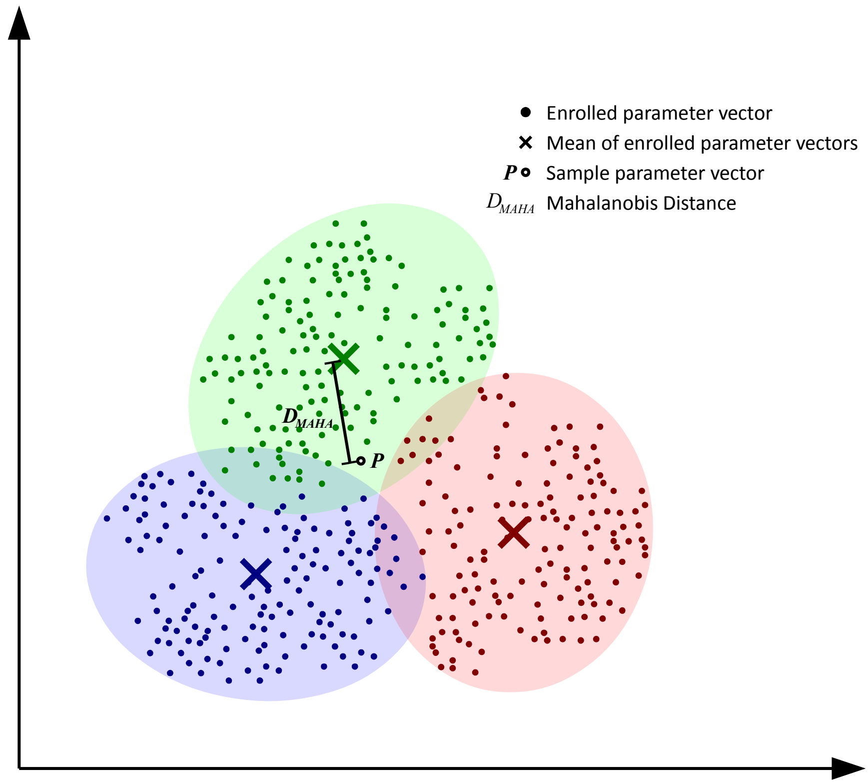

Figure 5.5: Graph of Active Appearance Model Recognition.

5.5.2.1 Verification

The verification process of the weight controller starts by obtaining a stable weight sample, where a stable value is obtained by the following algorithm

$w_{new} = 0$
$w_{old} = 0$
while $!stable$ **do**
  $w_s =$ sample from sensor...
  $p = \frac{c}{c+1}$
  $q = p - 1$
  $c = c + 1$
  $w_{new} = p \cdot w_{old} + q \cdot w_s$
  if $|w_{old} - w_{new}| < 0.001$ **then**
    $stable = true$
  else
    $w_{old} = w_{new}$
  end if
end while
return w_{new}

This algorithm calculates a progressive mean of the samples, which is returned if the difference between the newly calculated mean and the mean calculated in the previous iteration is smaller than a the threshold, which is 0.001. This is done for weight, and the x and y coordinates of the center of gravity provided by the balance board.

Using the the previously stored weight samples, W (in the enrollee database) of the user being authenticated, a normal distribution is generated. The probability value of the sample of the user, obtained during authentication, is then computed. This probability measure is then normalised to lie in the interval [0, 100] using the *scaled probability factor*, s_f, and using this factor a *scaled probability* is calculated. Let $W = \{w_1, w_2, ..., w_n\}$ denote the enrolled samples of the user, the scaled probability is then calculated using the following method:

$$f_{Norm}(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \mathrm{e}^{-\frac{(x-\mu)^2}{2\sigma^2}} \qquad -\infty < x < \infty \tag{5.6}$$

$$s_f = f_{Norm}(\mu_w; \mu_w, \sigma_w^2) \tag{5.7}$$

$$P_{w_{sample}} = \frac{f_{Norm}(w_{sample}; \mu_w, \sigma_w^2)}{s_f} \qquad w \in W \tag{5.8}$$

The same procedure is used for the center of gravity coordinates and the height.

A factor determines how wide the normal distribution of the weight is, as it is the factor by which the standard deviation is multiplied. This factor will be refered to as the *weight factor*. The implication of this is shown in Figure 5.6.

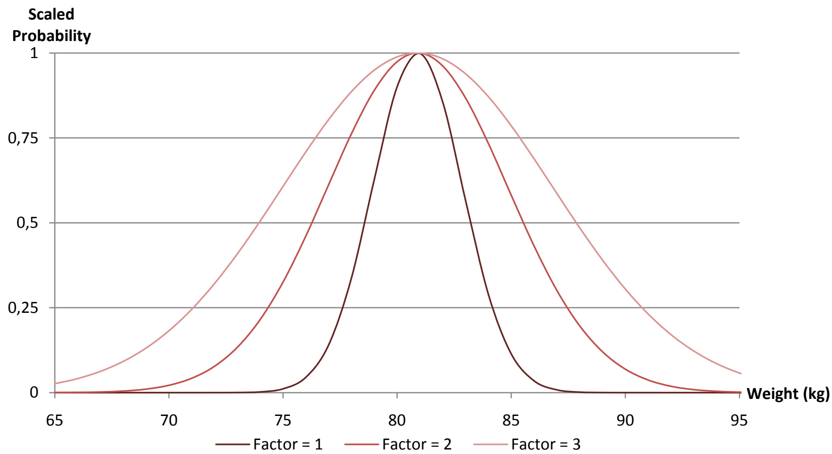

Figure 5.6: Scaled probability distribution of weight of subject0.

Chapter 6

Implementation

This chapter emphasizes on the details of the software implementation of the verification process of an authentication system. The motivation for developing this software was to create a structured piece of software, which is capable of performing face recognition as a primary concern, while utilizing soft biometric modalities to augment the performance of the face recogntion.

A screenshot of the BioMIS verification system is shown in Figure 6.1

6.1 External Dependencies

The software that has been implemented, utilizes three 3rd party application programming interfaces (API) or software development kits (SDK), which consists of

WiimoteLib - SDK for interacting with the Nintendo©Wii Balance Board^{TM} motion controller for the C# programming language[5].

Emgu OpenCV - A cross platform .Net wrapper to the Intel OpenCV image-processing library, which allows OpenCV[3] functions to be called from .NET compatible languages such as the C# programming language[6].

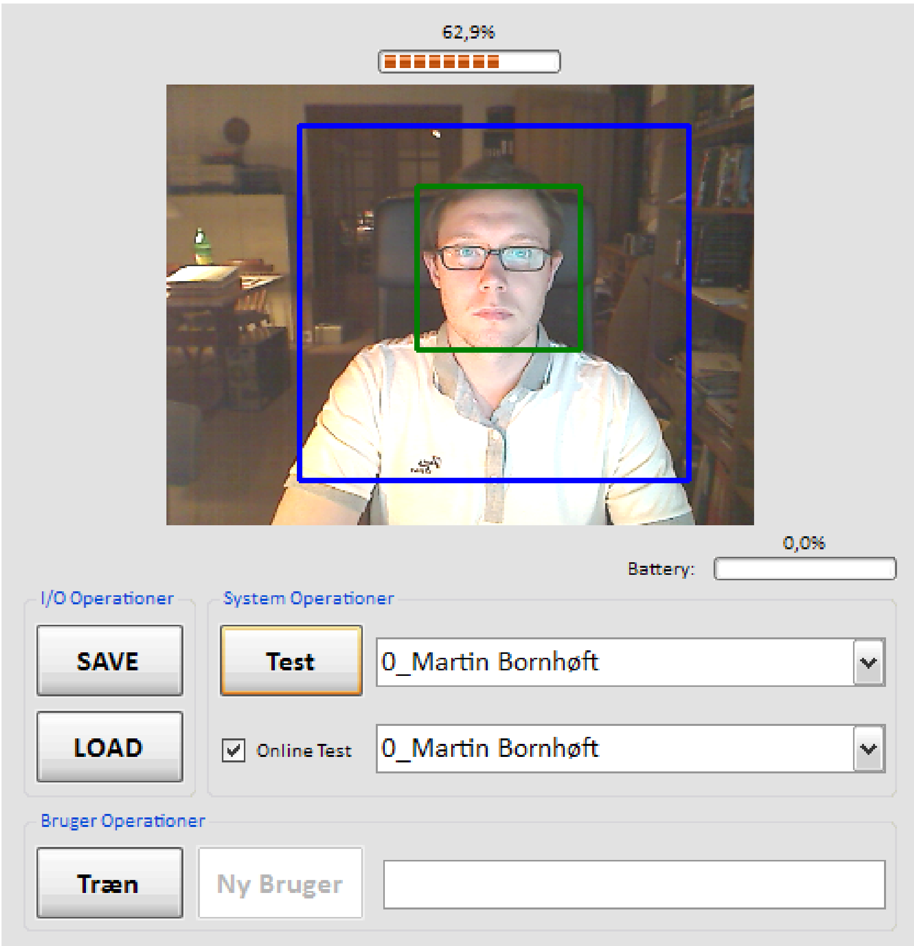

Figure 6.1: Screenshot of BioMIS in action.

AAM-API - A C++ implementation of the Active Appearance Model framework. It can be used to create and manipulate AAMs and was developed by Mikkel Bille Stegmann at the Department of Mathematical Modelling (IMM), at The Technical University of Denmark (DTU)[7].

6.2 Design Implementation

As stated in the design chapter, the software is comprised of two types of controllers, the master and the slave controller, where the master controller deligates the enrolment or verification query to the slave controllers.

6.2.1 Controller Interface

A controller implements the *IController* interface, which consists the following methods:

```
public interface IController
{
    void Device_Connect();
    void Device_Disconnect();

    void Save();
    void Load();

    void Enroll(UserRecord user);
    List<double> Verify(UserRecord user);
    List<double> Test(UserRecord dbUser,
                      UserRecord smUser);
    void Update(UserRecord user);
}
```

The `Device_Connect()` and `Device_Disconnect()` methods are used to connect to the sensor that is to obtain the sample data. For the face controller these methods would connect or disconnect to the webcam, or in case of the weight controller to the balance board.

The `Save()` and `Load()` methods are used to store or retrieve the database in which the samples of a user are stored. In this system the data is stored in an XML-file.

`Enroll(...)` takes a *UserRecord* as an argument, which is used to obtain the ID of the user so that the sample acquired from the sensor can be stored correctly in the database.

`Verify(...)` takes a *UserRecord* as an argument, which is used to obtain the ID of the user so that the sample acquired from the sensor can be stored correctly in the database.

`Test(...)` takes two *UserRecords* as arguments, where the first `dbUser` is the user record of the person enrolled in the database, and `smUser` is a user generated from a sample. The users are used to obtain the relevant database ID and obtain a sample with the right ID from the set of test samples. This method is used to obtain test results for further study.

6.2.2 Weight Controller

Enrolment using the weight controller obtains a sample by acquiring enough samples from the balance board, where *enough* means until a stable value has been obtained. The Wii©Balance Board^{TM} is connected to a computer using a bluetooth connection. This enables the balance board to accept connections as if the it was a *Human Interface Device*. Using the `Device_Connect()` method of the weight controller enables BioMIS to gain access to the values send from the balance board. The balance board supports both an event based value pushing system, where an event is triggered each time the balance board registers a new value, or a query value pulling system, where it is up to the controller to pull values when needed. BioMIS uses the latter method of obtaining data samples. A flowchart of the weight recognition process of the software is shown in Figure 6.2.

6.2.3 Face Controller

As outlined in the chapter on design the face controller is devided into modules, where each module performs a specific task, which consists of image acquisition, face localization, feature extraction and matching. A flowchart of the face recognition process of the software is shown in Figure 6.3, which starts with the acquisition of an image, which is captured by a webcam.

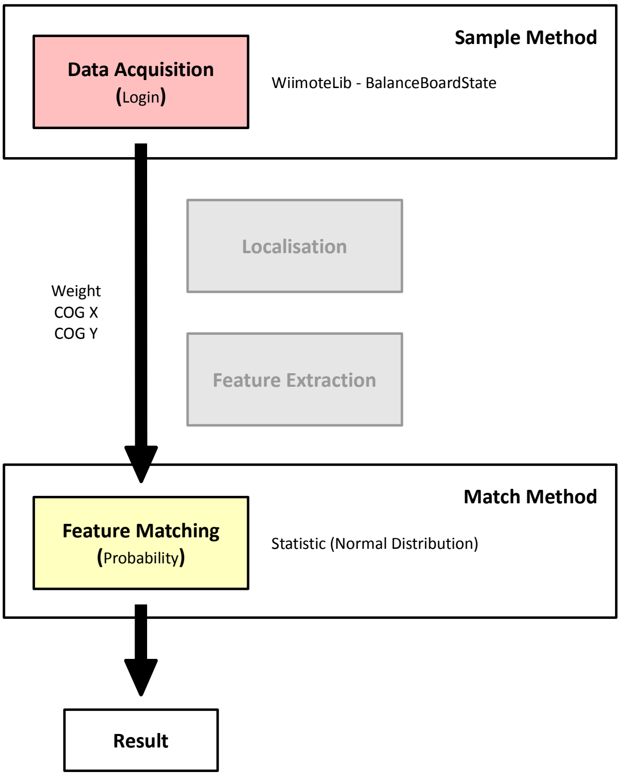

Figure 6.2: Program flow of the weight controller.

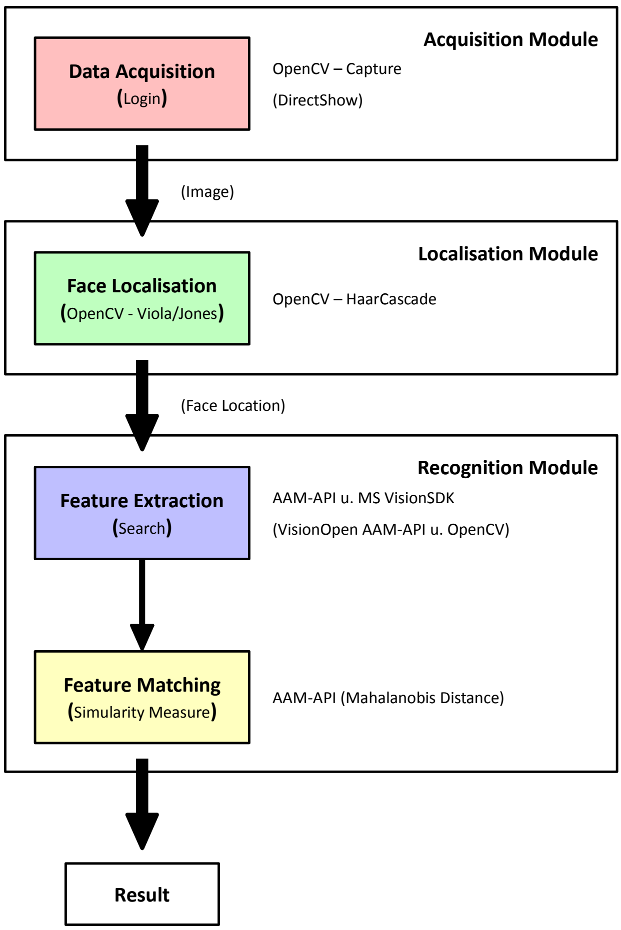

Figure 6.3: Program flow of the face controller.

6.2.3.1 FaceAcquisitionModule

The acquisition module handles the process of capturing an image from the webcam. This is achieved using the Capture class from the OpenCV API via the method `QueryFrame()`. If BioMIS is in verification or online mode the Capture class queries an image from the webcam. An example of the syntax used to obtain an image using the acquisition module is shown i Listing 6.1.

Listing 6.1: Obtaining an image from the ImageAcquisitionModule in online mode.

```
IAcquisitionModule acquisitionModule = new ImageAcquisitionModule();
Image<Bgr, Byte> image = AcquisitionModule.AcquireImage();
```

NOTE: A limitation of the Capture class is that the image acquired by QueryFrame() has a resolution of 640-by-480 pixels.

If BioMIS is in offline or test mode the acquisition module loads an image from a file given the filename. An example is shown in Listing 6.2

Listing 6.2: Obtaining an image from a file using the ImageAcquisitionModule.

```
Image<Bgr, Byte> result = null;
if (File.Exists(filename))
  result = new Image<Bgr,byte>(filename);
```

6.2.3.2 FaceLocalizationModule

The module only works in online mode. If BioMIS is in offline mode the location of the face is provided via the test set. The `LogInfo` class contains information about the location of the face in the acquired image. It also holds a reference to the acquired image.

```
public class LogInfo
{
    public Image<Bgr, Byte> FaceImage { get; set; }
    public PointF StartPoint { get; set; }
}
```

Listing 6.3: Locating a face in an image using the FaceLocalisationModule.

```
// First acquire an image from the acquisition module
Image<Bgr, Byte> image = imageAcquisitionModule.AcquireImage();
if (image != null)
{
   LogInfo log = faceLocalizationModule.Locate(image);
}
```

The `FaceLocalizationModule` uses the `HaarCascade` class from OpenCV. The `HaarCascade` is the OpenCV implementation of the Viola-Jones object detection algorithm. When in online mode the last known location of the face is stored and used to give a steady tracking of the face. To get a faster performance a cascade looking for the head and shoulders is used to filter out images where no person is detected. If a person is detected, the region of interest in which the person is located is passed on to the `FindFace` method, which performs the Viola-Jones object detection using a HaarCascade trained to detect frontal face images. The haar cascade classifier definitions are stored in pre-trained xml files, which can be obtained from different websites[1]. Supplied on the CD containing the BioMIS program are cascade classifiers for head/shoulders (person) and frontal faces.

6.2.3.3 AAMRecognitionModule

This module uses an active appearance model to extract features and match these to stored face prints. A face print is an AAM parameter vector. For the `AAMRecognitionModule` to work, an active appearance model file (example "model.amf") must be provided. Such a model file is created separately using the commandline program "aamcm" (developed by Mikkel B. Stegmann) for using active appearance models. It uses the following syntax

```
aamcm b . model config.acf
```

where the parameter "b" is the command for *building* a model, the "." is the path to the directory containing the training images and landmark annotations, "model" is the name of the binary output *model* file (it will have the extension

[1]For a frontal face classifier go to
http://code.google.com/p/warai/downloads/detail?name=haarcascade_frontalface_alt.xml.
For an eye classifier go to
http://www-personal.umich.edu/šhameem/haarcascade_eye.xml

".amf") and "config.acf" is the path to the configuration file. For more information on the AAM-API please consult the manual provided along with the AAM-API.

Storing the model parameter vectors for an individual is done using the `ExtractTrainParVectors(...)`, which takes as arguments the *path to the training directory of the individual* and the *path to the binary AAM output file.* Each of these are stored in the database. At present the database of the stored prints are in the form of a clear text xml file. This will have to be changed if a real world solution is to implemented.

At the time of writing BioMIS uses manual enrolment of faces using the program AAMLab (developed by Mikkel B. Stegmann). A screenshot of AAMLab in use is shown in Figure 6.4

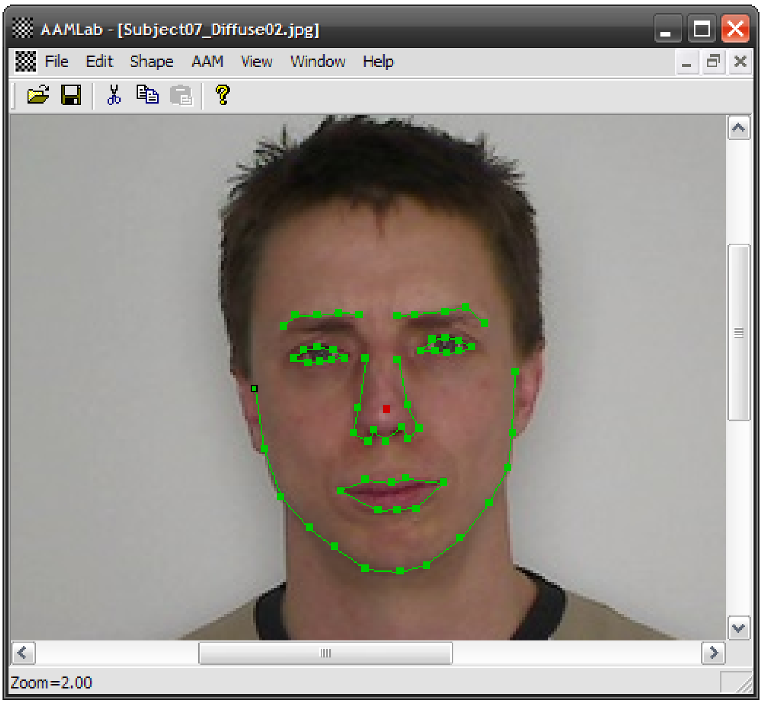

Figure 6.4: Program flow of the face controller.

The verification process used in the `AAMRecognitionModule` uses the class `AAMHelp`, located the assembly **AAM.Library.API.dll** written in C++ (by Martin Bornhøft), which was compiled as a managed assembly. It uses the

AAM-API to perform AAM searches [2]. A search is performed using the method `CSearch(...)` (contrained search), which has the signature shown in Listing 6.4

Listing 6.4: CSearch signature (C++).

```
int CSearch(
  System::String^ inModel,
  System::String^ inImage,
  int init_cog_x,
  int init_cog_y,
  [Out] double% mahalanobis,
  [Out] double% simularity,
  System::Collections::ArrayList^ c_Par )
```

Calling this method from C# is shown in Listing 6.5

Listing 6.5: Using CSearch from C#.

```
double mahalanobis = double.MaxValue;
double simularity = double.MaxValue;

// C# equivalent of C++ std:vector
ArrayList c_Pars = new ArrayList();

AAMHelp aam = new AAMHelp();
int ert = aam.CSearch(model, image, cog_x, cog_y, out mahalanobis,
     out simularity, c_Pars);
```

Fusion and match score normalisation of the mahalanobis distance and textural simularity measure is viewed in Listing 6.6

[2] Extensions to this class could move the general interaction (building models etc.) with active appearance models from the commandline program "aamcm", thereby limiting external program dependendies

Listing 6.6: Fusion of face recognition scores.

```
// For Mahalanobis
if (pMatch[0] <= 0.0)
{
  pMatch[0] = 100.0;
}
else
{
  pMatch[0] = Math.Min(50.0 * Math.Pow(pMatch[0], -1), 100.0);
}

// For Simularity
if (pMatch[1] <= 0.0)
{
  pMatch[1] = 100.0;
}
else
{
  pMatch[1] = Math.Min(4.5 * Math.Pow(pMatch[1], -1.2), 100.0);
}
result.Add(pMatch.Average());
```

6.2.4 Height Controller (SimulationController)

Given an xml file containing training samples The SimulationController uses a normal distribution, based on these samples, to perform the verification. It uses the same method of scaled probability as the weight controller to return a normalised matching score.

6.2.5 Master Controller (MasterController)

The master controller keeps track of the database of users along with their IDs. When the identification part of the system has obtained a user ID this is passed to the master controller for the verification part. The master controller passed checks if the user is stored in the database of enrollees, and if so, passes the user ID along to the slave controllers calling their `Verify` (or `Test`). The fusion of the matching scores is also performed in the master controller, using a simple arithmetic mean calculation. This is shown in Listing 6.7

Listing 6.7: Verfication method with match score fusion.

```
List<double> result = new List<double>();
if (Users.Contains(user))
{
  double sumValue = 0.0;

  // Fusion of match scores.
  for (int i = 0; i < m_Controllers.Count; ++i)
  {
    List<double> cResult = m_Controllers[i].Verify(user);

    if (cResult.Count > 0)
    {
      double cRes = cResult.Average() * m_Weights[i];
      sumValue += double.IsNaN(cRes) ? 0.0 : cRes;
    }
  }
  result.Add(sumValue);
}
return result;
```

6.3 Programs Developed for Calculating Results

As an extension to the main BioMIS program two programs, ResultMerger and DataGeneratorApp were created.

The program ResultMerger only works in debug mode, used as a script via the development invironment. It is used to

The program DataGeneratorApp is used to generate the sample values used by the simulated height controller, and to supplement the obtained values of the weight controller. It uses a mean and a standard deviation to generate random sample within the normal distribution. The number of samples generated can be specified. A sample is generated based on the method in Listing 6.8.

Listing 6.8: Sample generation.

```
Random rand = new Random((int)DateTime.Now.Ticks);
double factor = rand.NextDouble();
double x = Math.Round(rand.NextDouble()) == 0 ? m_Mean - (factor *
    m_StdDev) : m_Mean + (factor * m_StdDev);
return result;
```

The generated samples are stored in an xml file, which can be loaded by the SimulationController.

During the course of this project a class library of statistical classes was implemented called *Statistics*, which can be used to calculate the mean, variance, standard deviation and more of a sample set. This library also contains a class for instantiating a normal distribution, which, given a sample set or mean and standard deviation, is able to calculate the probability of a value belonging to the distribution.

An attempt at implementing a fingerprint controller using a Suprema BioMini fingerprint scanner was also performed during the course of the project. However due to complications and limitations with the free API provided by the driver manufacturer, implementing a working controller was unfeasable, as it was not possible to supply the verification process with previously obtained fingerprints, which means that the API only provided an online verification process. This made it unfeasible to setup test scenarios, and the controller was abandend. A developer license of the API could be purchased that would enable testing, but was deemed to expensive for the prospect of this project.

Automatic update of the soft biometric values has NOT been implemented. NEITHER has an automatic enrolment algorithm for enrolling annotated faces to the face controller database.

The source code can be viewed on the supplied CD in the folder marked "Source Code".

Chapter 7

Tests

In this section we will be cover how the different tests have been performed and the results obtained will be discussed. During the course of the project data samples were obtained to perform the tests. How these samples were obtained will be explained in the following section.

7.1 Obtaining Data Samples

Obtaining the data samples was in part done during two photo sessions for the acquisition of face portrait images and the weight samples were obtained continuesly during the course of this project. The following describes how the data samples were acquired.

7.1.1 Obtaining Images

The image data sets are captured using a Logitech®Webcam Pro 9000 and the corresponding Logitech®Webcam Software (QuickCam). The images are captured at a resolution of 1600x1200 pixels (or 2 megapixel) and then resized to a resolution of 640x480 pixels. This is done to save time on building the active appearance models, and because the OpenCV API capture size is limited to the latter resolution at the time of data sampling. It also saves space when training the models.

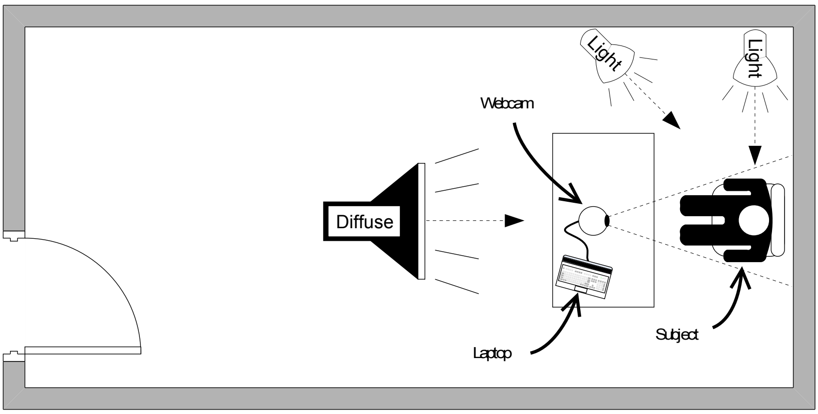

Figure 7.1: Setup of the photo session.

The images are sampled during two photo sessions that are two weeks apart; the images obtained during the first photo session are used as training samples; the images obtained during the second photo session are used as test samples. An overview of the photo session setup used is shown in Figure 7.1. During a photo session nine images were captured depicting the subject as either smiling, angry or looking neutral, and for each of these the three light setups diffuse frontal light, 45° and 90° offset spot light sources were used. The covered expressions can be seen in Figure 7.2

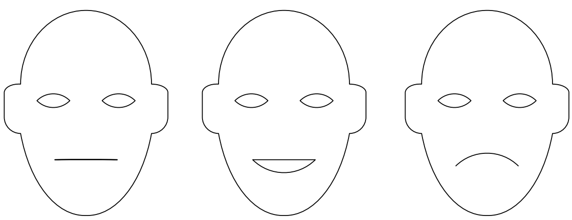

Figure 7.2: An image of a recognition system.

The tests that have been conducted focus on appearance and light source, to see which would have the greater variance on determining the person's identity. The

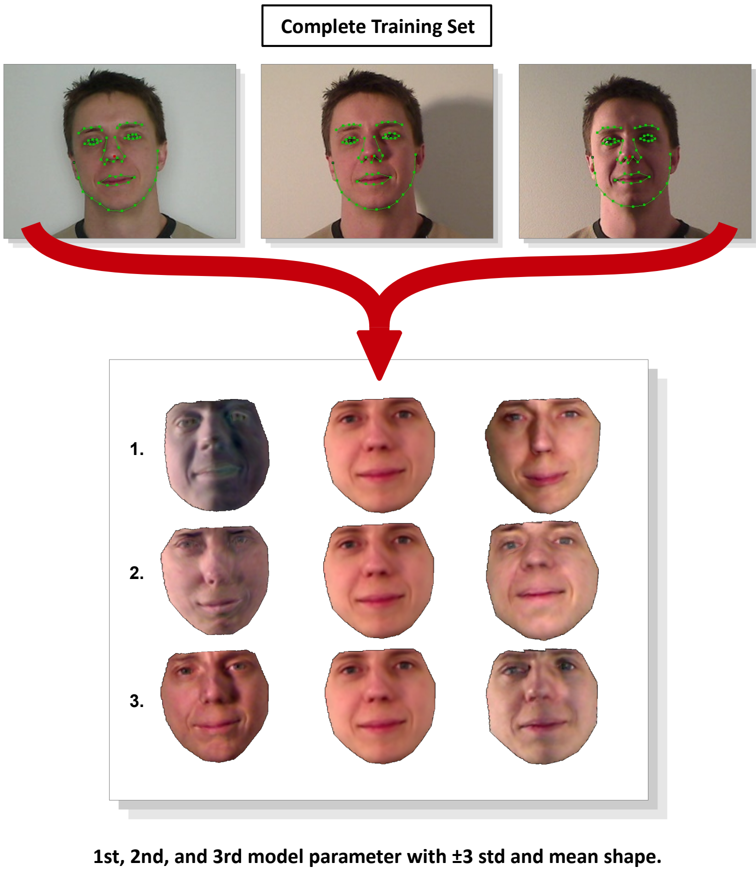

Figure 7.3: Depiction of the model training proces of the Complete Training Set, and model representation.

models are trained based on different image sets. Firstly we have the complete image set, where all images of a person are trained to one model (this will be referred to as the complete model).

This means that if you have ten persons enrolled in the face recognition system, there will be ten different models, one for each person 7.3. Looking at the first three parameters of the model we get some insight into where the model has the greatest variation. The most prominent variation is of the texture lighting, which makes sense as the variation in facial expression was limited to only three expressions. However the variation of expression is also prominent in the model.

7.1.2 Obtaining Weights & Center of Gravity

The weight and (x, y) coordinates of center of gravity are obtained using a Wii Balance Board. This was chosen due to having worked with the technology before.

7.1.3 Obtaining Heights

The height samples obtained during this project are simulated. A height sample of each enrollee was obtained later during an inquiry, in which the height of the enrollees was measured. These height samples were then used to generate additional synthetic height samples using a normal distribution where the mean was the measured height sample with a 2 cm standard deviation, which was used to generate a distribution.

7.2 Test Scenario

The scenario used in the tests is of the verification process, and measuring how well the controller performs. To measure the performance of a controller the *false acceptance rates* (FAR) and *false rejection rates* (FRR) are calculated at varying threshold level. Using the readings of the FAR and FRR, the *equal error rate* (EER) can be calculated. This is done as follows

$$FAR = \frac{FA}{FA + FR} \tag{7.1}$$

$$FRR = \frac{FR}{FA + FR} \tag{7.2}$$

$$EER = FAR - FRR = 0 \tag{7.3}$$

where FA is the number of *false acceptances* or *false positives*, and FR is the number of *false rejections* or *false negatives*, at a given threshold level. EER is given at the threshold level, t, where $FAR(t) = FRR(t)$. The equal error rate is used as the performance measure by which the controllers are tested.

7.3 Weight Controller Analysis

In this section we analyse the test result of the weight controller. To get a better understand of how the training database is varied, a graph of this is shown in Figure 7.4. Looking at this figure also gives some insight into which users are more likely to be recognised during test as false positives (aka. imposters). The more the distributions overlap the higher the likelihood of a false positive.

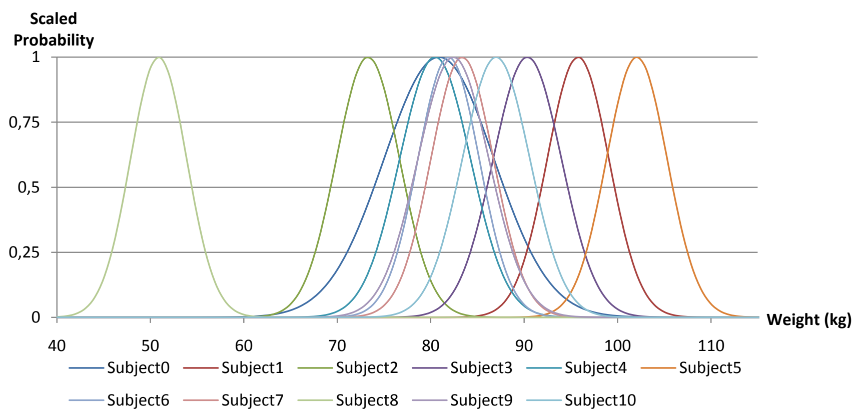

Figure 7.4: Overlaps in scaled weight distribution of training samples (database).

The graphs displayed in Figures 7.5, 7.6 and 7.7 show a constant performance even though the test are performed using different weight distribution factors (1.0 - 3.0). The weight distribution factor is also refered to as the *weight factor*. The only change observed is the rising threshold level. However this might be a result of not having enough samples. More samples would generate a higher number of false positives, which in turn would give a lower performance rate.

If the weight distribution factor is set low enough (0.1, only using 10% of the sample variance) it is observed that the performance rate markedly worsens, as shown in Figure 7.8.

Over a longer period of time the weight of a person is due to change if not strictly monitored for example loosing weight during a diet. An extension to the simulated weight samples could be a weight gain or loss to observe a more natural tendency in the performance of the weight controller.

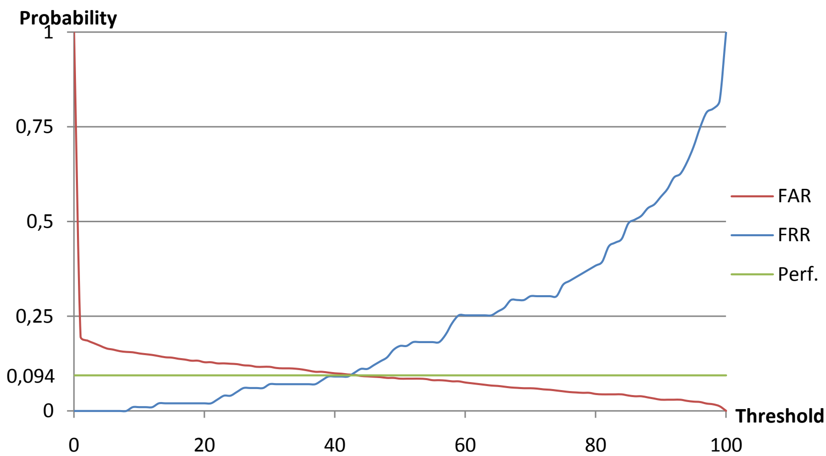

Figure 7.5: Performance of Weight Controller with weight factor = 1.

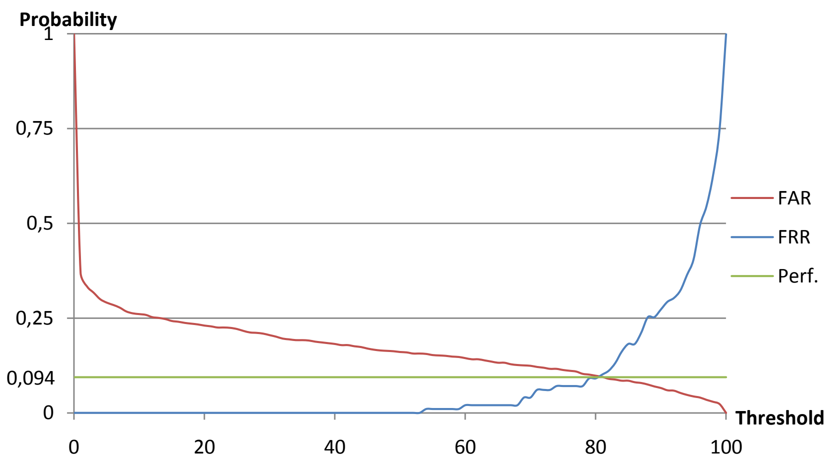

Figure 7.6: Performance of Weight Controller with weight factor = 2.

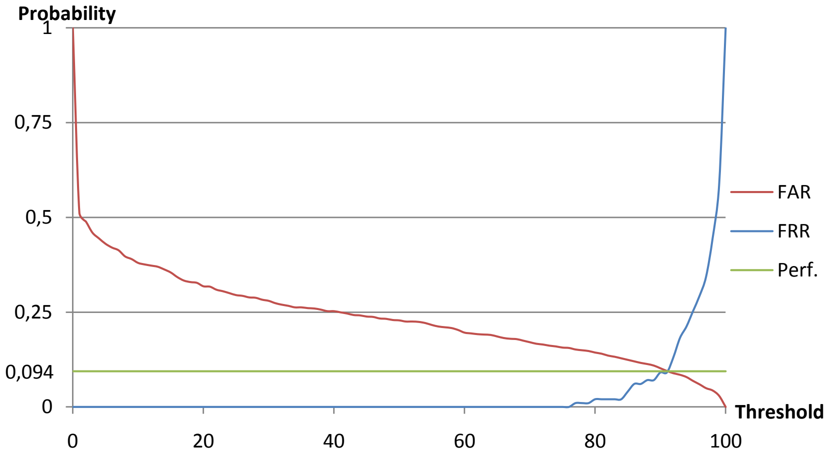

Figure 7.7: Performance of Weight Controller with weight factor = 3.

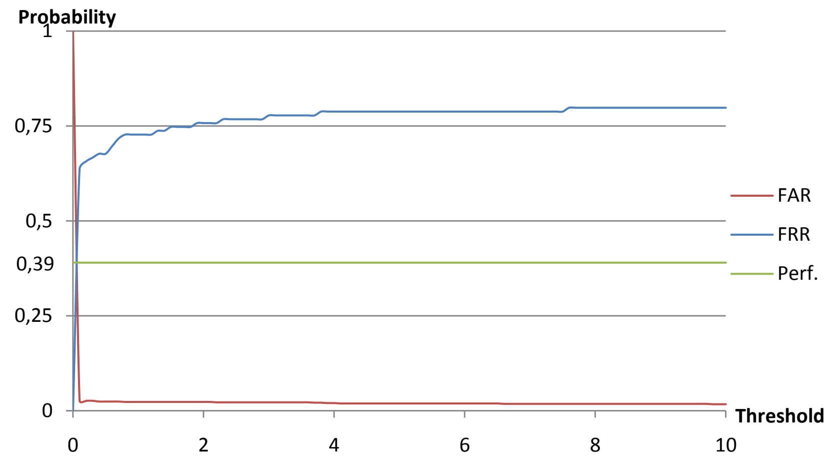

Figure 7.8: Performance of Weight Controller with weight factor = 0.1.

7.4 Simulated Height Controller Analysis

In this section we analyse the test result of the height controller. Before we look at the test scenarios, we might look at how well the training data is distributed, like we did before with the weight controller. In Figure 7.9 we get an overview of the individual distributions shown as scaled probabilities.

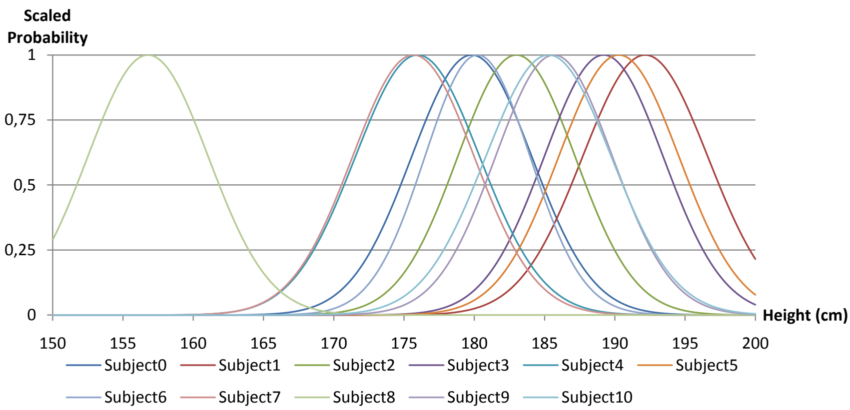

Figure 7.9: Overlaps in scaled height distribution of training samples (database).

7.4.1 Height Controller Performance Test

Looking at the tests performed to evaluate the performance of the controller, the same pattern of the static performance rate of $EER = 0.106$ while using different factors of the height distribution variance (height factor) as shown in Figure 7.10 through 7.12.

It was observed using extreme values of the height factor that it affected the performance rate in the same way as was the case with the weight controller.

7.5 Face Controller Analysis

In this section we discuss which tests that have been performed and then show the results of the tests using the face controller. The test scenario we look at is

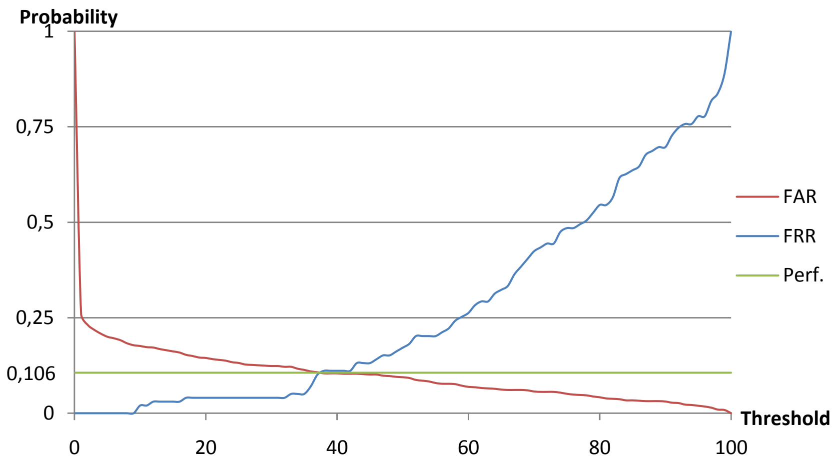

Figure 7.10: Performance of Height Controller with height factor = 1.

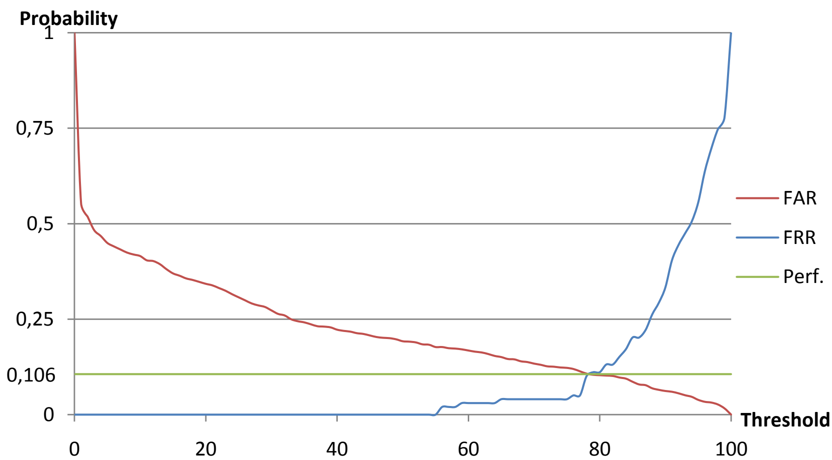

Figure 7.11: Performance of Height Controller with height factor = 2.

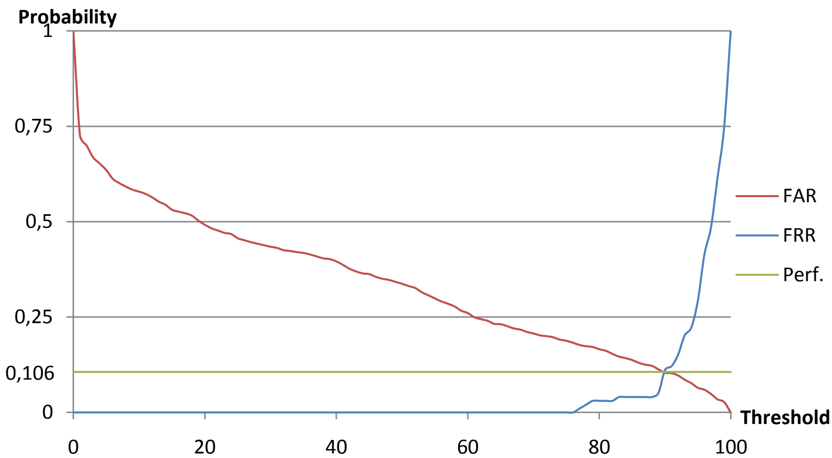

Figure 7.12: Performance of Height Controller with height factor = 3.

the performance of the verification process of the

7.5.1 Face Controller Performance Test

Looking at the face controller, three tests will be performed and examined. The first two test are performed without fusing the match scores, while the third test is performed on the fused scores. As the controller has two different values for measuring the distance between the active appearance model and the input image, the first test of the face controller will focues on measuring the performance of the face controller while only using the mahalanobis distance from model parameter vector to the mean (or closest enrolled sample model parameter vector) to calculate the match score. The second test focuses on using the textural similarity measure (pixel wise difference) and the third test looks at the performance of the controller, while using fusion of the two match scores as an average.

Figure 7.13 shows the best performance of the controller.

When looking at the graph in Figure 7.13, which shows the performance of the face controller when only using the Mahalanobis distance as a measure of distance, we can see that the overall performance is at best mediocre. It only classifies correctly about 65% of the time. As the data set for training is limited to only nine pictures with corresponding landmarks for each person, this is to

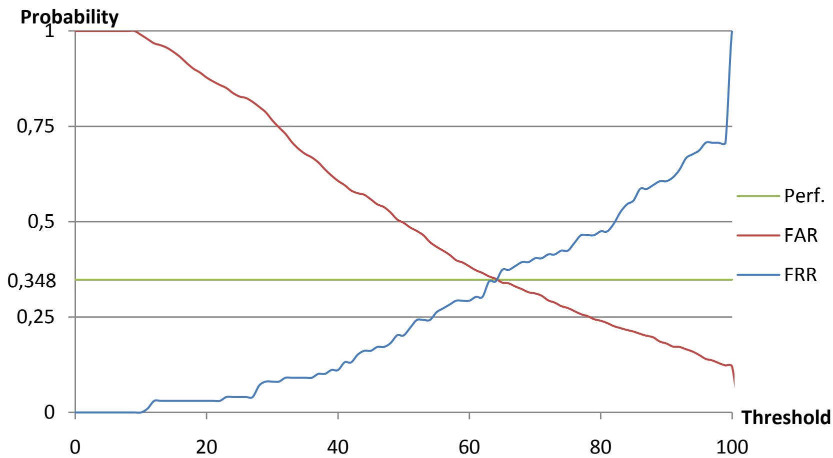

Figure 7.13: Performance of the Face Controller using only Mahalanobis distance.

be expected. The pose of the enrollee is limited to only three facial expressions (see Figure 7.2) and three different light sources (a spotlight at 45° and 90°, and a frontal diffuse light).

Looking at the graph in Figure 7.14 depicting the performance of the face controller using only the pixel wise similarity as a distance measure, you could be tempted to just use the textural similarity measure as a performance measure, but this could give some unwanted anomalies. As textural similarity is calculated as the pixel wise difference between the shape pixel and the input image pixel at the same location, the textural difference might be very low, but the shape of the model can be miles away, so to speak, from the actual face. If the model converges on a single pixel, where the colour of the pixel in the acquired image has the same value as the mean of the colour distribution of the AAM texture patch, the textural distance is 0 and will result in a perfect match.

Turning our attention to the last graph shown in Figure 7.15, which illustrates the performance of the face controller using the combined measure of the difference in the texture and the Mahalanobis distance, we can see that the performance is better than that of the individual measures (texture and Mahalanobis distance). Also by combining both the textural similarity measure and the shape model distance, as the Mahalanobis distance, we attempt to alleviate the problem of only using the textural similarity measure.

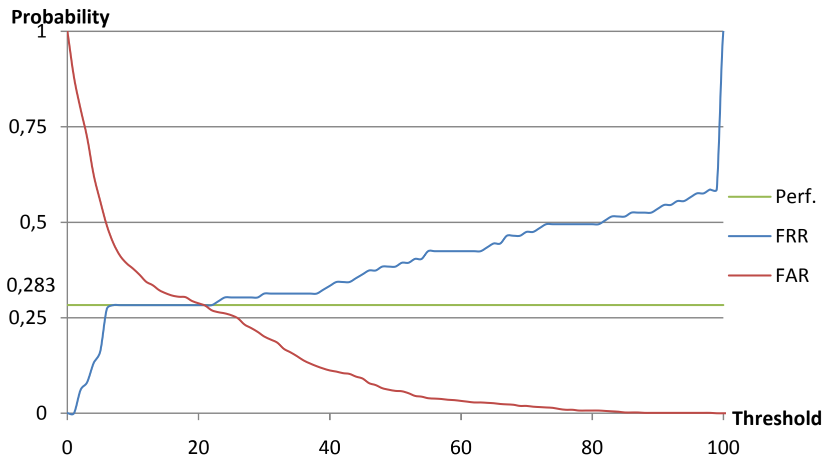

Figure 7.14: Performance of Face Controller using only textural difference.

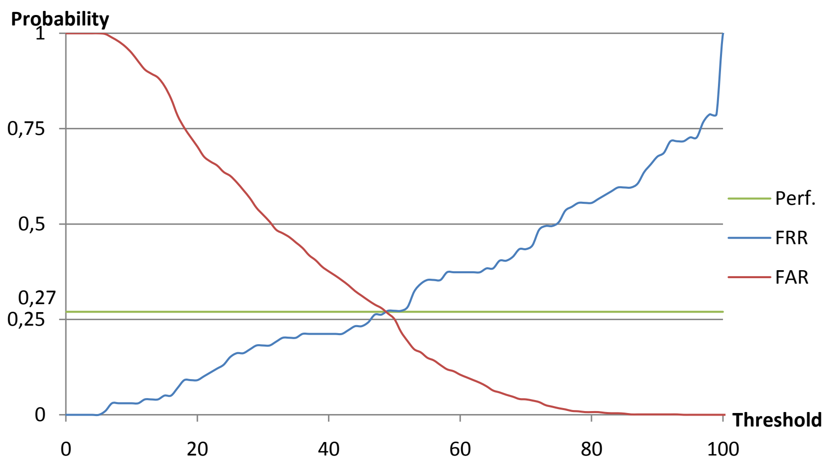

Figure 7.15: Performance of combined (Mahalanobis distance & textural difference) Face Controller.

7.6 Combined Performance Analysis

If we look at the combined performance of the system using all available controllers (height, weight and face) this section will discuss the obtained results. Looking at Figure 7.16, 7.17 and 7.18 we are able to get an understanding of how much of an impact changing the weight factor on the training data of the individual distributions will have on the performance of the overall system. By having multiple controllers contributing to the whole, the error of the individual controller will be less significant as shown.

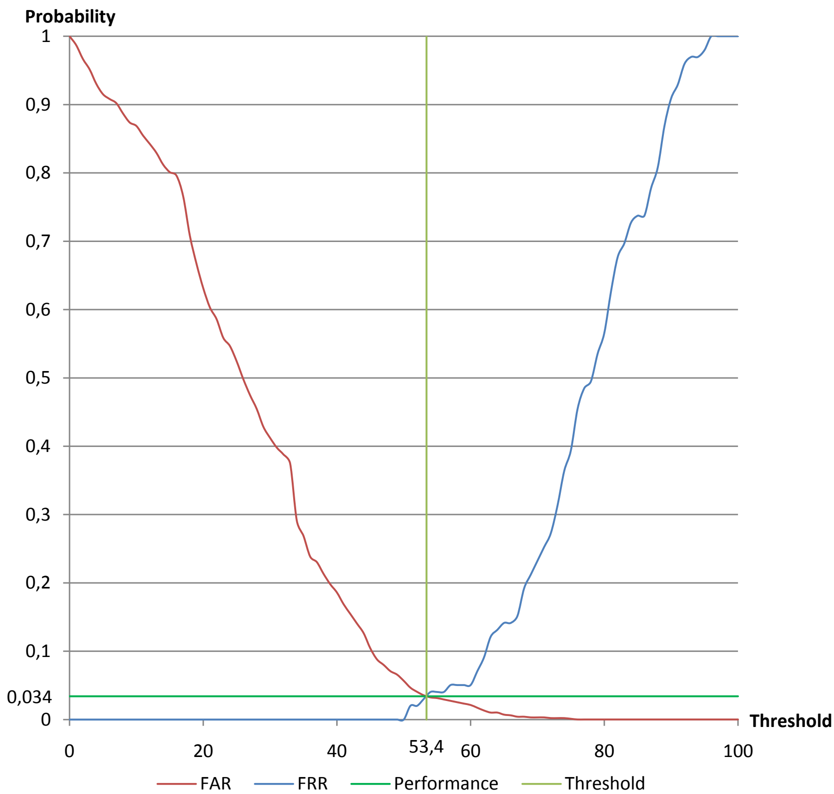

Figure 7.16: Performance of Master Controller with weight factor = 1.

Figure 7.16 shows a best EER of 0.034 - a correct classification occuring in 96.6 percent of the tests - using a threshold of 53.4.

This indicates that using a weight factor $w_f = 2$ and a threshold $t = 61.3$ yields

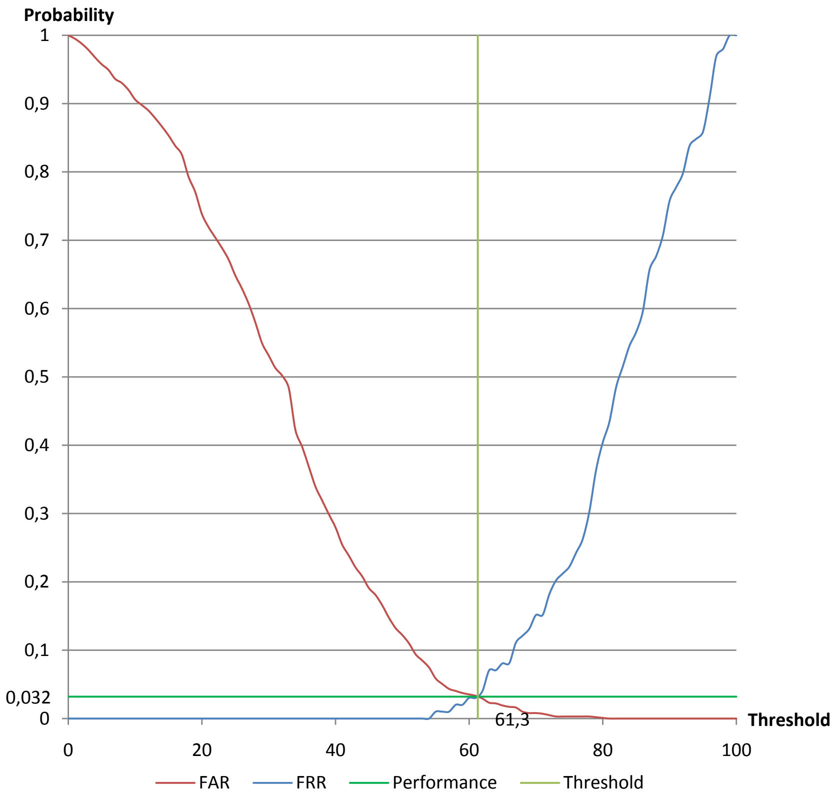

Figure 7.17: Performance of Master Controller with weight factor = 2.

Figure 7.17 shows a best EER of 0.032 - a correct classification occuring in 96.8 percent of the tests - using a threshold of 61.3.

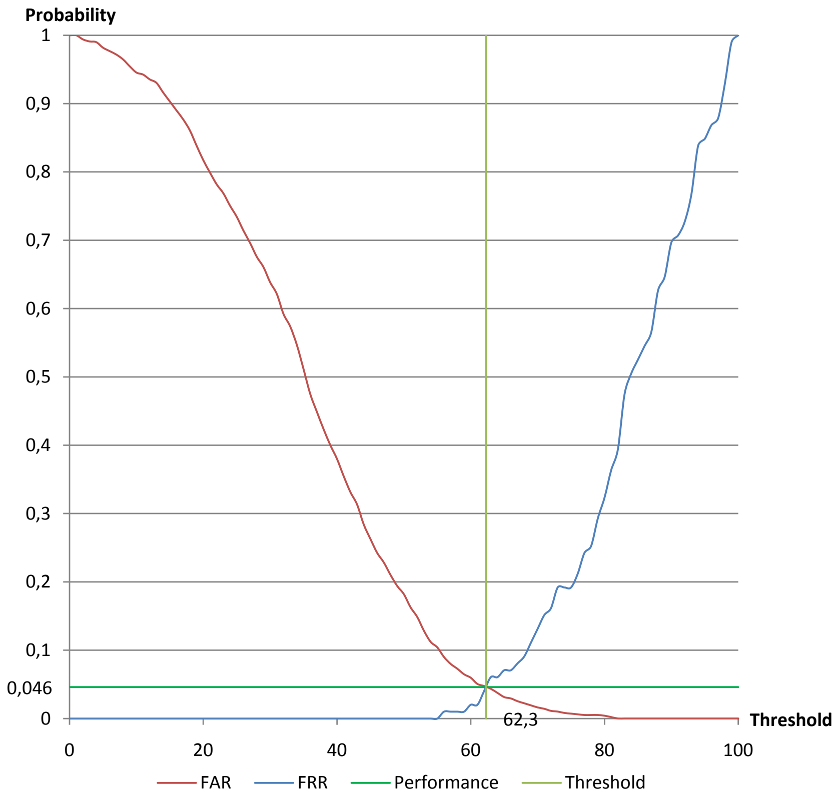

Figure 7.18: Performance of Master Controller with weight factor = 3.

Figure 7.18 shows a best EER of 0.046 - a correct classification occuring in 95.4 percent of the tests - using a threshold of 62.3.

better classification rates than using ($w_f = 1$, $t = 53.4$) or ($w_f = 3$, $t = 62.3$).

7.7 Test Conclusions

Looking at the result obtained during the performance tests performed on the different controller, a performance boost of 23.8 percentage points is observed when using the combined multi-modal match score for evaluating verification as opposed to uni-modal controllers. Using multiple modalities therefore increases the performance of the verification process of BioMIS, providing a more secure recognition.

Looking at the soft biometric controllers of the weight and height the performance is skewed. This is due to the fact that several samples for the weight controller and all the samples of the height controller are simulated. A greater variation in the samples in general will yield a more probable real world result. Using less correlated soft biometrics such as hair or eye colour, than weight and height (taller people tend to weigh more) might also give better results.

Looking at the primary controller, using the AAM to model the appearance of the face of the enrollees, the results are more accurate than those of the soft biometric controllers, because all of the images used were sampled during two separate photo sessions.

Chapter 8

Discussion & Further development

This section will contain some of the ideas that have been thought of, but due to time contraints could not be covered in this thesis, but to which BioMIS could benefit from.

8.1 Image Stream Acquisition

When receiving a stream of frames from a camera, some benefits could be gotten by using DirectShow as opposed to using OpenCV. At present the maximum resolution of the capture through OpenCV is 640-by-480 pixels, whereas the capture resolution through DirectShow is based on the driver provided by the camera vendor - in case of the Logitech Webcam Pro 9000 it has native 2MP Carl Zeiss optics, which can capture a resolution of 1600-by-1200 pixels. OpenCV would still be used for image processing. A higher resolution would retain a greater amount of information and detail, which could improve on the performance of the system.

8.2 Training of the Data Set

To maximize the performance and accuracy of the recognition system, add the login image and model search output to the training set, after a positive recognition. Small changes to a person's appearance, such as skin tone and aging, will then have less of an impact on the recognition model. Also adding a greater variety in facial expressions to the model, even adding different poses such as looking to the left and right, up and down as shown in Figure 8.1

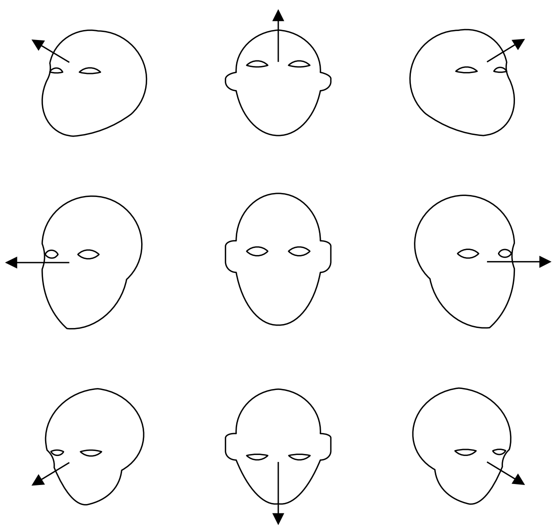

Figure 8.1: Using multiple poses in face direction.

We could also use a hierarchical model, where poses differentiating alot in shape from the frontal portrait, could be modelled using separeate models. An example could be using three appearance models, one for looking left, another for frontal protrait and a third for looking to the right. However this would require a method of distinguishing in which direction the subject is looking or all three appearance models could run in parallel, where the model with the hightest matching score detemines the output to the master controller.

8.3 Enrolment

When enrolling an individual into the recognition system, a lot of the time was used to change the lighting of the set, but also on the individual having to follow instructions about, in which direction they should turn their head, and people tend to overdo the directional poses (some images captured during the photo sessions are almost in full profile). To remedy some of these trends, Svend Vitting (Pallas Informatik) suggested using multiple cameras instead of just one, as was used during image data acquisition sessions.

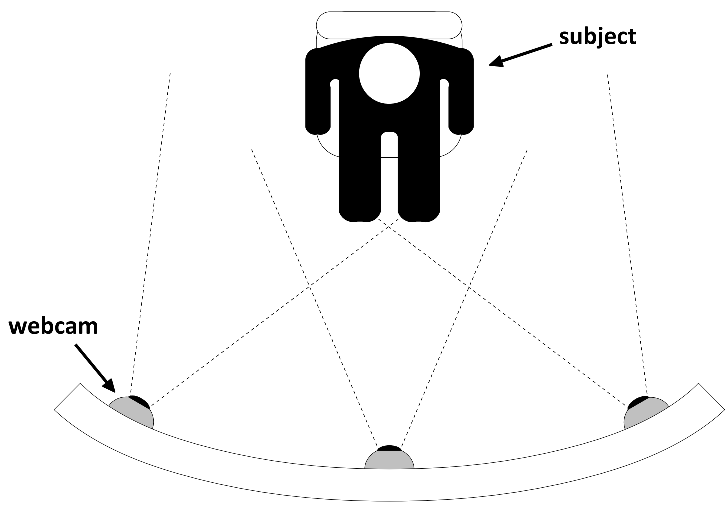

Figure 8.2: Multiple camera setup.

This way when capturing the training data of a person, the cameras that are off centre, will imitate the person turning their head in the desired direction. As such the images acquired will be more standardised, as the individual just needs to hold their head in a front facing position. This will also reduce the time it takes to acquire the images of each person (no need to wait for the individual to change the pose of their head).

8.4 Clustering

Using some type of clustring like *k-means* clustering to cluster an the parameter vectors of an individual might boost performance of the classification of the face controller. This might help cluster the parameter vectors according to face expression, light source or pose, if the before mentioned extension is used.

If using a global appearance model to characterise the enrollees of the Biometric Multi-factor Identification System, this might also help in separating the individuals stored in the database.

8.5 Larger Scale

The work done during the creation of this thesis has only been performed on fairly small sample sets and a study on a larger scale would be merrited.

Chapter 9

Conclusion

When a patient arrives at the Emergency Reception Ward at Frederiksberg Hospital the patient must be registered in their computer system, which can be done directly through a computer or using an electronic whiteboard, called "AMA-Tavlen" (the ER-Board). To access the ER-Board the staff uses a magnetic identification keycard.

As an extension of this authentication system the Biometric Multi-factor Identification System (BioMIS) has been devised. An implementation of this extension made in C# programming language has been developed. Based on the results given in the test section, it is demonstrated that the use of weak biometrics in connection with a primary biometric modality provides better performance than only using the primary biometric modality.

Results were obtained during the performance tests performed on the different controller, which show a performance boost of up to 23.8 percentage points, observed when using the combined multi-modal match score for evaluating verification as opposed to uni-modal controllers. Using multiple modalities therefore increases the performance of the verification process, providing a more secure recognition.

However the tests also indicate that the soft biometric controllers used in this project, of the weight and height, might give rise to a skew of the performance, due to the fact that several samples for the weight controller and all the samples of the height controller are simulated. Using real world samples will yield a more probable results. Furthermore using binary soft biometrics such as hair or eye colour, rather than probabilistic soft biometrics like weight and height, might also yield better results.

Looking at the performance of the face controller, using an AAM to model the appearance of the face of the enrollees, it should improve using a more diverse training set with a greater variety in face pose and facial expressions, along with more varied light sources. In regard to this, using a multiple camera setup during enrolment would help capture the training data of a person at different angles. This will also give a more uniform distribution in the training data as people, when ask to turn their head, each do this differently - some turn their head only slightly, while other turn their head for a profile view.

To solve the issue of automatic enrolment, we can use a global model of appearance, as this model spans the entirety of the training samples. This will give a better fit for model. The optimal fit values obtained this way can then be used as landmark annotation. (This can also be done for the individual models, but as they use a smaller training sample of faces and landmark annotations, the fitting performance is not as good). These landmarks can then be stored along with the acquired face image, and when training a new model the updated face and landmark database will contain more samples, which in turn will raise perform even more.

Also if using the global model as a verification model then extending the classification method of the face controller to use k-means clustering should help improve performance, as the training belonging to the same cluster would be more tightly grouped.

Appendix A

Appendix

A.1 Getting the AAM-API to work in a new project.

Following these steps should get the AAM-API to work with a VC++ 2005 project.

1. Create a new project.
2. Go to the property pages for that project.
3. Under ”Configuration Properties → General” set ”Use of MFC” under ”Project Defaults” to ”Use MFC in a Shared DLL” - requires that MFC 8.0 is installed on the system (Visual Studio 2005 Pro)
4. Also set the ”Character Set” to ”Use Multi-Byte Character Set”.
5. Next go to ”Configuration Properties → C/C++”.
6. Under ”Additional Include Directories” make sure that the AAM-API and sub dependencies (Diva and VisionSDK) are included, like so
   (a) <path to AAM-API>\inc

   (b) <path to AAM-API>\diva\inc
   (c) <path to VisionSDK>\inc

7. If you want to use multiband images (i.e. colour images), go to "Preprocessor" and edit the "Preprocessor Definitions" to include "AAM_3BAND".

8. Now go to "Configuration Properties → Linker".

9. Set the "Additional Library Directories" to include the AAM-API and sub dependencies (Diva and VisionSDK), like so

   (a) <path to AAM-API>\lib
   (b) <path to AAM-API>\diva\lib
   (c) <path to VisionSDK>\lib

10. Go to "Input" under "Linker".

11. Set the "Additional Dependencies" to include

   (a) <path to AAM-API>\lib\aam-apim.lib
   (or aam-apimDB.lib if you want debug info)

A.2 Compatibility Issues (.NET 4.0 & CLR/C++)

In order to use a CLR 2.0 *mixed mode assembly*, the app.config file needs to be modified to include:

```
<?xml version="1.0"?>
<configuration>
  <startup useLegacyV2RuntimeActivationPolicy="true">
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.0"/>
  </startup>
</configuration>
```

The key is the useLegacyV2RuntimeActivationPolicy flag. This causes the CLR to use the latest version (4.0) to load the mixed mode assembly. Without this, it will not work. Note that this only matters for mixed mode (C++/CLI) assemblies. All managed CLR 2 assemblies can be loaded without specifying this in app.config.

Bibliography

[1] http://en.wikipedia.org/wiki/Modular_programming.

[2] http://en.wikipedia.org/wiki/Pattern_recognition.

[3] http://opencv.willowgarage.com/wiki/.

[4] https://www.nemid.nu/.

[5] http://wiimotelib.codeplex.com/releases/view/21997.

[6] http://www.emgu.com/wiki/index.php/Main_Page.

[7] http://www.imm.dtu.dk/~aam/.

[8] P.N. Belhumeur, J.P. Hespanha, D.J. Kriegman, et al. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7):711–720, 1997.

[9] T. Christophersen. *Zero Interaction Multi-factor Authentication.* Informatik og Matematisk Modellering, Danmarks Tekniske Universitet, 2010.

[10] T.F. Cootes and C.J. Taylor. Statistical models of appearance for Computer Vision. *World Wide Web Publication, February*, 2001.

[11] T.F. Cootes and C.J. Taylor. Statistical models of appearance for medical image analysis and computer vision. In *Proc. SPIE Medical Imaging*, volume 4322, pages 236–248. Citeseer, 2001.

[12] K. Delac, M. Grgic, and S. Grgic. Independent comparative study of PCA, ICA, and LDA on the FERET data set. *International Journal of Imaging Systems and Technology*, 15(5):252, 2005.

[13] D. Delgado-Gomez, J. Fagertun, B. Ersbøll, F.M. Sukno, and A.F. Frangi. Similarity-based Fisherfaces. *Pattern Recognition Letters*, 30(12):1110–1116, 2009.

[14] A. K. Jain, S. C. Dass, and K. Nandakumar. Can soft biometric traits assist user recognition? In A. K. Jain & N. K. Ratha, editor, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 5404 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 561–572. August 2004.

[15] A. K. Jain, P. Flynn, and A. A. Ross. *Handbook of Biometrics*. Springer, 2008.

[16] Anil Jain, Sarat Dass, and Karthik Nandakumar. Soft biometric traits for personal recognition systems. In David Zhang and Anil Jain, editors, *Biometric Authentication*, volume 3072 of *Lecture Notes in Computer Science*, pages 1–40. Springer Berlin / Heidelberg, 2004.

[17] Anil Jain, Karthik Nandakumar, and Arun Ross. Score normalization in multimodal biometric systems. *Pattern Recognition*, 38(12):2270 – 2285, 2005.

[18] T. E. Larsen. *Biometric Solutions for Personal Identification*. Norwegian University of Science and Technology, Department of Telematics, 2008.

[19] A. Ross and A. K. Jain. Multimodal biometrics : An overview. In *Proceedings of 12th European Signal Processing Conference (EUSIPCO)*, pages 1221–1224. Citeseer, 2004.

[20] P. Viola and M.J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[21] L. Zhang, Q. Gao, and D. Zhang. Directional Independent Component Analysis with Tensor Representation. *In Proceeding of Computer Vision and Pattern Recognition*, 2008.