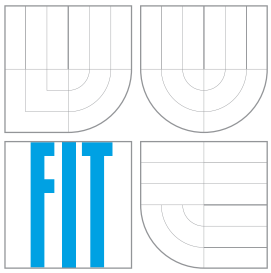
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

BIOMETRICKÉ ROZPOZNÁVÁNÍ 3D OBLIČEJE

BIOMETRIC RECOGNITION OF 3D FACES

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. ŠTĚPÁN MRÁČEK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. RADIM DVOŘÁK

BRNO 2010

Assignment

1. Study the biometric literature, especially oriented on biometric face recognition, based on 2D or 3D information. Learn the operating of 3D camera a4vision in the Biometric laboratory.
2. Propose a method for face recognition based on 3D information.
3. Practically implement the method proposed by you.
4. Realize experiments and discuss achieved results.

Abstrakt

Tato práce se zabývá rozpoznáváním 3D obličejů. Je zde popsán obecný biometrický systém a také konkrétní postupy používané při rozpoznávání 2D i 3D obličejů. Následně je navžena metoda pro rozpoznávání 3D obličejů. Algoritmus je vyvýjen a testován pomocí databáze Face Recognition Grand Challenge (FRGC). Během předzpracování jsou nalezeny význačné body v obličeji a následně je trojrozměrný model zarovnán do referenční polohy. Dále jsou vstupní data porovnávána s biometrickými šablonami uloženými v databázi, to je zajištěno využitím tří základních technik pro rozpoznávání obličejů – metoda eigenface (PCA), rozpoznávání založené na histogramu obličeje a rozpoznávání založené na anatomických rysech. Nakonec jsou jednotlivé metody spojeny do jednoho systému, jehož celková výsledná výkonnost převyšuje výkonnost jednotlivých použitých technik.

Abstract

This thesis is about biometric 3D face recognition. A general biometric system as well as specific techniques used in 2D and 3D face recognition are described. An automatic modular 3D face recognition method will be proposed. The algorithm is developed, tested and evaluated on the Face Recognition Grand Challenge (FRGC) database. During the preprocessing part, facial landmarks are located on the face surface and the three dimensional model is aligned to a predefined position. In the comparison module, the input probe scan is compared to the gallery template. There are three fundamental face recognition algorithms employed during the recognition pipeline – the eigenface method (PCA), the recognition using histogram-based features, and the recognition based on the anatomical-Bertillon features of the face. Finally the decision module fuses the scores provided by the utilized recognition techniques. The resulting performance is better than any of utilized recognition algorithms.

Klíčová slova

Biometrie, 3D obličej, Metody rozpoznávání obličejů, Eigenface, Detekce významných bodů v obličeji

Keywords

Biometrics, 3D face, Face recognition methods, Eigenface, Facial landmarks detection

Citace

Štěpán Mráček: Biometric Recognition of 3D Faces, diplomová práce, Brno, FIT VUT v Brně, 2010

Biometric Recognition of 3D Faces

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Radima Dvořáka

.....
Štěpán Mráček
May 30, 2010

Poděkování

I would like to thank my consultant Prof. Dr. Christoph Busch for valuable advices and consultations offered during the practical part of this thesis which took place at Gjøvik University College. The Face Recogntion Grand Challange dataset was offered by The Norwegian Information Security laboratory (NISlab).

Contents

Chapter 1

Introduction

Face recognition is one of the most used biometric techniques. In everyday life, we recognize other people by their faces. We are able to localize a face in a very large and complicated scene. Also the detection of anatomical features, like nose, eyes and mouth position within the face, doesn't pose us difficulties. Furthermore, we can recognize faces from various angles, even if face expressions are present or a part of a face is covered. Many activities that we are doing completely automatically with no effort, become quite difficult if we try to describe this process mathematically.

Nevertheless, a lot of research has been done in the area of the biometric face recognition, especially in the three-dimensional recognition in recent years. The face biometric has become together with fingerprints a part of biometric passports in The European Union and all member states of the ICAO (International Civil Aviation Organization) [1].

The biometric face recognition, which is the main focus of this thesis, has wide application in practice, e.g. the biometric passports, as was mentioned above, or in access control systems. Because of its nature, which is very similar to the way we usually recognize each other, it is very well accepted by users. No special activity is required by data subject and the recognition process is non-intrusive, which means that the data subject is not in the direct contact with the sensor.

1.1 Thesis structure

This thesis is about the biometric face recognition and all connected matters. In the first chapter, base terms related to the biometrics are explained and a general biometric system is described. The methods of evaluating the biometric system performance is also present in the first chapter.

The second chapter brings the overview of face recognition techniques. It is divided into two independent parts – the two-dimensional and the three-dimensional face recognition. Although there are lots of similarities between these two approaches, different methods can be used due to the form of the data.

The third chapter describes the A4 Vision Enrollment Station – a device for enrolling data subjects into the three-dimensional biometric system. Due to the impossibility of direct access to the three-dimensional data in stored scans, the usage of this camera has emerged as problematic for the purpose of this thesis, so the Gavab and FRGC face databases, that is used for testing the recognition algorithm, are described in this chapter too.

The proposal of the recognition algorithm is in the fourth chapter and the implementa-

Table 1.1: Examples of physiological and behavioral biometric characteristics

Physiological characteristics	Behavioral characteristics
Fingerprints	Voice
Face	Walk
Iris	Lips motions
DNA	Signature dynamics

tion of proposed algorithm is described in the fifth chapter.

Implemented system is evaluated on the FRGC database in the sixth chapter.

1.2 Biometrics

In this section, biometrics and related terms will be explained. The definitions explained here are from the standard Harmonized biometric vocabulary [2], the paper [3] by Jain and Ross and the first chapter in their book Handbook of Biometrics [4].

Biometrics refers to methods for uniquely recognizing individuals based upon one or more intrinsic physiological or behavioral characteristics. The physiological characteristics, sometimes called anatomical characteristics, refers to the characteristics that are always present on data subject's action independently. Biometric methods based on the physical characteristics are called static, while biometric methods based on the behavioral characteristics are called dynamic. Dynamic characteristics are connected with some data subject's action. Each capture in different time can provide different results. In Table 1.1, are some examples of the physiological and the behavioral characteristics.

There are some more terms related with the biometrics and the biometric recognition. A short list with explanation of these terms is provided here:

Identity – The identity of an individual may be viewed as the information associated with that person in a particular identity management system. An individual can have more than one identity.

Identification – The identification is the process when the biometric system recognizes an individual by comparing him with all templates in the database. The result of the identification is the data subject's identity or "not recognized".

Verification – On the other hand, the verification is the process when the data subject provides his claimed identity and the system has to decide if it is true or not.

Every biometric feature, that is used in some biometric system, should provide these characteristics:

Universality – Every person should have the characteristic.

Uniqueness – No two persons should be the same in terms of the characteristic.

Permanence – The characteristic should be time invariant.

Collectability – The characteristic should be measured quantitatively.

These issues should be also considered when implementing the biometric system:

Performance – Refers to the achievable identification accuracy.

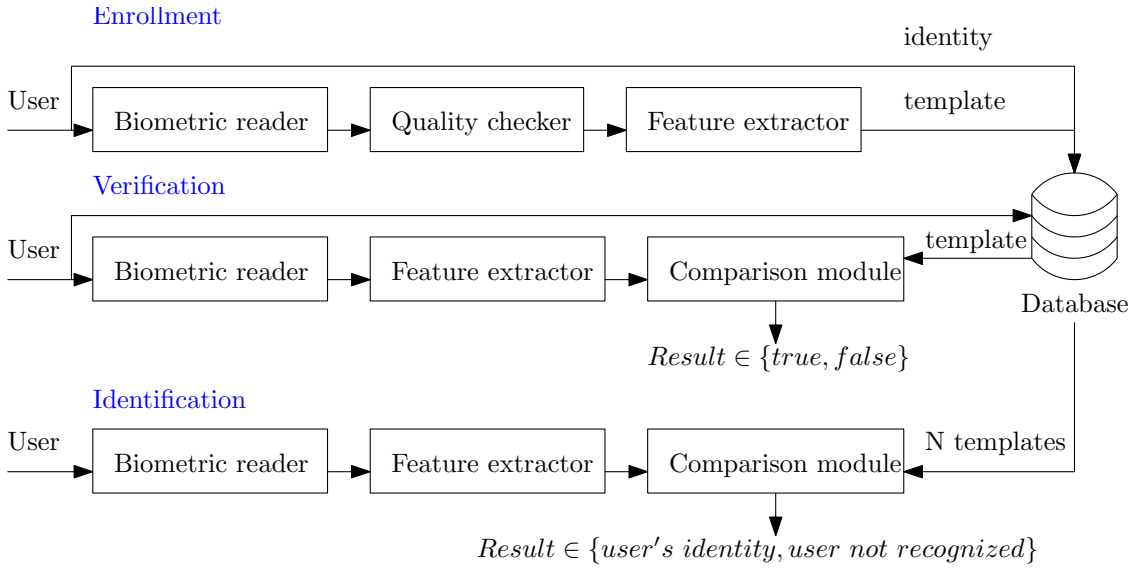


Figure 1.1: Generic biometric system

Acceptability – Indicates to what extent people are willing to accept the biometric system.

Circumvention – Refers to how easy it is to fool the system by fraudulent techniques.

The suitability of the biometric characteristic is often expressed with the terms intra-class variance and inter-class variance. The intra-class variance refers to the diversity between individual scans of the same person, while the inter-class varinace refers to the diversity between different persons. It is good to choose the biometric characteristic which has the inter-class variance as high as possible and, on the contrary, the intra-class variance as low as possible.

A generic biometric system consists of two main parts – the enrollment module and the identification/verification module. The enrollment module is for registering new data subjects to the system. During this process, a data subject is scanned by the biometric reader. If the scan satisfies the defined quality the features are extracted and stored to the database as a template subsequently.

The identification/verification module scans the data subject, extracts the features and compares them with other templates in the database. In the case of verification, data subject provides a claim about his identity and the biometric system has to decide if it is true or not. On the other hand, in the case of identification, the recognition system has to decide if the data subject is registered and if so which template in the database belongs to him. A generic biometric system is illustrated in Figure 1.1.

Comparing a template with a features extracted from a scan provided by a data subject produces the comparison score denoting how are the template and the extracted features similar to each other. The decision whether the data subject is accepted or not is based on the threshold, which determines the border between the acceptance and rejection.

1.3 Evaluating Performance of Biometric systems

One of the most important property of a biometric systems is how successful in recognition it is. There are two main errors that biometric system can make – false acceptance and false rejection. In the case of access control, where the biometric system has to control the access to some area or resources, false acceptance means that an intruder has been confused with some registered person and has been admitted. On the other hand, false rejection is the case when registered person is rejected.

The decision if some person is accepted or not is based on the comparison score, obtained during the comparison process, and the given threshold. If the *score* denotes distance between the gallery and the probe scan, the decision algorithm is as follows:

$score \leftarrow get\_comparison\_score(probe, gallery)$
if $score \geq Threshold$ **then**
  Reject
else
  Accept
end if

Four situations might occur:

True acceptance – The genuine person is truly recognized.

True rejection – The impostor is truly recognized.

False acceptance – The impostor is admitted.

False rejection – The genuine person is not recognized and therefore rejected.

The goal of every biometric system is to be as secure as possible. This means to minimalize the false rejection and false acceptance cases. The False acceptance rate (FAR) indicates what proportion of attempts resulted in a false recognition.

$$\text{FAR} = \frac{\sum \text{different measures classified as the same}}{\sum \text{measures of different persons or instances}} \tag{1.1}$$

The false rejection rate (FRR) indicates what percentage of attempts by legitimate users are incorrectly rejected.

$$\text{FRR} = \frac{\sum \text{misclassified measures of the same person or instance}}{\sum \text{measures of the same person or instance}} \tag{1.2}$$

The FAR and FRR are joined together by the threshold that decides if the data subject is accepted or not. However, to raise the threshold leads to the more secure system, where the impostors are refused, but genuine data subjects are sometimes refused too. On the other hand, the lower threshold leads to the comfortable system, where most genuine data subjects are accepted and sometimes impostors too.

The Equal error rate (EER) is the value where the FRR and FAR for given threshold are equal. It is often used as a criteria for evaluating performance of the biometric systems. The lower the value is, the better the system, compared to another is. The relation between the FAR and FRR is illustrated in Figure 1.2. The decision if the data subject is accepted or not is stricly based on the retrieved comparison score and the given threshold.

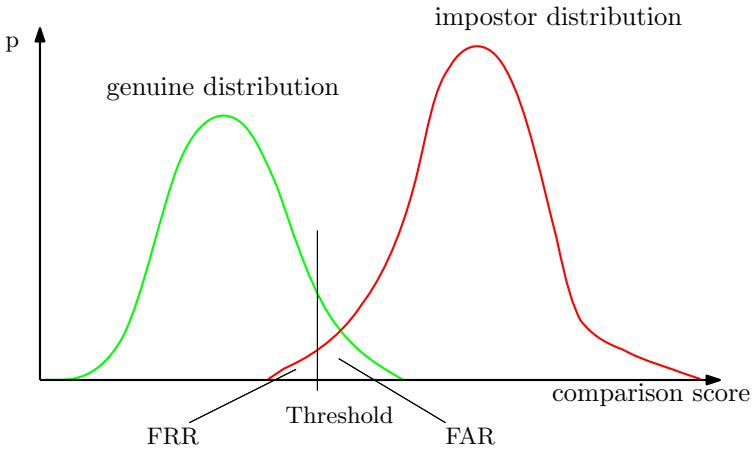


Figure 1.2: False acceptance rate and false rejection rate

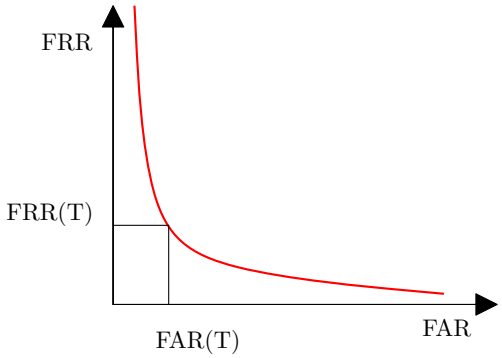


Figure 1.3: Detection error trade-off curve

The relation between the FAR and FRR values at the different thresholds depicted as a curve is reffered to as Detection Error Trade-off (DET). The example of DET curve is in Figure 1.3.

There are some more terms related with the evaluating of the biometric systems:

FTA – The failure to acquire rate is the portion of situations when the system is unable to acquire the data from the data subject. Its value is connected with the biometric sensor and its quality checker, especially with the defined quality that the each scan should have.

FTE – The failure to enroll rate is the portion of situations when the system is unable to generate the template from the input data.

FNMR – The false non-match rate refers to the portion of the false rejected persons. Contrary to the FRR the FNMR doesn't include attempts that have been unsuccessful before the comparison has started.

FMR – The false match rate refers to the portion of the false accepted persons. Like with the FNMR, unsuccessful attempts before the comparison has started are not counted.

While the FAR and FMR are related to the system performance, FMR and FNMR describe the algorithm performance.

1.4 Face Recognition

The biometric recognition of faces includes the methods and algorithms of finding the face within two-dimensional images and three-dimensional data, locating face features and the recognition itself. The input of the two-dimensional face recognition are ordinary photographs, while the three-dimensional face recognition is performed on the spatial data.

Although the three-dimensional face recognition may provide better results than the two-dimensional approach [5], special device for acquiring scans from data subject should be purchased. This fact leads to the much higher acquisition cost.

The face recognition is, along with fingerprint and iris recognition, one of the most used biometric techniques. It is well accepted by users due to its noninvasive character.

A lot of research work that deals with all parts of the face recognition has been done, but many problems have not been resolved sufficiently. Some basic tasks for human, like the localization of the nose tip, are not easy for computers as well. Another difficulties are related with incomplete input data. Many recognition algorithms fail when some part of data subject's face is covered or some facial expressions are present.

Chapter 2

Face Recognition Techniques Overview

In this chapter, the overview of face recognition methods will be provided. At first, some essential terms related to the face recognition will be explained. Then various techniques of two-dimensional and three-dimensional face recognition will be described.

2.1 Face Recognition Difficulties

The biggest challenge of the biometric face recognition is to deal with relatively big intra-class variation, which is related with many factors, mostly by various lighting conditions, face orientation and facial expressions. Light direction, color and intensity have negative influence on the performance of two-dimensional face recognition, where the recognition is performed on photographs obtained by commonly used cameras. Head orientation is also big problem in two-dimensional recognition.

Facial expressions affect both two-dimensional and three-dimensional recognition. Various techniques that deal with facial expressions have been invented. These techniques are described further in text.

Two pictures with different ligting conditions of the same person are shown in Figure 2.1 (taken from [6]).

2.2 Theoretical Background

Face recognition is in principle a pattern recognition. Each face is presented as a vector that could be located in a multi-dimensional face space, e.g. in the three-dimensional face recognition a face could be represented as a range image with resolution 150x100 pixels.

Figure 2.1: Different lighting conditions

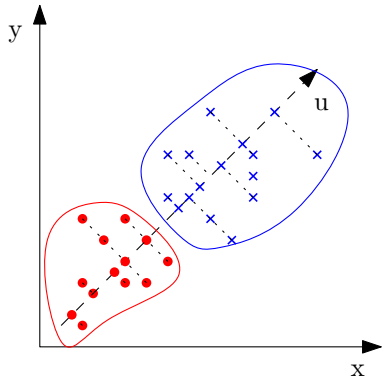


Figure 2.2: Principal component analysis

This produces 15 000-dimensional space in which each face scan is stored. Face scans of the same person should be situated close to each other, while face scans of another person are further. Calculating distances between the face scans in this multi-dimensional space, thus comparing faces, is very time consuming due to multi-dimensionality of the space. Therefore, various techniques that decrease number of dimensions were invented. The best known are the principal component analysis (PCA) and the linear discriminant analysis (LDA).

2.2.1 Principal Component Analysis

Principal component analysis was first introduced by Karl Pearson [7] and covers mathematical methods which reduce the number of dimensions of given multi-dimensional space. The dimensionality reduction is based on the data distribution. The first principal component describes best the data in a minimum-squared-error sense. Other succeeding components describes as much of the remaining variability as possible.

Model situation is shown in Figure 2.2. Two-dimensional space containing data in two classes is reduced to the one-dimensional space. Every point is projected to the new dimension u. Classification is then based on the position of projected point on the dimension u.

The calculation of the principal components is unsupervised learning. The class membership is not taken into account during learning. The principal component analysis seeks for direction in which the data varies the most. This could cause in some cases wrong classification. This problem is illustrated in Figure 2.3.

2.2.2 Linear Discriminant Analysis

Linear discriminant analysis (LDA), introduced by Ronald Aylmer Fisher [8], is an example of supervised learning. The class membership is taken into account during learning. LDA seeks for vectors that provide the best discrimation between classes after the projection. Therefore, the LDA is applicable to the classification problems where PCA fails (Figure 2.4).

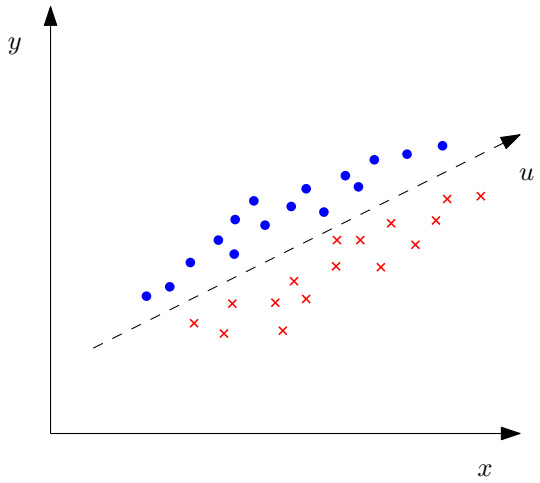


Figure 2.3: Wrong class separation with principal component analysis

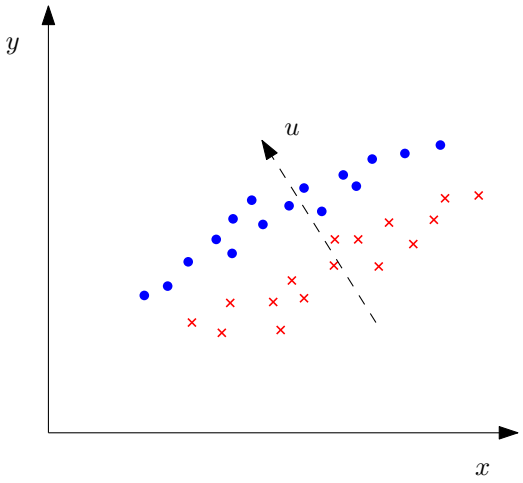


Figure 2.4: Linear Discriminant Analysis

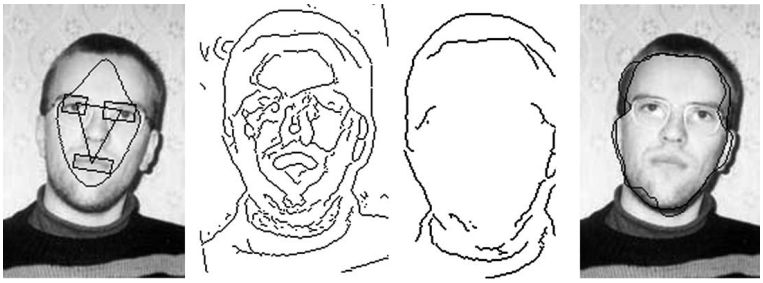


Figure 2.5: Face contour localization

2.3 Classification of Face Recognition Methods

The two-dimensional face recognition as well as the three-dimensional approach can be performed as holistic or feature based. Holistic face recognition takes face as a whole with no semantic analysis or landmark detection. Faces are typicaly represented as vectors on which some method of dimensionality reduction, like PCA or LDA, is performed. On the contrary, the feature-based methods are focused on the detection of facial landmarks and their size, location and mutual position. The comparison score for recognition, then, could be acquired by a calculation of the weighted sum of distances between corresponding face features.

2.4 2D Face Recognition

In this chapter various approaches of the two-dimensional face recognition will be described. First some methods of locating faces and facial features within images will be mentioned. Then two major two-dimensional face recognition techniques will be described – eigenface and fisherface. Both of them are based on projection of multidimensional space into face subspace where is then performed comparison by measuring distance between two projected faces.

Face features and face detection is used as preliminary process of face recognition [5]. The eyes localization method is used to align the two-dimensional face images to some general position and size.

2.4.1 Finding Face Features

The detection of a face and facial features is an important task when creating the automatical face analysis and processing systems [9]. The results of the detection could be used for example in face recognition systems.

Perlibakas [9] proposed a method based on the mathematical morphology and the variational calculus for the detection of face contour in grayscale images. First the localization of an ellipse and contour based methods are used to locate face in the image. Then inside each ellipse, mathematical morphology is used to find eyes and lips. Face contour is detected using active contour model by minimizing spline, whose energy depends on its form and position in the image. The results of this process can be seen in Figure 2.5 (taken from [9]), where the eyes and lips localization is shown, then the edge map is calculated and finally its energy is minimized to calculate face contour.

Craw et al. [10] designed a system aimed on locating 40 various feature points within greyscale images. The system consists of two distinct parts – recognition modules and a control structure. The recognition modules are responsible for locating individual parts of the face, while the control structure, driven by a high level hypothesis about the location of the face, invokes the feature finding modules in order to support or refute its current hypothesis.

Zhi-fang Liu et el. [11] detect fatial features from color images. First, the input is transformed to YCrCb color space. Then skin color pixels are detected from chrominance components Cr and Cb. Within this region, the edge detection algorithm is applied and principal component analysis is used to locate eyes. Nose and mouth detection is then significantly simplified, when the position of eyes is known. Geometrical modeling is used to locate them.

2.4.2 Eigenface Method

The eigenface method is an example of application of the principal component analysis. It is a holistic face recognition method which takes grayscale photographs of persons that are normalized with respect to size and resolution represented as vectors. It was introduced by M. Turk and A. Pentland [12].

First the mean face from the set of training images is calculated. We take the set of P training images $\Gamma_1, \Gamma_2, \ldots, \Gamma_P$ and the mean face Ψ is calculated:

$$\Psi = \frac{1}{P} \sum_{i=1}^{P} \Gamma_i \tag{2.1}$$

Then the mean face Ψ is substracted from each training image Γ_i

$$\Phi_i = \Gamma_i - \Psi \tag{2.2}$$

After that the covariance matrix C is constructed

$$C = A\,A^T = [\Phi_1 \Phi_2 \ldots \Phi_M]\,[\Phi_1 \Phi_2 \ldots \Phi_M]^T \tag{2.3}$$

where A stands for a matrix where each column i contains a corresponding vector Φ_i. A^T stands for transposed matrix A.

The next step is the calculation of the eigenvalues and eigenvectors of the covariance matrix. This could be achieved by standard linear algebra methods [13]. Given a matrix M, a non-zero vector $\mathbf{x}$ is defined to be an eigenvector of the matrix if it satisfies the eigenvalue equation

$$M\,\mathbf{x} = \lambda\,\mathbf{x} \tag{2.4}$$

for some scalar λ. In this situation, the scalar λ is called an eigenvalue of M corresponding to the eigenvector $\mathbf{x}$ [14].

However, the covariance matrix is very large and the computation of it's eigenvectors and eigenvalues would be time and memory consuming. If the amount of training images P is sufficiently smaller then size n of training images, eigenvectors and eigenvalues could be retrieved from matrix C' [15].

$$C' = A^T\,A \tag{2.5}$$

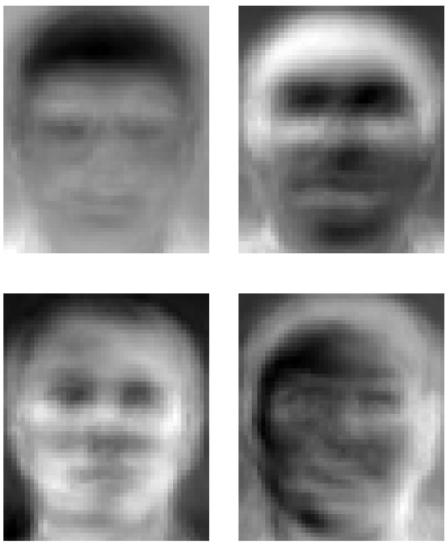


Figure 2.6: Eigenfaces obtained from AT&T face database.

The size of matrix C' is determined by the size of the training set and it is $P \times P$. The first P sorted eigenvalues of matrix C' are also eigenvalues of matrix C. The eigenvectors of matrix C are calculated by multiplying matrix A^T by matrix V'. V' is the matrix containing in each row one eigenvector v' of the matrix C'.

$$V = A^T V' = A^T \begin{bmatrix} v'_{11} & v'_{12} & \cdots & v'_{1P} \\ v'_{21} & v'_{22} & \cdots & v'_{2P} \\ \vdots & \vdots & \ddots & \vdots \\ v'_{P1} & v'_{P2} & \cdots & v'_{PP} \end{bmatrix} \tag{2.6}$$

The resulting matrix V contains an eigenvector of the covariance matrix C in each row. These eigenvectors define a set of mutually orthogonal axes within facial space, along which there is the most variance. The corresponding eigenvalues represent the degree of variance along these axes. In Figure 2.6 there are displayed some eigenvectors from AT&T face database (taken from [6]). Due to the likeness to faces, Turk and Pentland refer them as eigenfaces [5].

Projection of the facial image Γ to the face space is as follows: first the mean face Ψ is subtracted from input face Γ. Then each component ω_i is calculated by multiplying the corresponding eigenvector v_i by difference of the input face and the mean face.

$$\omega_i = v_i \left(\Gamma - \Psi\right) \tag{2.7}$$

The comparison of two faces in this face space is performed by calculation of the distance between these two faces. Various distance calculation could be used [5] on projected face images Ω_A and Ω_B, such as Euclidian distance

$$d(\Omega_A, \Omega_B) = \sqrt{\sum_{i=1}^{P} (\omega_{Ai} - \omega_{Bi})^2} \tag{2.8}$$

or the city block distance

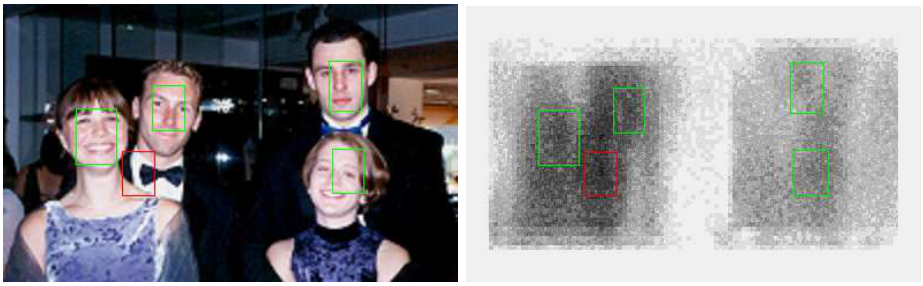


Figure 2.7: Locating face using eigenface method

$$d(\Omega_A, \Omega_B) = \sum_{i=1}^{P} |\omega_{Ai} - \omega_{Bi}| \tag{2.9}$$

and the cosine distance.

$$d(\Omega_A, \Omega_B) = 1 - \frac{\Omega_A^T \Omega_B}{\|\Omega_A\| \, \|\Omega_B\|} \tag{2.10}$$

Achieved Results

If the eigenface method is applied on the pictures with various lighting conditions, much of the variation from one image to the next is due to illumination changes. It has been suggested that by discarding the three most significant principal components, the variation due to lighting is reduced. The assumption is that if the first principal components capture the variation due to lighting, then better clustering of projected samples is achieved by ignoring them [16].

Locating Face in the Picture

Turk and Pentland [17] also proposed the method for locating the face in the image. This process is sometimes performed before the localization of eyes or other parts of the face, because it reduces the search space. Searching for face lies on passing a window over the image. This window could be resized or rotated in order to detect faces with different sizes or rotations. Selected window is projected into the face space and then back to image. If the window contains face, then the backward projection of the window should be similar to the original content of the window. The likeness of face similarity is then expressed as grayscale value and assigned to the center pixel of each window. Darker values mean more face similarity, brighter values mean regions with no face similarity. The example of resulting image is shown in Figure 2.7. On the left side, there is the original image, on the right side, there is the associated face likeness map. The picture was taken from [5]. Areas with higher density of face likeness are considered to be faces.

2.4.3 Fisherface Method

Fisherface method is the combination of principal component analysis and linear discriminant analysis. PCA is used to compute the face subspace in which the variance is maximized, while LDA takes advantage of inner-class information. The method was introduced by Belhumeur et al. [18].

To gain advantage of inner-class variation, a training set cointaining multiple images of the same persons is needed. Training set τ is defined as

$$\tau = \{X_1, X_2, \ldots, X_c\} \tag{2.11}$$

where $X_i = \{\Gamma_1, \Gamma_2, \ldots\}$. Γ_j stands for individual picture of the same person with different facial expressions or lighting conditions.

From this training set, three scatter matricies describing inner-class, intra-class and total variation of space are calculated. Total distribution matrix S_T describes the variance within all images in the training set.

$$S_T = \sum_{i=1}^{M} (\Gamma_i - \Psi)\,(\Gamma_i - \Psi)^T \tag{2.12}$$

where $\Gamma_1, \ldots, \Gamma_M$ stand for all face images from the training set and Ψ represents mean face image.

Then, the inter-class (between-class) distribution matrix S_B is calculated. This matrix describes variation between each person from the training set.

$$S_B = \sum_{i=1}^{C} |X_i|(\Psi_i - \Psi)\,(\Psi_i - \Psi)^T \tag{2.13}$$

where Ψ_i is the average of each person X_i

$$\Psi_i = \frac{1}{|X_i|} \sum_{\Gamma_i \in X_i} \Gamma_i \tag{2.14}$$

Finally the intra-class (within-class) distribution matrix S_W describing variation inside the specific class is calculated.

$$S_W = \sum_{i=1}^{C} \sum_{x_k \in X_i} (x_k - \Psi_i)\,(x_k - \Psi_i)^T \tag{2.15}$$

After that, the principal component analysis is applied to the total distribution matrix. Matrix U_{pca} contains top $M - C$ eigenvectors of matrix S_T.

$$U_{pca} = \arg\max_U |U^T\, S_T\, U| \tag{2.16}$$

$$U_{pca} = [\mathbf{w}_1\, \mathbf{w}_2\, \ldots\, \mathbf{w}_{M-C}] \tag{2.17}$$

The projection matrix U_{pca} is then used to reduce the dimensionality of intra-class and inter-class matricies S_W and S_B. First the projection matrix U_{fld} containing top $C - 1$ eigenvectors of the reduced scatter matricies is computed.

$$U_{fld} = \arg\max_U \left(\frac{|U^T\, U_{pca}^T\, S_B\, U_{pca}\, U|}{|U^T\, U_{pca}^T\, S_W\, U_{pca}\, U|} \right) \tag{2.18}$$

In the end, the final projection matrix U_{ff}, that transforms an input image into the face space, is computed.

$$U_{ff}^T = U_{fld}^T\, U_{pca}^T \tag{2.19}$$

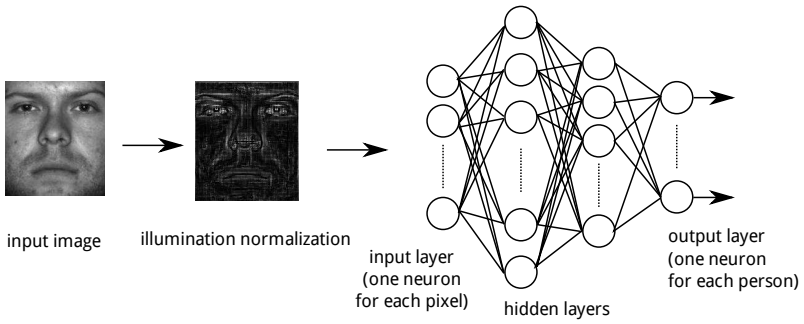


Figure 2.8: Face recognition with feedforward neural network

The resulting transformation matrix contains an eigenvector in each row, which is called fisherface [18] in this context. Transformation of the input image to the reduced face space is similar to the transformation described in equation 2.7.

It has been shown, that fisherface method provides better results than eigenface method [18, 5]. This is caused mainly because fisherface methods takes advantage of inter-class variation. To achieve significantly better results compared to eigenface method, good training set containing various facial expression and lighting condition is required.

2.4.4 Using Neural Networks

Artificial neural networks are applicable for both face recognition and feature extraction [19].

Bild et al. [20] applied a feedforward neural network in order to recognise a set of 39 people with 9 poses and under 64 illumination conditions. Two preprocessing techniques were applied on the input image set, edge detection and illumination normalization. The neural network consisted of two hidden layers and variation of backpropagation learning algorithm was used. Output layer were constructed from 39 neurons (one neuron for each enrolled person). Although the final results of recognition after the application of the illumination normalization filter were 100%, the training of the network required approximetly 24 hours. A diagram of described recognition process is in Figure 2.8. Images from [20] are used in the diagram.

Slightly different approach were chosen by Intrator et al. [21]. Automatic detection of eyes and mouth was followed by spatial normalization of images. Then, a hybrid artificial neural network involving both unsupervised and supervised learning were used for classification. The unsupervised training portion was focused on finding the facial features and the supervised portion for reducing classification error. Intrator et al. applied the recognition mechanism on original images containing background and cropped images with face only. Success ration were better when images including background were used. This was caused mainly because different subjects were photographed at different places. Thus, the background such as a corner of a blackboard, picture on the wall etc., was common to the same subject but not to all of them [21].

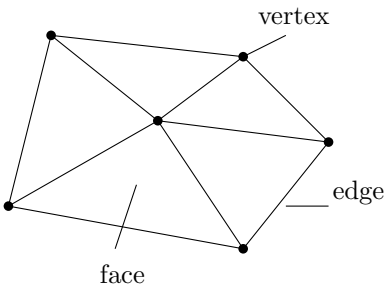


Figure 2.9: Mesh representation of the three-dimensional model

2.5 3D Face Recognition

In this section, the various approaches of the three-dimensional face recognition is described. First, the advantages of the three-dimensional face recognition compared to the two-dimensional face recognition is mentioned. The relative head rotation independence in the three-dimensional face recognition is achieved by the normalization of the face orientation. Several techniques that handle face orientation is mentioned.

Then the three-dimensional adaptation of eigenface and fisherface methods is described. Although these two methods were originally proposed for the two-dimensional recognition, they can be applied to the three-dimensional data as well. At the end of the section, some purely three-dimensional methods, like the three-dimensional model based face recognition, is described.

As were mentioned above, the three-dimensional face recognition brings several advantages compared to the two-dimensional recognition. Mainly because it provides new facilities of discrimination, like for example analysing the curvature of a face, measuring nose depth, comparing various facial curves etc.

Another advantage of the three-dimensional face recognition is related to the data capture technique. Many three-dimensional cameras operates in the infrared spectrum [22] that is independent to lighting conditions, such that direction of light and shadows on the face doesn't negatively affect face recognition.

2.5.1 Representations of the Three-Dimensional Face Model

A three-dimensional face model could be presented in various forms. Most often pointclouds [23], meshes [24] and range images [5] are used. Although all of these representations could be mutually converted to other representations, face recognition technique depends on the form of a three-dimensional model.

Almost all of freely available three-dimensional face databases use mesh representation for the stored face models [25]. First, the coordinates of each vertex are defined. Then polygons (faces) are defined as a list of their vertices.

The pointcloud representation is just simplification of the polygon mesh, because a model is defined only by it's vertices. Backward conversion to the polygon mesh can be achieved by using the marching triangles or Delaunay triangulation algorithm.

Next common used representation of three-dimensional models is a range image. Because of their implementation, range images are sometimes called depthmaps. It is an array where each element represents the distance from camera, therefore this array could be stored as greyscale bitmap images. Range images are not universal for all types of three-

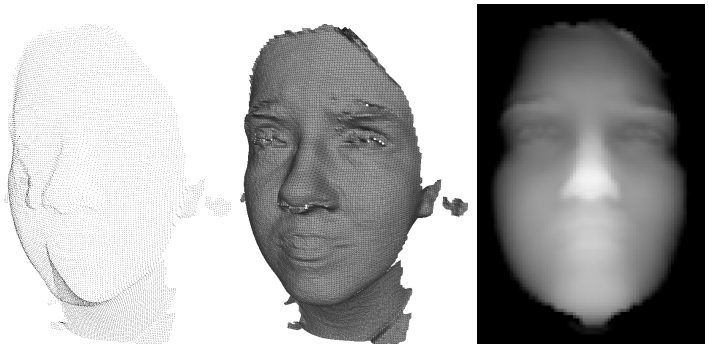


Figure 2.10: Pointcloud, mesh and range image representations of the same face

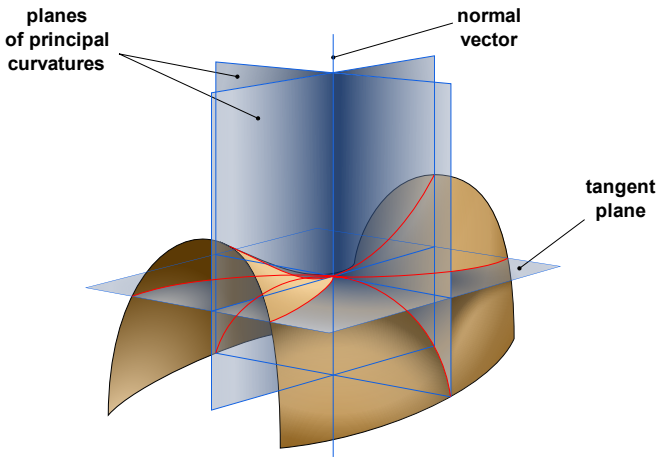


Figure 2.11: Principal curvatures

dimensional models. Mainly because they can't store information about points that are hidden by other points. Using range image representation on three-dimensional faces is not affected by this limitation, because the frontal view of a face doesn't contain much points that are hidden by other parts of face.

All three representations of the same face (from GavabDB face database [25]) are in Figure 2.10.

2.5.2 Curvature Analysis

Since the three-dimensional face model can be described as a surface in the three-dimensional space, curvature analysis can be applied on it. Curvature is the amount by which surface derivates from being flat.

Assume a parametric curve $\gamma(S)$, where S is a parameter which determinines tangent vector $T(S)$ and normal vector $N(S)$ at each point of the curve. This parameter also determines curvature $k(S)$ and radius of curvature $R(S) = \frac{1}{k(S)}$.

For the analysis of the face surface, the principal curvatures are important. At a given point of a surface they measure, how the surface bends by different amounts in different directions at that point. An illustration is in Figure 2.11 (taken from Wikipedia [26]). Planes of principal curvatures k_1 and k_2 and the tangent plane are orthogonal.

To express a surface curvature characteristics with only one value at each point on the

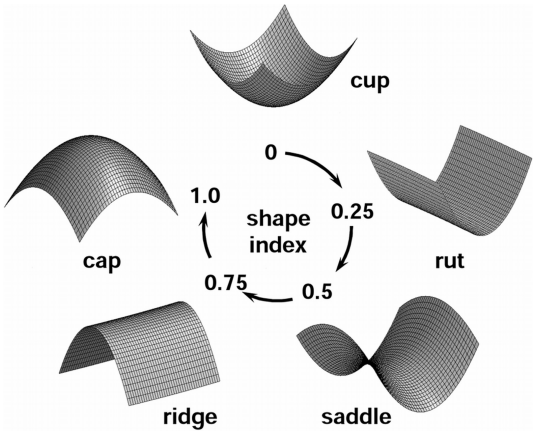


Figure 2.12: Shape index shapes

surface, several options are available.

Gaussian curvature K of a point on a surface is defined as the product of principal curvatures k_1 and k_2.

$$K = k_1 \, k_2 \tag{2.20}$$

Mean curvature H is the average of principal curvatures k_1 and k_2.

$$H = \frac{1}{2}(k_1 + k_2) \tag{2.21}$$

Shape index S is for the classification of the surface into categories. See Figure 2.12 [26].

$$S = \frac{1}{2} - \frac{1}{\pi} \tan^{-1} \frac{k_1 + k_2}{k1 - k2} \tag{2.22}$$

where the principal component k_1 is greater than k_2.

In Figure 2.13 Gaussian curvature, mean curvature, and shape index of the same face are shown. Pits and peaks and other types of the shape can be found by comparison of the signs of the Gaussian and mean curvature [27], see table below. The pits and peaks points of the face are shown in also in Figure 2.13.

K / H	< 0	$= 0$	> 0
< 0	saddle ridge	minimal	sadle valley
$= 0$	ridge	flat	valley
> 0	peak	(none)	pit

2.5.3 Facial Landmarks Detection

Detecting the facial landmarks from the three-dimensional data can't be performed using the same algorithms as on two-dimensional data. Mainly because the two-dimensional landmark detection lies on analysing the color space of the input face picture, which is not present in the raw three-dimensional data.

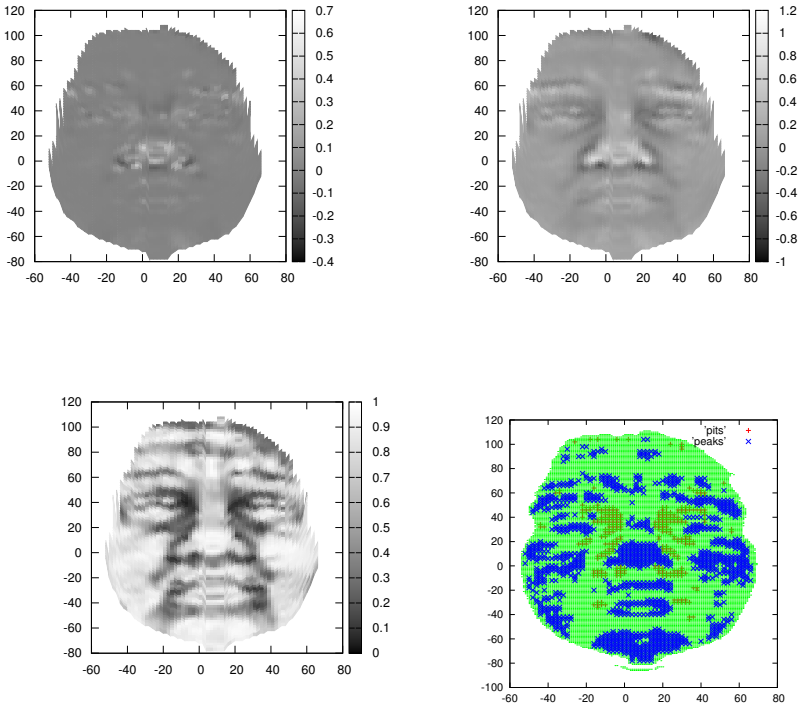


Figure 2.13: Gaussian curvature, mean curvature, shape index and pits and peaks on the face surface

In this section several techniques of obtaining the facial landmarks from the three-dimensional data will be described. These methods could be divided into relatively simple algorithms and more sophisticated ones. Although simplier algorithms don't provide error free results, they can be used as preprocessing for more complex three-dimensional face recognition, mainly because they are not time expensive.

Locating the Face

Although the three-dimensional facial scans mostly contain foremost a face, various unusable data, such as clothes, hair and ears, are present. The face locating method that is used here was described by Segundo et al. [27]. The algorithm consists of two stages. At first, the face candidates are selected, within these candidates a face is located then.

Segundo claimed that face is the closest region to the acquisition system. First, the K-means algorithm with the initialization to divide into three clusters is applied. This will segment the scan into three regions: background, body and face. However, this step alone is not correct enough to extract the face region properly because some characteristics, like hair or neck may interfere. Simultaneously, with the K-means the sobel (edge detection) algorithm is applied to the range image. On the foremost cluster obtained by K-means and range image with detected edges is then applied logical and operator. Resulted regions are considered to be face candidates. On this regions the edge detection algorithm is applied again.

Second stage of the face localization is initiated with Hough transform, looking for an ellipse. After that, the candidate regions containing the found ellipse are selected. On these selected regions, closing process to fill holes is then applied (application of the dilation filter followed by the erosion filter). This resulting region is then used as a mask to the original range image and a face is after this manner located.

Diagram of the face locating algorithm is in Figure 2.14, pictures from [27] are used.

Locating the Nose Tip

Location of the nostip is in many three-dimensional facial recognition methods fundamental parts of preprocessing [5, 24, 28, 29]. Various techniques of localization of this point are used.

Heseltine during the face preprocessing in his recognition approach claimed that the nose is the most protruding point on the surface [5]. To handle the head rotation, the face is iteratively rotated about x and y axes. The result is that the nose tip has the smallest z coordinate on more occasions than any other vertex.

Segundo et al. [27] proposed the algorithm for the nose tip localization that consists of two stages. First, the y-coordinate is found and then appropriate x-coordinate is assigned. To find the y-coordinate, two y-projections of the face are computed – the profile curve and the median curve. The profile curve is determined by the maximum depth value, while the median curve by the median depth value of every set of points with the same y-coordinate. Another curve that represents the difference between the profile and the median curves were created. maximum of this curve along y-axis is the y-coordinate of the nose. See figure 2.15.

The x-coordinate of the nose is founded by projecting the x-projection of the curvature image (2.5.2). This is done by calculating the percentage of peak curvature points of neighbor rows centered in the nose y-coordinate of every column. The nose tip x-coordinate can be determined by looking for a peak at this projection, as can be seen in Figure 2.16.

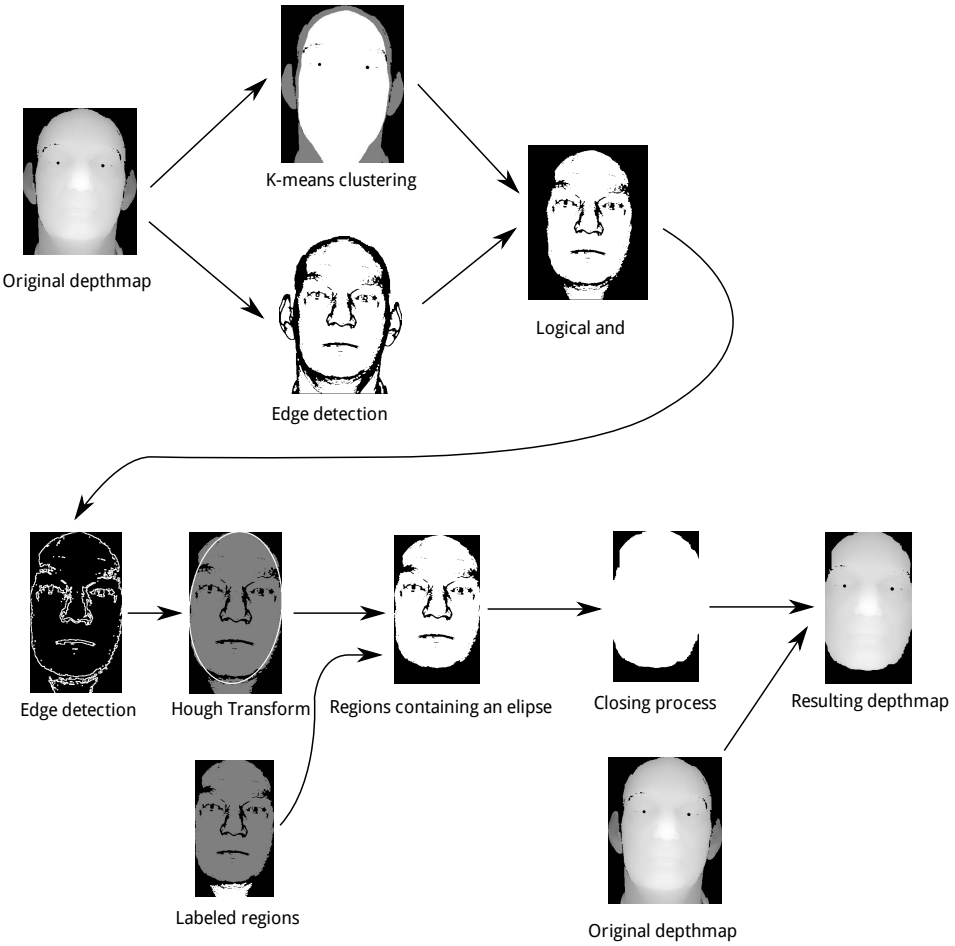


Figure 2.14: Diagram of the face locating algorithm

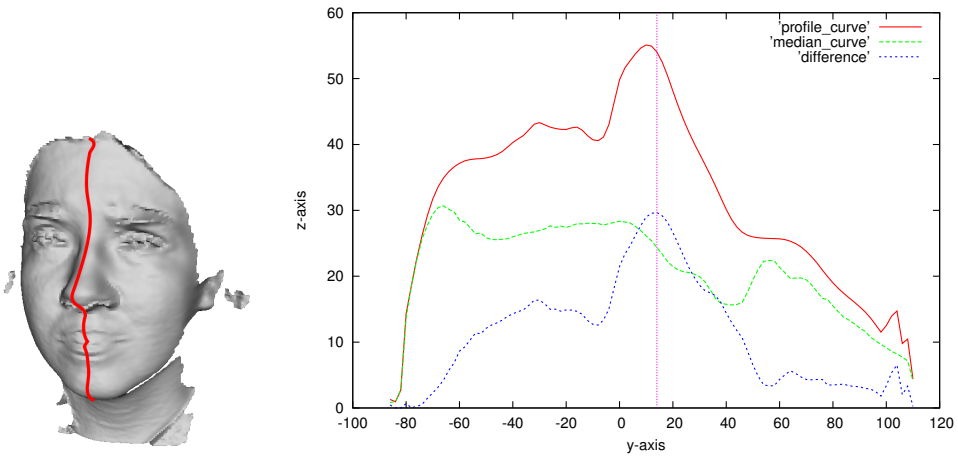


Figure 2.15: Localizing the y-coordinate of the nose

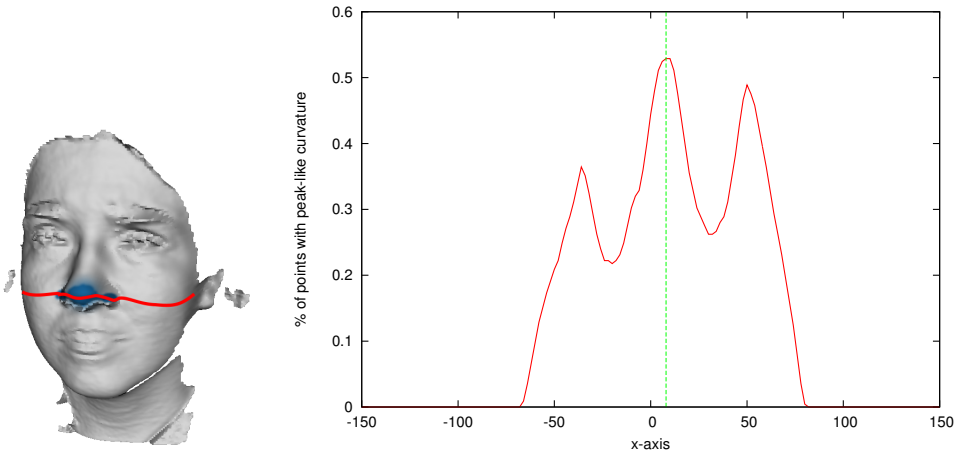


Figure 2.16: Localizing x-coordinate of the nose

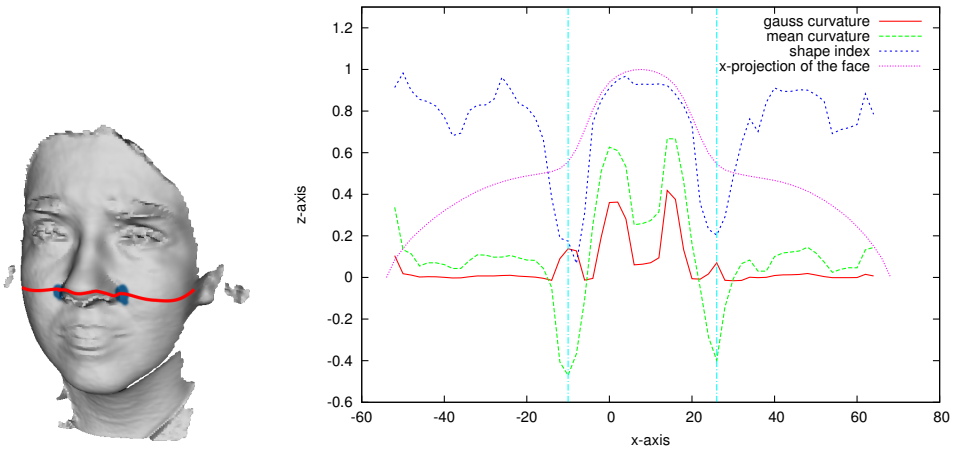


Figure 2.17: Localizing nose corners

Nose Corners

A nose corners localization method described in [27] is similar to the the nose tip in principle. Segundo et al. recommends to find the maximum variations in the horizontal profile curve. The horizontal profile curve is the x-projection that represents the set of points with the same y-coordinate value, in this case, the nose tip y-coordinate. To detect the nose corners, Segundo et al. calculate a gradinet information of this curve and look for one peak on each side of the nose tip. However, nose corners seems to be located a bit far from the peaks on the gradient curve.

A more accurate nose corner localization could be achieved by the use of x-projection data obtained from curvature analysis. Either searching for mininimum values in mean curvature projection or searching for mininimum values in shape index projection is possible. See figure 2.17.

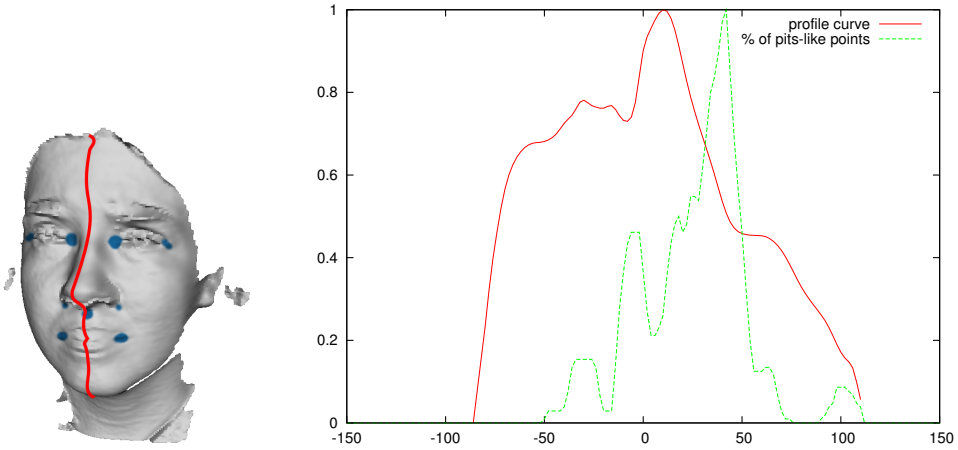


Figure 2.18: Localizing y-coordinate of the eye corners

Inner Eye Corners

Inner eye corners detection proposed by Segundo is based on the curvature analysis again. First, the percentage of pit curvature points of every set of points with the same y-coordinate is calculated. There are three peaks, representing eyes, nose base and mouth (Figure 2.18). As the nose coordinates are known, each peak can be assigned to it's respective facial feature. The x-coordinates of the eye corners are computed from another x-projection curve, where the ratio of the pit curvature points for each x-coordinate are calculated. The eyes match to the two archs on this curve.

Finding the facial landmarks is sometimes an iterative process. Detected landmarks are then used for normalization of the face orientation and then the landmark detection process is applied again.

2.5.4 Eigenface for the Three-Dimensional Face Recognition

The eigenface method applied on three-dimensional face recognition is very similar to it's two-dimensional variant described in section 2.4.2. The mean face and afterwards the calculation of the eigenvectors and eigenvalues from covariance matrix is performed on the range images. In this section several improvements of applying eigenface method on the three-dimensional data will be described.

Face Orientation Normalization

The face orientation normalization plays an important role during the recognition process, because it improves the recognition performance. The distance between the aligned face scan and the face of the same person stored in the database is much smaller than in the case of unaligned face scans.

Heseltine [5] in his recognition algorithm normalizes the face orientation during the landmark detection. First, the nose tip is detected and the face is translated such the nose tip is located at the coordinate origin. Then, the roll correction is performed by locating the nose bridge and rotating whole face about the z-axis to make nose bridge vertical in the x-y plane. After that, the forehead is located and the face is rotated about the x-axis to move the forehead directly above the nose tip. The final step of the alignment is the

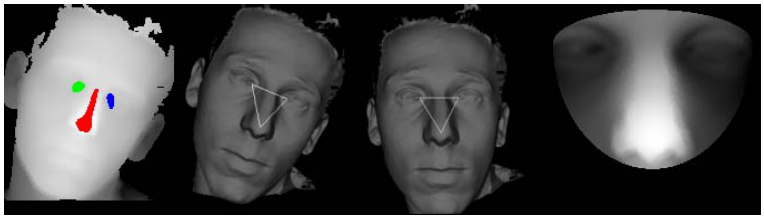


Figure 2.19: Localization of eye corners and nose, projection to generic position and conversion to range image

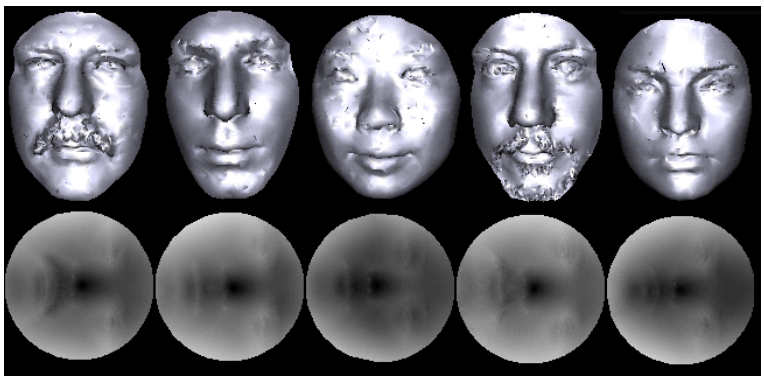


Figure 2.20: Examples of mapped range images

rotation about the y-axis to correct the head pan. During this step, the symmetry of the face is used.

Colombo et al. [30] locates both eye corners and nose tip first. Then this triplet is projected into the generic position, the whole face is converted to the range image and cropped with some mask. The result of this process is the input for eigenface-based face recognition. The illustration of this process is in Figure 2.19 (taken from [30]).

Range Image Processing

The face recognition method propsed by Pan et al. [29] maps the face surface to a planar circle. First, the nose tip is located and a region of interest (ROI) is picked. The ROI is a sphere centered at the nose tip. After that, a face surface within the ROI is selected and mapped to the planar circle. A function E that measure the distortion between original surface and plane is used. The transformation to the planar circle is performed so that E is minimal. Some examples of mapped range images from [29] are in Figure 2.20.

Heseltine [5] shows, that the application of some image processing techniques to the range image has a positive impact to the recognition. Mainly the application of the sobel filter increases the recognition performance. In Figure 2.21 the original range image and some applied filters are shown.

2.5.5 Model Based 3D Face Recognition

So far, only the two-dimensional methods or their adaptation to the three dimensions were described.

Lu et. al [24] proposed a method, that compares a face scan to a 3D model stored in a

Figure 2.21: Original depth map, sobel x, sobel y and sobel magnitude

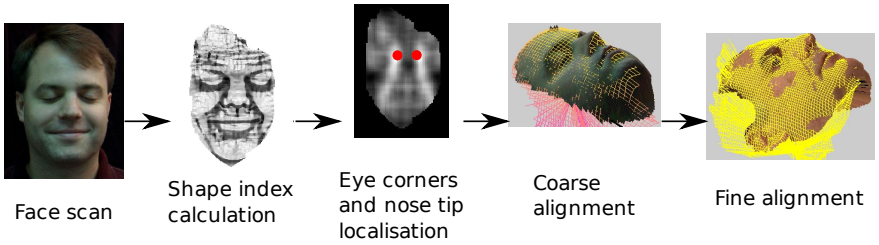


Figure 2.22: Process of the model based recognition

database. The method consists of three stages. First the landmarks are located. Lu uses the nose tip, the inside of one eye and the outside of the same eye. Localization is based on the curvature analysis of the scanned face.

These three points obtained in the previous step are used for the coarse alignment to the 3D model stored in database. A rigid transform of three pairs of corresponding points [31] is performed in the second step.

Fine registration process, the final step, uses the Iterative Closest Point algorithm [32]. The root mean square distance minimized by the ICP algorithm is used as the primary comparison score of the face scans. Additionally, a cross-correlation between the shape index maps are calculated and then used as the second comparison score. The whole process of model based recognition is in Figure 2.22.

2.5.6 Histogram based face recognition

The face recognition algorithm introduced by Zhou et al. [33] is able to deal with small variations caused by facial expressions, noisy data and spikes on the three-dimensional scans. After localization of the nose, the face is aligned, so the nose tip is situated in the origin of coordinates and the surface is converted to the range image. After that, a rectangle area around the nose is selected. This rectangle is divided into N equal stripes. Each stripe n contains S_n points. Maximal $Z_{n,max}$ and minimal $Z_{n,min}$ z-coordinates within each stripe are calculated and the z-coordinate space is divided into K equal width bins. Each $bin_{n,i}$ is defined by its z-coordinate boundaries:

$$bin_{n,i} = [Z_{n,k-1}, Z_{n,k}] \tag{2.23}$$

$$Z_{n,0} = Z_{n,min}, Z_{n,1}, \ldots, Z_{n,K} = Z_{n,max} \tag{2.24}$$

The feature vector v containing $N \cdot K$ components is calculated:

$$v_{k,n} = \frac{|\{p_i(x_i, y_i, z_i)|\ p_i \in S_n, Z_{k-1} < z_i < Z_k\}|}{|S_n|} \tag{2.25}$$

where $k \in [1, \ldots, K]$ and $n \in [1, \ldots, N]$.

The input range image and the corresponding feature vector is in Figure 2.23. The comparison between two faces is performed by a distance calculation between the two corresponding feature vectors.

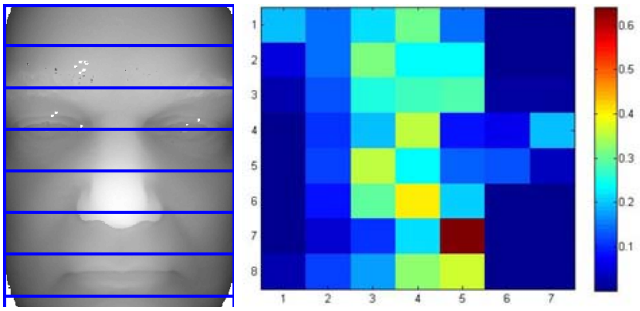


Figure 2.23: Range image divided into stripes and corresponding feature vector

Chapter 3

Obtaining 3D Face Data

To evaluate face recognition performance, testing data are needed. In this section the A4 Vision Enrollment Station and two three-dimensional face databases will be described.

3.1 A4 Vision Enrollment Station

The A4 Vision Enrollment Station (Figure 3.1) is a camera for enrolling users to the biometric access control systems. It is a part of solution, provided originally by A4 Vision, that consists of the enrollment station, access control face reader and the software development kit.

The enrollment station is connected to the PC or laptop and the enrollment application allows the administrator to enroll data subjects to the database. The SDK shipped with the camera enables the addition of facial biometrics and secure connection into back-end applications, deployment of an overall system and integration into existing third-party applications. The Vision Access SDK manages face readers, performs user enrollment, auditing, system analysis and monitoring of stand-alone solutions locally and centrally of network-configured physical access solutions.

A structured infrared light is used in the enrollment station for obtaining the facial scans. Two infrared sources project the striped pattern on the scanned surface. Due to the surface shape of the object, the pattern appears geometrically distorted from different angles and from this distortion, the three-dimensional shape is calculated. The usage of

Figure 3.1: A4 Vision Enrollment Station

Figure 3.2: Example of face scan captured by the A4 Vision Enrollment Station

the infrared light in the camera allows to operate in poor lighting conditions. An example of face scan obtained by the enrollment station is in Figure 3.2.

Unfortunately, the API provided by the SDK is very high-level and doesn't provide direct access to the proprietary stored facial scans. It has emerged, that the usage of the A4 Vision Enrollment Station for face scans acquisition that will be further used in implementation and testing of face recognition algorithm, described in this thesis, is unsuitable.

3.2 Three-dimensional Face Database

The usage of the A4 Vision Enrollment Station for purposes of this thesis has been shown as problematic. The impossibility of direct access with A4 Vision software development kit to the three-dimensional data in stored facial scans has led to search for alternative source of data.

3.2.1 GavabDB

GavabDB [25] is relatively small, freely available, three dimensional face database that consists of 549 scans of 61 individuals. Each person in the database has been scanned with the various facial expressions and head orientation. The examples are in Figure 3.3.

The data in the database contain, except the facial scans, also noise, spikes and the data that are not part of a face, like clothes and hair. Although it is not desired for recognition, it can prove the robustness of the recognition algorithm. Preprocessing techniques should be applied on the data to extract the face and eliminate the impact of the noise on the recognition performance.

Each person in the database is represented with 9 scans stored in VRML format. Two frontal scans with neutral face expression, four scans with head rotation (up, down, left and right) and three scans with facial expressions are present.

3.2.2 FRGC Dataset

Face Recognition Grand Challenge database [23] is a large dataset of three-dimensional face scans as well as high and low resolution photographs captured in controlled and uncontrolled lighting conditions. It is not freely available, but can be obtained for research purposes.

Figure 3.3: GavabDB facial scans examples

The 3D images were taken under controlled illumination conditions appropriate for the Minolta Vivid 900/910 sensor during spring 2003, fall 2003 and spring 2004. The dataset contains 466 subjects and almost 5000 individual scans. Neutral face expressions as well as smile and other facial grimaces are present. Contrary to the A4 Vision Enrollment Station, the Minolta Vivid scanner use a laser beam to obtain a three-dimensional model of scanned surface. The Minolta Vivid 910 is depicted in Figure 3.4. An example of some scans from the FRGC dataset is in Figure 3.5.

Figure 3.4: Minolta Vivid 910

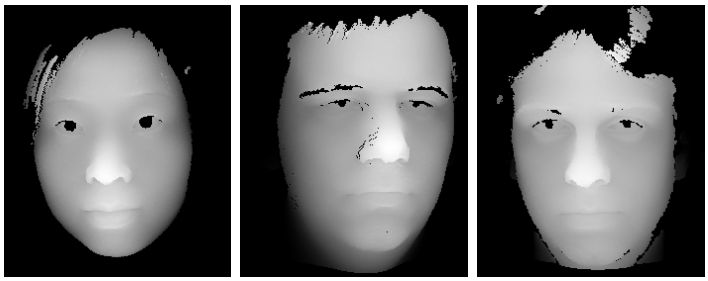


Figure 3.5: FRGC dataset examples

Chapter 4

Proposed 3D Face Recognition method

A three-dimensional face recognition method is proposed in this chapter. The method covers all parts of recognition, including training, preprocessing, calculating the comparison score and the final decision. It is based on the previous works described in the chapter 2. It is modular, so each module deals with the specific problem and can be replaced with another which does the same task.

Mostly different recognition techniques were developed and tested on different databases, therefore it is hard to say which algorithm is better. Some databases contain noise in the data, like spikes and holes in scans, another ones are focused on various facial expressions. As a result, the face recognition method which has a greater EER value could be better then another method with lower EER value. The same situation is in the facial landmarks finding. Success rate depends both on the quality of the algorithm and the input data.

The overview of the proposed method is in Figure 4.1. The landmark detection module is responsible for locating the face features. The input of this module is the pointcloud representation of the face surface and the output is located landmarks, e.g. the nose tip, nasal bridge and inner eye corners. A method described in [27] and in section 2.5.3 is used.

The control is passed to the normalization module then. This module aligns the face to the predefined position within the space using the algorithm described in [31]. Three pairs of corresponding points are used to transform the whole face scan to be aligned to a mean face, which is calculated from the training set.

The next step in the recognition process is the comparison. In the case of verification,

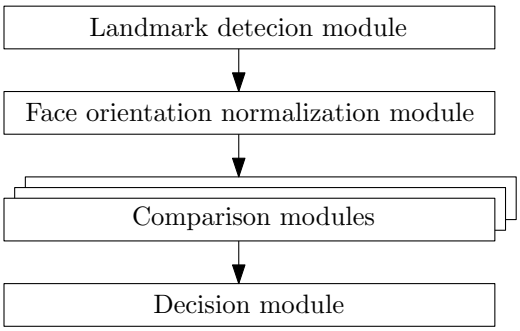


Figure 4.1: Proposed face recognition overview

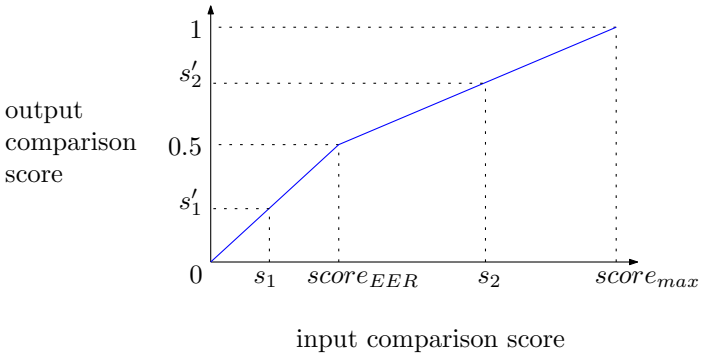


Figure 4.2: Calculation of the final score

the comparison score will be calculated from the probe scan and selected reference template in the gallery database. The selection of the reference template is based on the claimed identity provided by the data subject. In the case of identification, the comparison score is calculated between the face scan and all templates in the database. More than one comparison module can be used simultaneously. The input of a comparison module is the normalized face scan and the result is the comparison score.

The eigenface method ([12, 5]) and the recognition using histogram-based features ([33]), both described in the section 2.5, are used. Each method provides different comparison score to the decision module, where the final conclusion is made.

The calculation of the final score, thus the decision if the data subject is authenticated or rejected, can be performed in a number of ways. To increase the security of the recognition, the lowest score is chosen as the final score. This will lead to lower FAR values but also greater FRR values. On the other hand, choosing the greatest score from the comparison modules will cause lower FAR values and thus the users' satisfaction.

Various recognition techniques usually use different threshold values for their decision. The final score calculation method that takes into account different thresholds is as follows. In this method, not only the fact if the threshold was exceeded or not, but the score distance from the threshold is relevant. Assume the threshold s_{EER}, for which the EER is reached, and the comparison score s provided by one of the comparison modules. If s is lower or equal than s_{EER} then the normalized score s' is

$$s' = \frac{s}{2 \cdot score_{EER}} \tag{4.1}$$

and when s is greater than s_{EER}, the normalized score s' is

$$s' = \frac{1}{2} + \frac{s - score_{EER}}{2 \cdot (score_{max} - score_{EER})} \tag{4.2}$$

The input scores are transformed into the normalized values (see Figure 4.2) and then they can be compared. The selection can be performed by choosing the greatest value, the lowest value, average or weighted average. Some other possibilities of score normalization and combination are described in ISO standard Multi-Modal and Other Multi-Biometric Fusion [34]. One of the goals of this thesis is to find the optimal decision technique which minimalizes the EER. This will be achieved by batch testing of the recognition algorithm and the decision module with various parameters.

4.1 Database binning

Once the face is properly oriented and all the landmarks are detected, the first stage of the biometric recognition begins. Landmarks, their mutual positions and curves on the face surface that connects the landmarks provide several measurable characteristics that can be used for classification of the face to the specific bin. Subsequent recognition is then performed only within the specific bin. This process is similar to the two-level fingerprint recognition, where the classification into arch, tended arch, loop, whorl or twin loop of the fingerprint pattern is performed first and then the recognition continues within the specified pattern type only.

The goal of this part is to find out which facial characteristics provide good binning results. There are two main binning performance criteria – the penetration rate and the binning error rate [35]. The penetration rate is the average ratio between the searched data and total number of templates. The binning error is the ratio between the number of matching template-sample pairs that were placed in different bins and the number of pairs assessed.

4.2 Eigenface

The Eigenface method uses principal component analysis for reduction of the image space, where each face image is represented as a vector. Individual components of the vector correspond to the pixels of the input images. The Eigenface recognition could be performed on the same images on which the dimension reduction was made, but this approach leads to the need of recomputation principal components every time the new face scan is added to the database.

The second option is to compute the principal components on a separated training set. In this case, the sufficiently large training set, which covers a diversity across the face space, is needed. The optimal size of the training set, which provides the best results, will be found.

4.3 Histogram Based Recognition

The recognition method that uses histogram based features has two important parameters – the number of stripes that split the face and the number of bins in each stripe to which the z-coordinates are divided. The optimal number of stripes and bins was published in [33].

It may be obvious that some components of the calculated feature vector have more discriminating potential, while other components may be strongly affected by facial expressions and it is desired not to rely on them too much. Therefore, a weighted distance between the feature vectors will be used. The weight coefficients will be discovered using a genetic algorithm. Initial population will consist of random vectors that will represent the weight coefficients. In each step of the genetic algorithm the weights will be randomly mutated and crossed. The evaluation of the candidate solutions is based on the equal error rate that the recognition procedure provides with the current candidate.

Chapter 5

Implementation

In this chapter, specific algorithms that are used during the whole recognition pipeline process are described. This chapter is concentrated on the recognition pipeline in general. No programming language specific matters will be taken into account.

The face recognition pipeline is divided into several modules as was described in Chapter 4. Each module deals with specific task and the output of one module is taken as the input to the subsequent module. The result of the preprocessing part is properly oriented face with indicated landmarks and facial curves.

From the properly oriented face scan, the feature extraction module extracts desired features and stores them in the template that can be used to direct comparison between two face scans. Utilized feature extraction algorithm strongly depends on the selected recognition method.

5.1 3D face data representation

The FRGC dataset [23] uses the pointcloud representation of the data. Each file contains the x, y, z coordinates of each point that forms the face. For the purposes of the preprocessing steps, the pointcloud representation should be converted into the range image.

A parallel projection is used during the conversion to the range image. Each point in the three-dimensional space is lineary projected to the xy-plane that is divided into regular grid. One cell in the grid corresponds to one pixel in resulting range image. If more points are projected into the same resulting pixel, then the point with the maximal z-coordinate is used as the pixel value. Although the average or the median value of points that were projected into the same pixel can be used, the selection of the maximum solves the problem of overlapping two surfaces (see Figure 5.1).

After the range image is successfully computed, holes that may occur due to low density of points in the three-dimensional space or due to completly missing points have to be filled. This is achieved in two steps. First, the missing coordinates of pixels are located through the use of the morphological closing process and then the values in new pixels are interpolated according to their neighbor values.

In the binary morphological operations [36], an image is viewed as a subset of the two-dimensional integer grid $\mathbb{Z}^2$. The predefined floating shape (called the structuring element) is then probed with the original image and basic pixel-to-pixel binary operations are performed.

The dilation of the image A with the structuring element B is defined as:

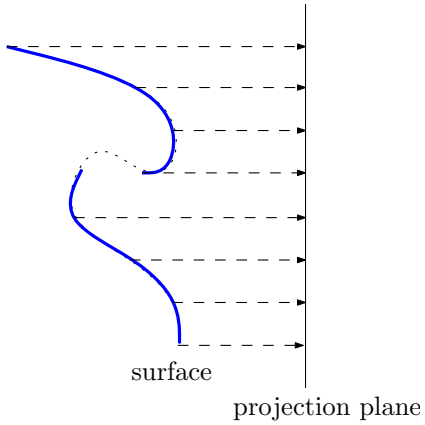


Figure 5.1: Range image projection

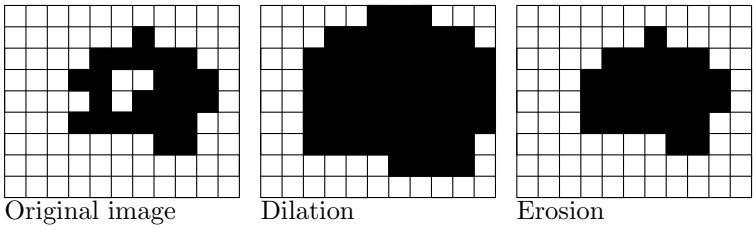


Figure 5.2: Morphological closing – dillation followed by erosion

$$A \oplus B = \{z \in \mathbb{Z}^2 | (\hat{B})_z \cap A \neq \emptyset\} \tag{5.1}$$

In each point of the original image A, the structuring element B is compared with the original value. If the intersection of the structuring element with the original value is not an empty set, then point is added to the resulting image.

The erosion is defined in the similar way:

$$A \ominus B = \{z \in \mathbb{Z}^2 | (B)_z \subseteq A\} \tag{5.2}$$

The morphological closing is the dillation followed by the erosion. An example of this process is in the Figure 5.2. The structuring element 3×3 pixels was used.

After the closing process on the range image finishes, the original image O is then substracted from the result of the closing process C using the difference operation. This will provide a list L of pixels whose values have to be interpolated:

$$L = C - O = \{z \in \mathbb{Z}^2 | z \in C \wedge z \notin O\} \tag{5.3}$$

The interpolation of the missing values is an iterative process. In each iteration, one pixel from L is selected and if it has neighbours with non-zero values, then the new value of the selected pixel is equal to the average of its neighbours (see Algorithm 5.1).

The usage of maximal z-coordinate of the point during the calculation of the range image as well as some noise in the original pointcloud data may result presence of spikes that have to be removed from the range image. A simple method, when value of each pixel is compared with the average value of surrounding pixels, is used. If the difference between

Algorithm 5.1 Interpolation of the missing values.
Input: L – list of points whose values have to be interpolated

```
while L ≠ ∅ do
  select p; p ∈ L
  if has_neighbours(p) then
    newvalue := average_of_neigbours(p)
    L := L − p
  end if
end while
```

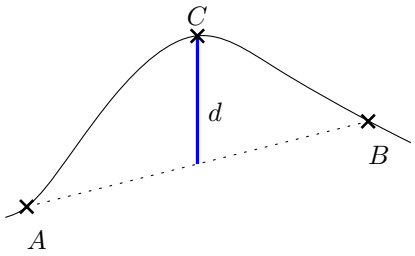


Figure 5.3: Estimation of the curvature at the given point C

the pixel and its surroundings exceeds specific limit, then the pixel is considered as a spike and its value is made the same as the average of its neighbours.

5.2 Face surface processing

The Landmark detection module depends on the face surface processing, especially on the curvature analysis. In this process the surface is divided into areas that are considered as peaks, pits, saddles, valleys and ridges. The curvature analysis is performed on the smoothed range image to avoid the influence of insignificant local deformations on the surface. The iterative Laplacian smoothing algorithm is used. Value of every pixel x_i is modified in every iteration according to its 8 neighbours [37]:

$$x_i = x_i + \alpha \frac{\sum_{x_j \in neighbours(x_i)} x_j}{N} \tag{5.4}$$

where $\alpha \in (0, 1]$ is smoothing coefficient and N is number of neighnours of point x_i.

The idea of function that estimates the curvature at given point is situated in Figure 5.3. The curvature at the point C is estimated from the difference d between the z-coordinate of the point C and the connecting line between points A and B. This function provides greater results for the points located on a sharp part of surface and smaller results located on a straighter part of the surface. The curvature c at the given point C is then estimated as $c(C) \approx d$.

The curvature of every point is calculated in four directions (horizontal, vertical and two diagonal). As the principal curvatures is selected the pair of curvatures which is more different, e.g. if the difference between the horizontal and the vertical curvatures is greater than between the diagonal pair, then the principal curvatures k_1 and k_2 are selected as the horizontal and the vertical curvature.

Finally the mean curvature (Equ. 2.21), Gaussian curvature (Equ. 2.20) and the shape index (Equ. 2.22) is calculated.

The results of the curvature analysis are shown in Figure 5.4. Brighter pixels stands for higher values and darker pixels for lower values.

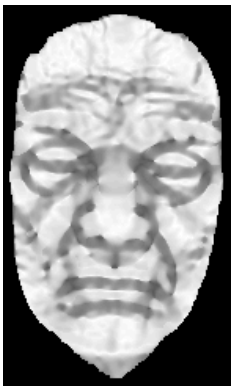
a) Original range image

b) Principal curvature k_1

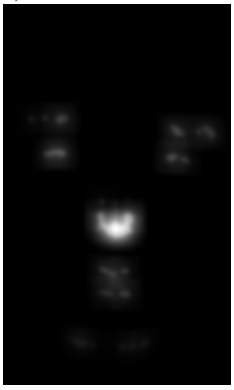
c) Principal curvature k_2

d) Mean curvature

e) Gaussian curvature



f) Shape index



g) Peak density

Figure 5.4: Results of the curvature analysis

5.3 Landmark detection

Proper landmark detection on the face surface is key part during the recognition pipeline. The nasal tip, nasal bridge, inner corners of the eyes, the lower corner of the nose and the

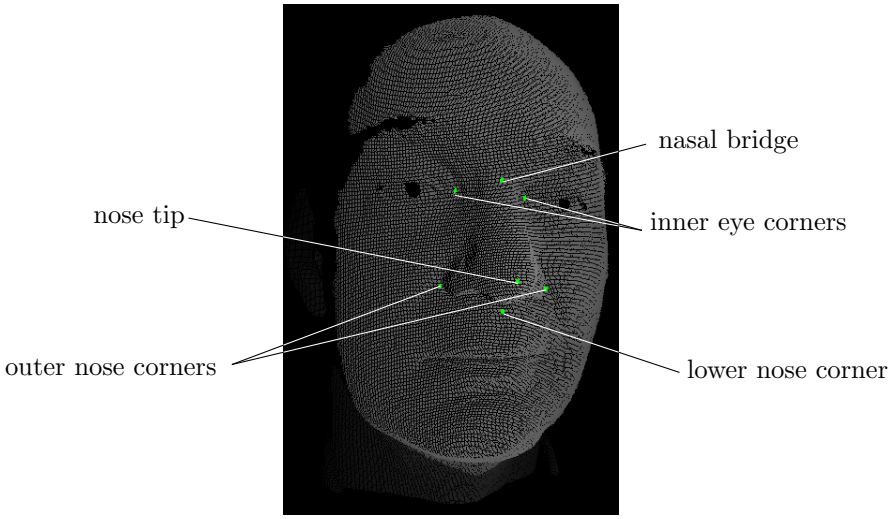


Figure 5.5: Located landmark on the face – tip of the nose, nasal bridge, innercorner of the eyes, outer nose corners and lower corner of the nose

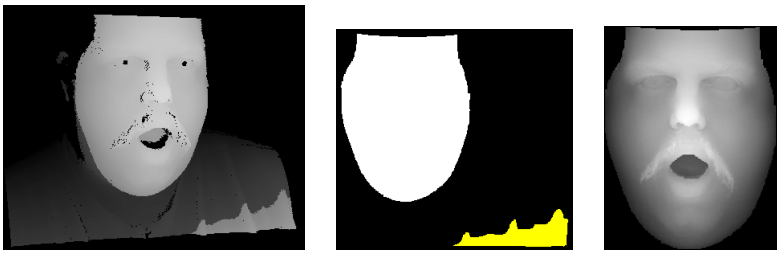


Figure 5.6: K-means clustering and face region selection

outer nose corners are detected in this process. An example of located landmarks is in the Figure 5.5.

Before the landmark detection process can start, area of the face has to be separated from the other data, such are clothes, neck and hair. This is achieved in two steps. First, the K-means clustering separates the data into two clusters. This will isolate parts like ears, shoulders and other parts that are not protruding and therefore they are not considered as a surface which may contain searched landmarks. However, some protruding parts, like shirt collar or haircut, may be still present after the K-means clustering. Areas of these regions are compared and the greatest one is finally considered as a region that contains searched landmarks (See Figure 5.6).

After the face region is selected, the nose tip is located as a point with the highest peak density inside the region (Figure 5.4 g). The eye corners are located analogically as points with the highest pit density. An additional heuristics is used during the eye corners detection. Since the location of the nose tip is known and with the assumption that the face is not rotated upside down, a highest pit density point can be searched only within the reduced area above the nose tip.

As the nasal bridge is marked a point on the connecting line between the eye corners with the highest curvature. However, the localization of the nasal bridge and nose tip is not completely accurate. Therefore, an additional processing has to be done. The nasal

bridge is usually situated a bit higher on the face and the nose tip location may be shifted due to nose asymmetry or false peak detection. The area between temporary located nasal bridge and nose tip is divided into horizontal curves. In each curve, that consists of pixels from one range image row, is located a point with the highest horizontal curvature and added to a set P. A line that fits all points in set P in least square mean is calculated after that. This line is then projected to the face surface and the profile curve is founded by this manner.

The nose tip as well as the nasal bridge and the lower nose corner is located on the profile curve using the second derivation of the profile curve.

5.4 Face orientation normalization

For the purpose of the orientation normalization, the nose tip, nasal bridge and lower nose corner are used. The whole model is moved first, such the nose tip is situated in the origin of coordinates ($[0, 0, 0]$). After that, the rotation around all three main axes is performed. Two orientation normalization algorithms were implemented and compared.

5.4.1 Non-iterative orientation normalization

The rotation around the y axis is performed first. A properly oriented face has the nasal bridge and lower nose corners x-coordinates equal to zero. If the current face does not meet these criteriae, then the face has to be rotated. The principle of the y axis rotation is illustrated in Figure 5.7. An intersection point I between the nose tip xy-plane and the line between nasal bridge and lower nose corner is calculated. After that, the rotation angle is computed as an angle between the vector $(0, 0, 1)$ and $I - [0, 0, 0]$.

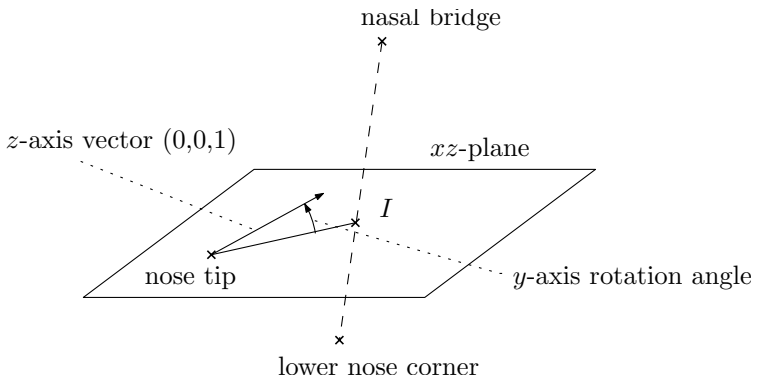


Figure 5.7: Principle of y axis rotation

The head lean compensation is achieved through the rotation along z axis. The angle of rotation is equal to the angle between the vector $(0, 1, 0)$ and connecting line between the nasal bridge and the lower nose corner (see Figure 5.8).

Finally, the rotation around x-axis compensate head forward and backward bending. The head is rotated so that the angle between vector $(0, 0, 1)$ and connecting line between nasal bridge and nose tip is equal to 1.1 radians. This angle was established by the manual inspection of the face scans. Situation is illustrated in Figure 5.9.

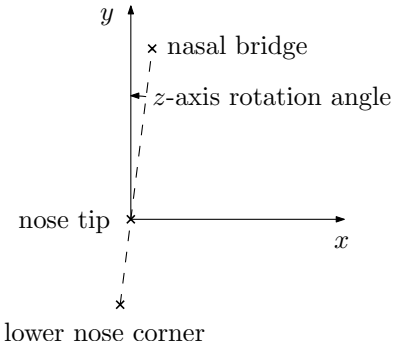


Figure 5.8: Principle of z axis rotation

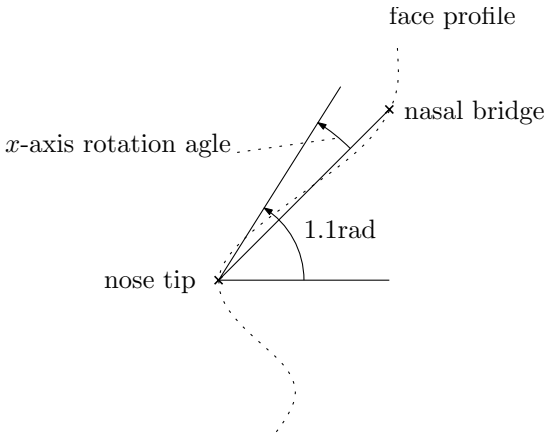


Figure 5.9: Principle of x axis rotation

5.4.2 Iterative orientation normalization

Contrary to the non-iterative variation, the iterative orientation normalization algorithm uses only two reference points (nose tip and nasal bridge). Additional information is gained from the assumption that the face is symmetrical with respect to the xy-plane.

In every iteration, the rotation angles along all three axes are estimated and the face is rotated. This process is repeated until the rotation angles are sufficiently small or the maximum number of iterations were reached.

The estimation of the rotation angles along the x and z axes is similar to the rotation in non-iterative normalization variant. The vector between the nose tip and the nasal bridge is rotated in such manner that x-coordinates of its ending points are equal to zero and the angle between the xz plane and the nasal bridge vector is 1.1 radians. Situation is illustrated in Figure 5.10.

The rotation along the y axis is estimated with the aid of planar symmetry of the face. From both left and right part of the face is calculated the centroid point, which represets the cumulative z coordinates of the points on left or right side. Correctly oriented face has both centroid values equal. The rotation angle is calculated from the angle between the centroid connecting line and x axis.

It has been observed that 5 iterations is usually enough for correct orientation. Rotation angles in further iterations are very small and do not have big influence on orientation correctness.

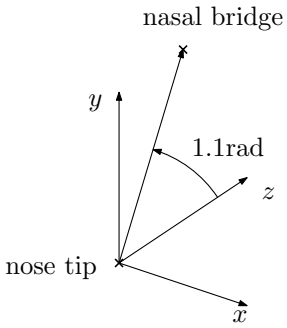


Figure 5.10: Nasal vector normalization

Table 5.1: Anatomical-Bertillon face features

	Feature name	description
1	Nose width	distance between outer nose corners
2	Nose depth	distance between line connecting the outer nose corners and the nose tip
3	Nose height	distance between the lower nose corner and the nasal bridge
4	Nose sharpness	Angle between the nasal bridge, nose tip and the lower nose corner
5	$\frac{\text{Nosedepth}}{\text{Nosewidth}}$	Ratio between the nose depth and the nose width
6	Nose volume	Count of voxels that are located under nose area
7	Bridge of the nose depth	z-axis difference between the eye corners and the nasal bridge

5.5 Feature extraction

The feature extraction module requests a properly oriented face as an input. Further manipulation strongly depends on the used recognition method. The output of the feature extraction module is a biometric template which can be used in subsequent comparison module.

5.5.1 Anatomical-Bertillon face features

Anatomical-Bertillon face features can be directly extracted from the properly oriented face with located landmarks. The list of these features together with their description is provided in Table 5.1.

Along with this basic features, another 48 features connected with facial curves on the face were extracted. Four main facial curves are shown in Figure 5.11. Similarity between the selected curve and the corresponding curve extracted from the mean face is used as a feature. There are utilized four different similarity coefficients:

Pearson product between set X and Y is defined as:

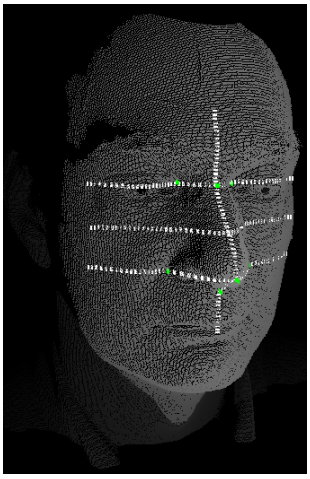


Figure 5.11: Extracted facial curves

$$r = \frac{\sum_{x \in X}(x - \mu_X) \sum_{y \in Y}(y - \mu_Y)}{N \sigma_X \sigma_Y} \tag{5.5}$$

where μ_X is the arithmetic mean of the set X and σ_X is its the standard deviation.

Cross correlation between set X and Y is defined as:

$$r = \frac{\langle X, Y \rangle}{\|X\| \cdot \|Y\|} \tag{5.6}$$

where $\langle X, Y \rangle$ is dot product and $\|X\|$ is norm of X.

Hausdorff distance d_H is defined as:

$$d_H(X, Y) = \max\{\sup_{x \in X} \inf_{y \in Y} d(x, y), \sup_{y \in Y} \inf_{x \in X} d(x, y)\} \tag{5.7}$$

where $d(x, y)$ can be arbitrary metric function in the given metric space. Euclidian distance was used.

Sum of differences is simply defined as:

$$d = \sum_{i=1}^{N} X_i - Y_i \tag{5.8}$$

where N is size of sets X and Y.

Each curve was compared with the corresponding curve extracted from the mean face using all similarity coefficients mentioned above. Moreover, the similarity of the first and the second derivation of all facial curves to their corresponding mean counterparts were calculated.

Another 6 metrices were extracted from the curvature at the nose tip, nasal bridge and the point halfway between the nosetip and the nasal bridge. Both vertical and horizontal curvatures were measured in all three points.

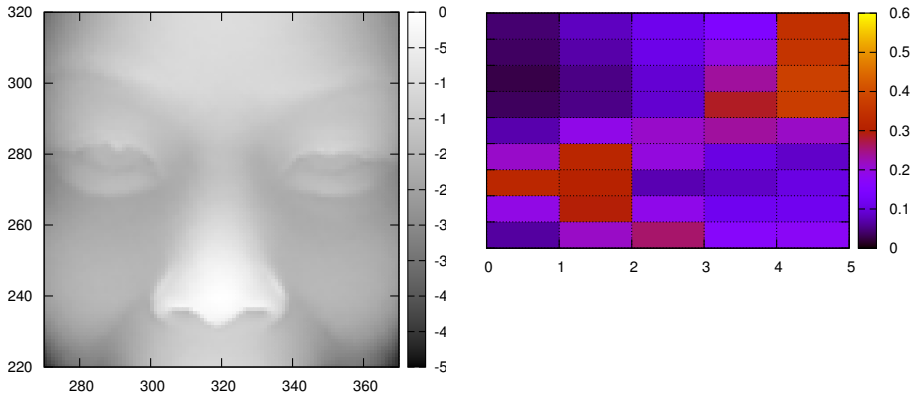


Figure 5.12: Area of the face selected for the extraction of histogram-based features and encoding of the corresponding template

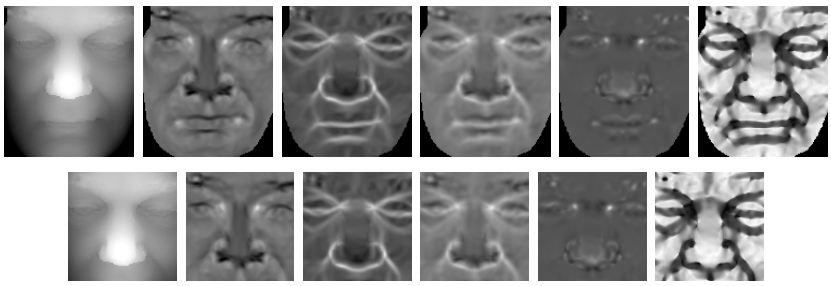


Figure 5.13: Various inputs of PCA

5.5.2 Histogram-based features

The process of extracting histogram-based features was described in section 2.5.6 in theoretical part of this thesis. Histogram of z-coordinates in face surface divided into stripes was calculated directly from the pointcloud as well as from the range image. Although the range image does not contain so many points as the pointcloud representation of the face, additional processing such as smoothing may be easily applied to reduce the noise influence on the recognition performance.

To avoid the impact of face parts that suffer from high intra-class variability, such is the mouth region, only a selected area of interest was used to calculate the histogram-based feature vector. Both eyes, nose, upper part of cheeks and part of a forehead are included in that region of interest (ROI) (see Figure 5.12). The exact coordinates of the area was selected by manual inspection of the training set.

5.5.3 PCA features

The PCA is a holistic face recognition method that seeks for the directions in which data varies most. Although it was originally intended for the two-dimensional face recognition, it can be used on the three-dimensional data as well. Furthermore, not only range image, but also other images that describe the face surface may be used as an input. All the resulting curvature maps from the preprocessing module were used (see Figure 5.13). Furthermore, two different areas were selected – first contains entire face and the second one only region around nose, both eyes and upper parts of cheeks.

Chapter 6

Performance evaluation

This chapter reports an overview of achieved results. Success rate of the landmark detection module, binning error and penetration rate as well as the equal error rates of utilized recognition methods is described here. All evaluations were performed on the FRGC dataset [23].

6.1 Landmark detection

As the training set for the landmark detection module were selected 100 scans from the "spring2004" part of the FRGC dataset. Some of the training set images were choosen randomly from the entire dataset, but some potentially problematic scans were added. Since the landmark detection module operates with the surface curvature, persons with indistinct facial features, such is small nose or relatively flat eye corners, were selected. For the examination of the face orientation normalization were added persons whose scans were rotated more than it is usual in the rest of the dataset. Preprocessing parametres were selected in order to achieve as high success rate as possible.

All 1837 face scans from the "spring2004" part of the dataset were probed and both iterative and non-iterative orientation normalization algorithms were tested. In both cases, the landmark detection module failed in only 4 cases. Although it is only 0.2%, this number is not accurate. Number of failures were computed automatically during the batch landmark detection process. Some cases when the landmark is situated in a place that would be classified as a fail by the human are not counted, so the fail rate is likely to be higher. However, the precision of the landmark detection and orientation normalization directly correlates to the recognition performance. Therefore it can be derived from the achieved equal error rate.

6.2 Database binning

The database subdivision to the smaller parts could rapidly improve the recognition performance and reduce time needed for identification. The main parameter of database binning is the size of one bin. Consider some feature with normalized values ranging from 0 to 1. If the size of a bin is 0.1 then the extracted features from templates in the database can be assigned to one of 10 bins. Subsequently, when the probe scan is compared to other templates in the database, its bins are calculated first and only a part of the data that are classified in the same bins is searched.

The main criteria of the binning performance are the binning error rate and the penetration rate [35]. The penetration rate is the average ratio between the searched data and total number of templates. Binning errors is the ratio between the number of matching template-sample pairs that were placed in different bins and the number of pairs assessed.

As the binning criteria were selected 61 anatomical features extracted after landmark detection and orientation normalization. This was described in the section 5.5.1. There is a $2^{61} - 1$ possibilities how to combine extracted features as the binning criteria. Statistical inspection of measured features was performed in order to decide which of them has good binning suitability in reasonable time.

Good binning feature has low intra-class varinace and high inter-class variance. This means that values extracted from the same person are similar even if some facial mimic is present while the inter-class variance refers to the uniform distibution of values in face space.

In order to compare two different binning candidates, normalization of values to the range between 0 and 1 has been done. The evaluation of intra-class variance of a feature is expressed with three values:

- mean deviation μ – mean value range for each person
- standard deviation s – standard deviation of value ranges for each person
- maximum deviation m – maximal value range for each person

The evaluation of the inter-class variance was performed with the following assumption. In order to achieve good penetration rate, values have to be uniformly distributed among the space. The suitability of binning candidate in the mean of inter-class variance is expressed as the correlation c of measured values to the uniform probability density function.

The final decision, if the binning candidate is good or not is based on the discrimative potential score:

$$\text{dp\_score} = (1 - (\mu + s + m)) + c \tag{6.1}$$

This function provides greater values for binning candidates that fits the criteria mentioned above and lower values for the unsuitable binning candidates. Evaluation of features with discriminative potential score greater than 0.5 extracted after non-iterative and iterative orientation normalization algorithms are in Tables 6.1 and 6.2. It is obvious, that discriminative potential scores from the non-iterative variant are not so high as the scores from the iterative variant. Manual inspection of the normalized database also confirmed that faces aligned with the iterative variant are oriented more properly. Therefore in further tests, only the iterative variation will be evaluated as it can be expected that it will provide better results.

The inspection of feature scatter plots, where on the horizontal axis are individual persons and on the vertical axis are plotted all values of selected feature as one line, revealed additional failures in the orientation normalization procedure. As the length of the line indicates variation of measured values, failures can be spotted as unusualy length line. For example, the scatter plot of measured nose depths is in the Figure 6.1.

It is desired that there is no direct connection between selected binning features. In order to achieve good penetration rate when more that one binning criteria is used, mutual correlation of binning candidates with score greater than 0.5 was calculated (see Table 6.3).

Table 6.1: Evaluation of selected anatomical features extracted after iterative orientation normalization

Feature		Value range		Intraclass var.			Interclass var.	Score
		min.	max.	μ	s	m	c	
1	Nose width	30.0000	54.0000	0.1381	0.0852	0.5000	0.3096	0.5863
2	Nose depth	17.2859	37.4124	0.0954	0.0690	0.3967	0.4179	0.8569
3	Nose height	32.0663	64.2384	0.1263	0.0730	0.5254	0.3023	0.5776
4	Nose sharpness	1.4580	2.3431	0.1601	0.0768	0.4958	0.2457	0.5131
5	Nose ratio	0.3850	0.9588	0.1184	0.0647	0.3874	0.4095	0.8390
6	Nose volume	9210	46119	0.0909	0.0695	0.4495	0.3253	0.7155
7	Nasal bridge depth	2.8760	24.9746	0.0992	0.0524	0.2801	0.3771	0.9453
11	Profile (sum of distances)	-471.7160	585.6140	0.1425	0.0727	0.4453	0.3017	0.6413
15	Nose curve (sum of distances)	-909.2180	946.7610	0.1322	0.0651	0.3414	0.4489	0.9101
19	Middle curve (sum of distances)	-1002.2800	1158.6700	0.0862	0.0438	0.2327	0.4073	1.0446
27	Profile (1^{st} der,. sum of distances)	-41.4310	21.7782	0.1294	0.0685	0.4052	0.2288	0.6257
57	Tip of the nose vertical curv.	-0.2196	-0.0738	0.1705	0.0895	0.5112	0.2904	0.5192
60	Nasal bridge horiz. curv.	-0.2399	-0.0203	0.1441	0.1038	0.6084	0.3843	0.5279

Table 6.2: Evaluation of selected anatomical features extracted after non-iterative orientation normalization

Feature		Value range		Intraclass var.			Interclass var.	Score
		min.	max.	μ	s	m	c	
1	Nose width	19.0000	53.0000	0.0966	0.0667	0.6765	0.2334	0.3937
2	Nose depth	5.7592	36.9078	0.0611	0.0771	0.8848	0.2690	0.2460
3	Nose height	32.6760	75.1553	0.1020	0.0730	0.7231	0.2416	0.3434
4	Nose sharpness	1.4788	2.3166	0.1696	0.0886	0.5915	0.2611	0.4114
5	Nose ratio	0.1516	0.9482	0.0876	0.0685	0.8554	0.2892	0.2777
6	Nose volume	1843.8200	42721.0000	0.0687	0.0668	0.6400	0.2913	0.5158
7	Nasal bridge depth	0.1138	25.6314	0.0933	0.0720	0.7283	0.3320	0.4385
11	Profile (sum of distances)	-541.4310	853.0640	0.1044	0.0832	0.9912	0.2340	0.0551
15	Nose curve (sum of distances)	-913.6470	1698.5300	0.1004	0.0776	0.8112	0.2910	0.3018
19	Middle curve (sum of distances)	-1136.9000	1838.0000	0.0714	0.0734	0.7697	0.2945	0.3800
27	Profile (1^{st} der,. sum of distances)	-31.4570	24.0572	0.1332	0.0724	0.5233	0.2249	0.4961
57	Tip of the nose vertical curv.	-0.2153	0.0006	0.1096	0.0791	0.9349	0.2086	0.0849
60	Nasal bridge horiz. curv.	-0.2405	0.0277	0.1332	0.0999	0.5781	0.3162	0.5051

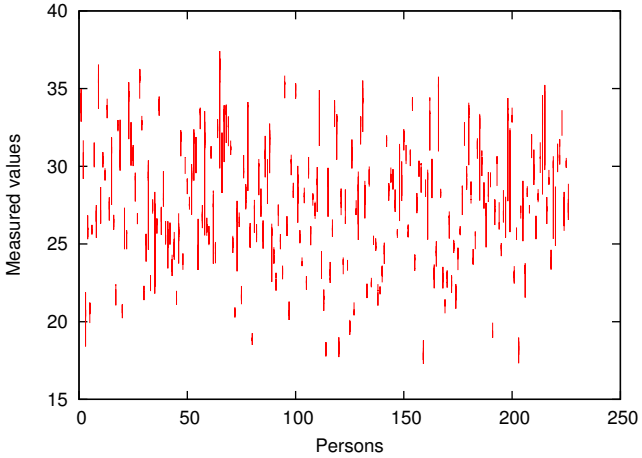


Figure 6.1: Scatter plot of measured nose depths

Table 6.3: Cross correlation of the binning candidates

	1	2	3	4	5	6	7	11	15	19	27	57	60
1	1.00	-0.02	0.21	0.31	-0.55	0.34	-0.24	0.09	0.33	0.21	0.29	0.34	0.34
2	-0.02	1.00	0.57	-0.76	0.84	0.90	0.78	-0.71	-0.92	-0.91	-0.24	-0.72	-0.68
3	0.21	0.57	1.00	-0.20	0.38	0.69	0.19	-0.78	-0.44	-0.57	0.35	-0.23	-0.10
4	0.31	-0.76	-0.20	1.00	-0.80	-0.55	-0.68	0.51	0.78	0.71	0.57	0.90	0.64
5	-0.55	0.84	0.38	-0.80	1.00	0.57	0.77	-0.65	-0.94	-0.87	-0.35	-0.78	-0.74
6	0.34	0.90	0.69	-0.55	0.57	1.00	0.58	-0.63	-0.71	-0.73	-0.09	-0.50	-0.44
7	-0.24	0.78	0.19	-0.68	0.77	0.58	1.00	-0.48	-0.83	-0.83	-0.29	-0.63	-0.83
11	0.09	-0.71	-0.78	0.51	-0.65	-0.63	-0.48	1.00	0.67	0.79	-0.29	0.53	0.37
15	0.33	-0.92	-0.44	0.78	-0.94	-0.71	-0.83	0.67	1.00	0.94	0.30	0.74	0.76
19	0.21	-0.91	-0.57	0.71	-0.87	-0.73	-0.83	0.79	0.94	1.00	0.09	0.67	0.73
27	0.29	-0.24	0.35	0.57	-0.35	-0.09	-0.29	-0.29	0.30	0.09	1.00	0.54	0.36
57	0.34	-0.72	-0.23	0.90	-0.78	-0.50	-0.63	0.53	0.74	0.67	0.54	1.00	0.58
60	0.34	-0.68	-0.10	0.64	-0.74	-0.44	-0.83	0.37	0.76	0.73	0.36	0.58	1.00

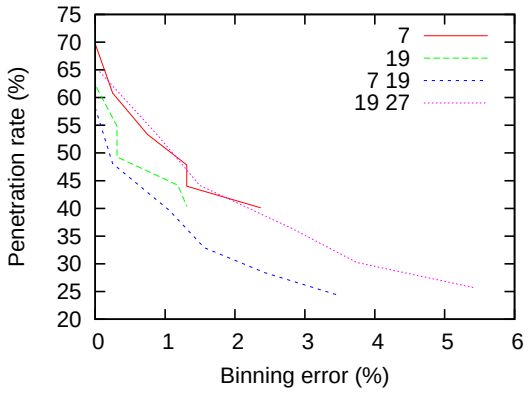


Figure 6.2: Binning error-penetration rate curves

For the binning purposes is good to choose candidates with mutual coefficients whoose absolute value is close to zero.

To avoid the bin bordering errors, when both probe and database template has the value of selected binning criteria very similar, but they are situated in different bins, bin overlapping is present. Not only the calculated bin but also its surroundings are searched. Although this will increase the penetration rate, the binning error is rapidly reduced.

The example results of the database binning for various combinations are in Figure 6.2. (7 – the nasal bridge depth, 19 – the distance of horizontal profile curve to the corresponding curve on the mean face and 27 – the correlation between the 1^{st} derivation of vertical profile curve and corresponding curve on the mean face). The parameter of every curve is the number of bins which the dataset is divided to. For 5 bins per dimension no false bin assigment was detected. However, the penetration rate was still quite high. The combination of metrics number 7 and 19 brings the penetration rate 40% together with 1% binning error. This means that when the database binning is employed during the identification process, in average less than a half of templates can be searched. On the other hand, the equal error rate will never be lower than 1%.

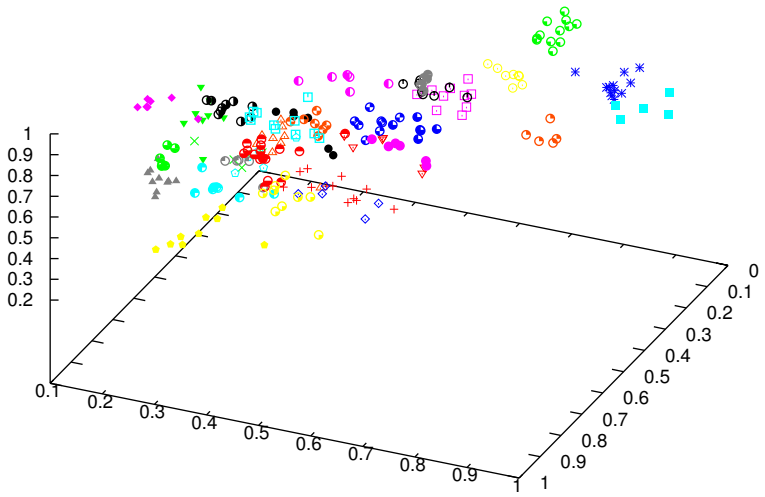


Figure 6.3: Distribution of selected anatomical-Bertillon features in the face space

The influence of database binning on the specific recognition methods will be described further in the text is sections 6.3.1, 6.4.2 and 6.5.1.

6.3 Recognition based on anatomical-Bertillon features

Beside the binning purposes, the anatomical-Bertillon features extracted from the properly oriented face can be also used for direct face comparison. All 61 measurements can be seen as a feature vector and distance between two vectors is therefore a similarity score.

The suitability of anatomical features for the face recognition is illustrated in Figure 6.3. Only three features were selected for each face. The scans from the same person create clusters in the face space.

However, as it was already mentioned in chapter 6.2, not all the features have the same discriminating ability. As a starting point, performance evaluation was run using both Euclidian and city block distance on entire "spring2004" part of the FRGC dataset. The results are in Table 6.4. Significantly better performance was achieved utilizing the city block distance.

Table 6.4: Initial performance evaluation for the recognition based on the anatomical features

Used distance function	achieved EER
Euclidian	0.15
City block	0.12

One method, how to improve the recognition performance is to select only the best

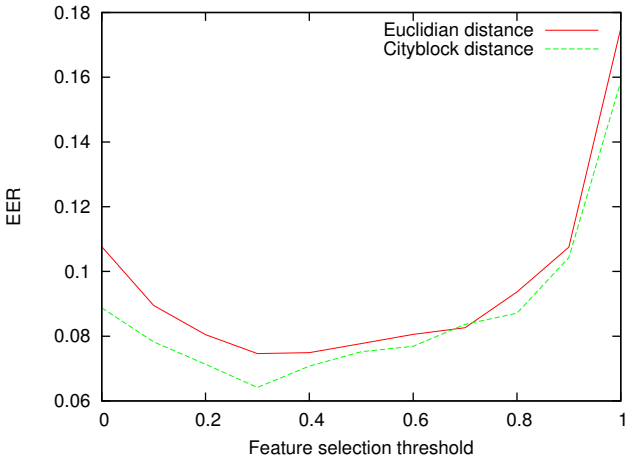


Figure 6.4: Evaluation of recognition based on anatomical features

discriminating features. The selection of those features is based on their discriminative potential score that was calculated previously for the binning suitability purposes (see Table 6.1). In the test, all features with score $s \geq t$, where $t \in \{0.0, 0.1, \ldots, 1.0\}$, were consecutively selected and the recognition performance was calculated. The result is in Figure 6.4. The best performance (6.4% EER) was achieved by using the city block distance on the features with score $s \geq 0.3$. 20 out of 61 features met this criteria.

Another performance improving approach is the usage of weighted distance. The weights vector w was derived from the feature discriminative potential scores and subsequently normalized, such the the greatest component has value 1 and the lowest component has value 0. The distance functions are then defined as:

$$d_{Euclidian}(X,Y) = \sqrt{\sum_{i=1}^{N} w_i(x_i - y_i)^2} \tag{6.2}$$

$$d_{city\ block}(X,Y) = \sum_{i=1}^{N} w_i|x_i - y_i| \tag{6.3}$$

The usage of the weighted distance function has the positive influence on the recognition performance (See Table 6.5). The results are better than the initial performance, but not as good as the results that were achieved when only the best features were selected.

Table 6.5: Evaluation of recognition based on the anatomical features with weighted distance function

Distance function	achieved EER
Euclidian	0.09
City block	0.07

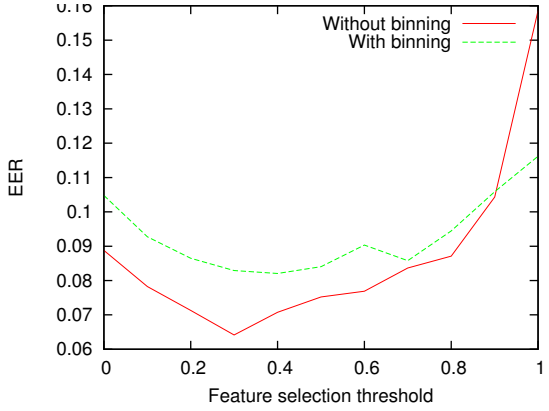


Figure 6.5: Impact of database binning on the anatomical-based recognition

6.3.1 Binning

The influence of database binning to the recognition based on the anatomical features was also preformed. The result is in Figure 6.5. Two best binning candidates and division into 10 bins for each was used. The database binning does not provide any improvement. By contrast, the performance is lower. This is probably caused by the fact, that the same criteria are used for recognition as well as for binning and therefore no additional recognition information is gained.

6.4 Histogram-based features recognition

The performance of the face recognition method using histogram-based features was also evaluated on the "spring2004" part of the FRGC dataset. The initial batch evaluation was performed on templates, that were extracted from the range images, smoothed range images and the pointclouds. The division into 10, 20, 25, 30, 40, 50, 60 rows and 2, 6, 10 bins in each row was utilized during the feature extraction. Part of the results of the initial batch test is in Table 6.6. The best performance was achieved with the smoothed range image as an input and division into 10 rows and 6 bins in each row. City block distance function provides better results regardless on the other parameters.

Table 6.6: Initial evaluation of histogram-based recognition

Source data	Rows	Bins in rows	Distance function	EER
Smoothed range image	10	6	city block	11.9%
Smoothed range image	10	6	Euclidian	15.8%
Range image	10	6	city block	12.2%
Range image	10	6	Euclidian	16.4%
Pointcloud	50	10	city block	14.8%
Pointcloud	50	10	Euclidian	20.0%

The weighted city block distance were used in order to improve achieved results. Not all the components of the feature vector has the same discrimative ability, therefore the same discriminative potential score evaluation as in case of the anatomical-based recognition was

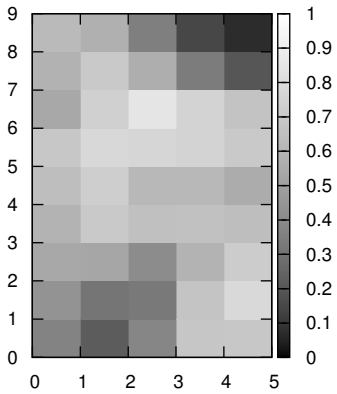


Figure 6.6: Illustration of the weight vector gained from the genetic optimization

performed on each component. These scores was subsequently normalized into the range $(0; 1)$ and this values were used directly as a weight coefficients. The performance of the recognition system increased and the EER was lowered to 9.1%.

6.4.1 Genetic optimization

The analyticaly gained weight vector was afterwards taken as an input into the genetic optimization process. The initial population contained 50 slightly modified weight vectors. The training set of the genetic optimization was formed by 200 face scans and as a fitting function was utilized achieved EER.

In every iteration 20 best candidates were selected and new 30 candidates were created by crossing of the parents with 0.1 probability of mutation. Although after 25 iterations EER on the training set was 7.1%, when the same weight vector was applied on the entire dataset, the EER was 8.6%. However, this result is better than original 9.1%.

In Figure 6.6 is illustrated the resulting weight vector. Lighter areas correspond to the components that has bigger influence on the distance, while the darker areas correspond to the insignificant components.

6.4.2 Binning

The influence of the database binning on the performance of the histogram-based recognition is illustrated in Figure 6.7. Two-dimensional binning criteria utilizing two best binning candidates were used. The consecutive binning into 5,6,. . . ,15 bins per dimension were used. At the level of 7 bins per dimension, the lowest EER = 9.3% was achieved.

6.5 PCA

The PCA evaluation relied on the batch processing of all combinations of input image sizes and curvature map types. Entire “spring 2004” portion of the FRGC dataset was tested. Generally better result was achieved using only selected part of a face, that covers only

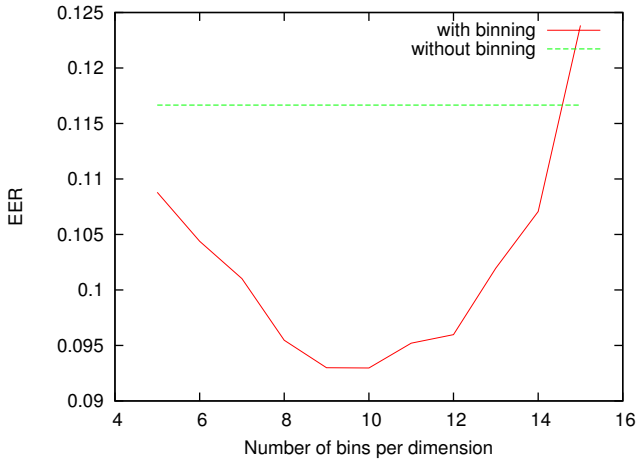


Figure 6.7: Influence of database binning to the histogram-based recognition performance

regions not so much influenced with facial mimics. Although the amount of contained information is not so high as in the case when entire face is taken as an input, performance is more accurate due to the fact that areas around mouth may cause recogntion errors.

Additionally, two approaches of generating eigenvectors from the training set were tested. In first case, the training set was formed from 50 different persons, in second case 200 face scans formed the training set, but 5 to 12 scans per one person were included.

Some of the results are in Table 6.7. The best result (EER = 8.8%) was achieved with utilizing the Euclidian distance function in projected face space from the shape index map.

Table 6.7: Evaluation of PCA

Source data	image size	Distance function	Number of eigenvectors	EER
Shape index map	reduced	Euclidian	49	8.8%
Shape index map	reduced	city block	49	9.2%
Shape index map	entire face	Euclidian	49	9.3%
Shape index map	entire face	city block	49	9.2%
Range image	entire face	Euclidian	199	14.4%
Range image	entire face	city block	199	19.8%

6.5.1 Binning

The influence of the database binning on the performance of the PCA is illustrated in Figure 6.8. Two-dimensional binning criteria utilizing two best binning candidates were used. The consecutive binning into 5,6,...,15 bins per dimension were used. At the level of 8 bins per dimension, the lowest EER = 7.2% was achieved.

6.6 Fusion

There are two main possibilities how to combine results from the utilized recognition methods. The Fusion can be made on the decision or on the score level (see Figures 6.9 and

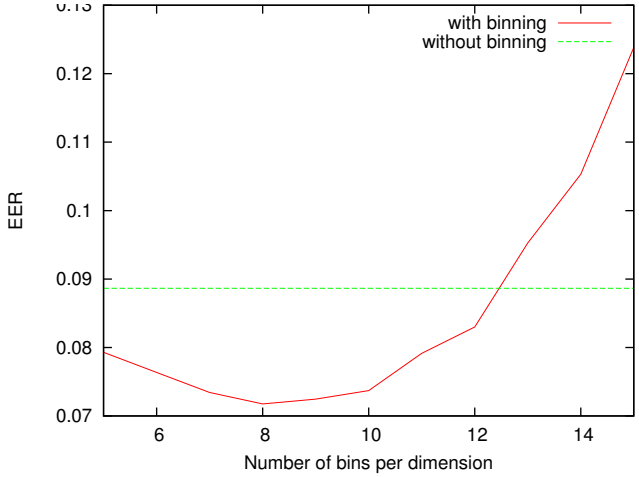


Figure 6.8: Influence of database binning to the PCA recognition performance

6.10).

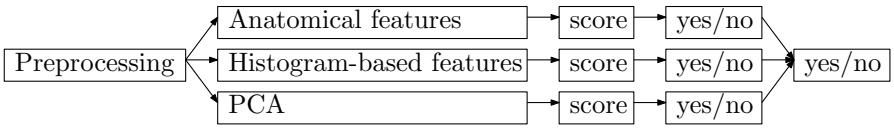


Figure 6.9: Fusion on the decision level

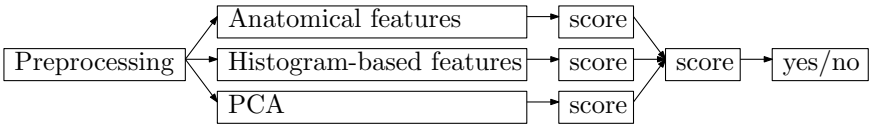


Figure 6.10: Fusion on the score level

In the evaluation, the fusion of following methods were performed:

- The anatomical-based recogntion when the 20 best facial metrics were used
- The histogram-based recognition with optimized weight coefficients for the distance calculation
- The eigenface method performed on the shape index images

6.6.1 Decision-level fusion

When the decision-level fusion is employed in the recognition pipeline, each utilized recognition algorithm decides if the probe provided by a data subject matches with some template stored in the database. These binary values are subsequently fused. The logical OR, AND or decision based on majority was used. In the tests, the decision threshold for each algorithm was choosen in such way that the EER for the selected method was achieved. The results are in Table 6.8.

Utilizing the AND function provides very secure system with very low false acceptance rate, but on the other hand, high false rejection rate of this system may cause problems

Table 6.8: Decision-level fusion

Fusion function	FAR	FRR
OR	19.2%	0.8%
AND	0.6%	19.0%
majority	4.6%	4.4%

to the users. Additional tuning of the recognition system can be done on the recognition algorithms with shifting particular decision threshold.

6.6.2 Score-level fusion

In order to fuse comparison scores from the employed algorithms, the score normalization described in the recognition pipeline proposal in chapter 4 was performed. Table 6.9 shows measured scores for each algorithm. The $score_{EER}$ stands for the comparison score when the EER is achieved and the $score_{max}$ stands for the maximal measured distance between two templates in the database.

Table 6.9: Comparison scores for each employed algorithm

Algorithm	$score_{EER}$	$score_{max}$	EER
Anatomical recognition	1.80	10.00	6.4%
Histogram-based recognition	1.18	5.12	8.6%
PCA	3 281.44	10 254.50	8.8%

The weighted score fusion s for scores $s_1, s_2, \ldots, s_n$ is defined as:

$$s = w_1 \cdot norm(s_1) + w_2 \cdot norm(s_2) + \cdots + w_n \cdot norm(s_n) \tag{6.4}$$

where $norm(s_i)$ is score normalization and $w_1, w_2, \ldots, w_n$ are weights derived from equal error rates of each utilized method in following way. The weight should reflect the method precision, therefore it is higher if the EER_i of corresponding method i is lower. Additionally, the sum of weights is equal to 1. The computation of weight vector component w_i is as follows:

$$\begin{aligned}
1 &= w_1 + w_2 + \cdots + w_n \\
1 &= \frac{(1 - EER_1)}{x} + \frac{(1 - EER_2)}{x} + \cdots + \frac{(1 - EER_n)}{x} \\
x &= \sum_{i=1}^{n} (1 - EER_i) \\
w_i &= \frac{1 - EER_i}{x} \\
w_i &= \frac{1 - EER_i}{\sum_{i=1}^{n} (1 - EER_i)}
\end{aligned}$$

Table 6.10: Weights for score-level fusion

Algorithm	w_i
Anatomical recognition	0.339
Histogram-based recognition	0.331
PCA	0.330

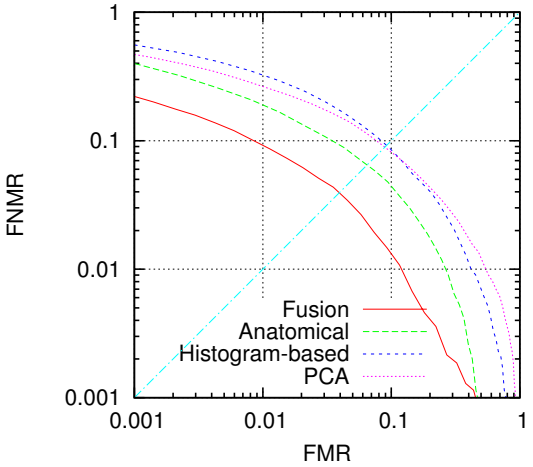


Figure 6.11: DET curves of fused system and its components

Calculated weights for the employed methods are in Table 6.10.

The equal error rate achieved with the score-level fusion is 3.9%. Although the methods independently does not provide so good recognition performance, it has emerged that by their fusion it is possible to create a system that is better than every of employed method. The DET curve of this system together with DET curves of its components is in Figure 6.11.

Finally, the database binning was added to the system. Like in previous tests, two best binning candidates were selected and consecutively $5, 6, \ldots, 15$ bins per one dimension were utilized. The recognition performance was not improved in any case. By contrast, the precision was progressively decreasing, as the number of bins per dimension was increasing (see Figure 6.12).

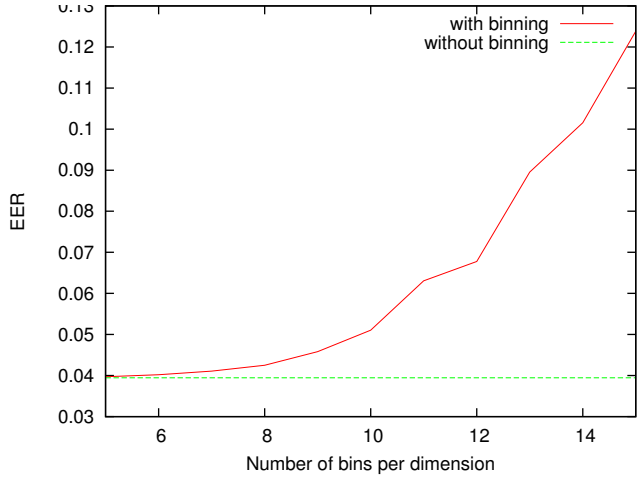


Figure 6.12: Influence of database binning to the performance of the fused system

Chapter 7

Conclusion

In this thesis, the introduction to biometric face recognition was provided. Essential terms related to the face recognition and biometrics in general were described. The comparison of the two-dimensional and three-dimensional techniques was made. The methods of detecting facial landmarks were mentioned as well as various techniques of face recognition were described.

The approach of three-dimensional face recognition algorithm, that includes the landmark detection, face normalization and recognition itself, was described in the chapter 4. The multialgorithmic system was proposed and subsequently implemented and its performance evaluated. This is described in chapters 5 and 6.

The text below describes the goals that were established or revealed in this thesis:

Database binning – The database binning has emerged as a good way how to speed up the identification process, as there is no need to browse and compare every template in the database. This contribution can be expressed as a penetration rate. On the other hand, binning influence on the recognition accuracy is not always positive. The binning features can be used directly for the comparison and the recognition performance gained from this approach is better then if the binning is applied.

Face recognition based on anatomical features – The recognition based on the anatomical-Bertillon features of the face outperformed both histogram-based recogntion and PCA. Further improvement can be achieved with the usage of a weighted distance function or feature selection, both based on the analytical evaluation of feature vector components.

Improvement of histogram-based recognition through the genetic algorithm – The application of the genetic algorithm is a good approach, how to improve the comparison score calculation. If the training set is sufficiently large and the weight vector is not too specific for the training data, EER is lowered with the usage of genetic-optimized weight coefficients.

Optimal input of PCA – Various inputs of the principal component analysis was tested and as the best solution shape index map of the face was choosen.

Fusion – If the methods independently do not provide so good recognition performance, their fusion can create a system that is better than every of employed methods.

7.1 Further work

The orientation normalization process based on the landmark detection makes the system independent on the head position and rotation, but the recognition performance is still affected by facial expressions. This was compensated in the later part of the recognition pipeline, when not entire face, but only its most rigid parts are used for the feature extraction. However, by this approach a lot of possibly significant information is lost. The other solution, that can be done in further work, is to estimate the current facial expression and shape the face surface to its neutral form.

Recognition inaccuracy can be also lowered, when the Fisherface recognition method (described in chapter 2.4.3) is used instead of the PCA. Fisherface takes also advantage of inner-class information in a training set and can provide better results than PCA.

Face recognition and especially its three-dimensional variant has became very evolving research area in last years. Some of today bests algorithms are even better than human [23], but the biometric face recognition is still open for further improvements.

Bibliography

[1] International civil aviation organization. http://www.icao.int/.

[2] Harmonized biometric vocabulary. http://www.3dface.org/media/vocabulary.html.

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

[4] A. K. Jain, P. Flynn, and A. A. Ross. *Handbook of Biometrics*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[5] T. D. Heseltine. *Face Recognition: Two-Dimensional and Three-Dimensional Techniques*. PhD thesis, The University of York, 2005.

[6] At&t database of faces. http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html, 1994.

[7] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6):559–572, 1901.

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

[9] V. Perlibakas. Automatical detection of face features and exact face contour. *Pattern Recogn. Lett.*, 24(16):2977–2985, 2003.

[10] I. Craw, D. Tock, and A. Bennett. Finding face features. In *ECCV '92: Proceedings of the Second European Conference on Computer Vision*, pages 92–96, London, UK, 1992. Springer-Verlag.

[11] Z. Liu, Z. You, A. K. Jain, and Y. Wang. Face detection and facial feature extraction in color image. *Computational Intelligence and Multimedia Applications, International Conference on*, 0:126, 2003.

[12] M. Turk and A. Pentland. Face recognition using eigenfaces. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–591, 1991.

[13] M. Haag and J. Romberg. Eigenvectors and eigenvalues. Technical report, 2009.

[14] G. A. Korn and T. M. Korn. *Mathematical Handbook for Scientists and Engineers: Definitions, Theorems, and Formulas for Reference and Review*. Dover Publications, 2 revised edition edition, 2000.

[15] L. Fritsh. Metoda pca a její implementace v jazyce c++. Technical report, Czech Technical University in Prague, Faculty of Electrical Engineering, 2007.

[16] Y. Adini, Y. Moses, and S. Ullman. Face recognition: the problem of compensating for changes in illumination direction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:721–732, 1997.

[17] M. Turk and A. Pentland. Eigenfaces for recognition. *J. Cognitive Neuroscience*, 3(1):71–86, 1991.

[18] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: recognition using class specific linearprojection. *Pattern Analysis and Machine Intelligence*, 19:711–720, 1997.

[19] R. Rak, V. Matyáš, and Z. Říha. *Biometrie a identita člověka - ve forenzních a komerčních aplikacích*. Grada, 2009.

[20] D. Bild, G. Bok, and O. Taofifenua. Face recognition with neural networks. Technical report, Northwestern University, 2007.

[21] N. Intrator, D. Reisfeld, and Y. Yeshurun. Face recognition using a hybrid supervised/unsupervised neural network. *Pattern Recogn. Lett.*, 17(1):67–76, 1996.

[22] D. Modrow, C. Laloni, G. Doemens, and G. Rigoll. 3d face scanning systems based on invisible infrared coded light. In *ISVC (1)*, pages 521–530, 2007.

[23] P. Jonathon Phillips, Patrick J. Flynn, Todd Scruggs, Kevin W. Bowyer, Jin Chang, Kevin Hoffman, Joe Marques, Jaesik Min, and William Worek. Overview of the face recognition grand challenge. pages 947–954, 2005.

[24] X. Lu, D. Colbry, and A. K. Jain. Three-dimensional model based face recognition. In *ICPR 04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR 04) Volume 1*, pages 362–366, Washington, DC, USA, 2004. IEEE Computer Society.

[25] A. B. Moreno and A. Sanchez. Gavabdb: A 3d face database. *Proc. 2nd COST Workshop on Biometrics on the Internet: Fundamentals, Advances and Applications, Ed. Univ. Vigo*, pages 77–82, 2004.

[26] Wikipedia, the free encyclopedia. http://en.wikipedia.org/.

[27] M. P. Segundo, C. Queirolo, O. R. P. Bellon, and L. Silva. Automatic 3d facial segmentation and landmark detection. In *ICIAP '07: Proceedings of the 14th International Conference on Image Analysis and Processing*, pages 431–436, Washington, DC, USA, 2007. IEEE Computer Society.

[28] M. H. Mahoor and M. Abdel-Mottaleb. Face recognition based on 3d ridge images obtained from range data. *Pattern Recognition*, 42(3):445–451, 2009.

[29] G. Pan, S. Han, Z. Wu, and Y. Wang. 3d face recognition using mapped depth images. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, page 175, Washington, DC, USA, 2005. IEEE Computer Society.

[30] A. Colombo, C. Cusano, and R. Schettini. 3d face detection using curvature analysis. *Pattern Recogn.*, 39(3):444–455, 2006.

[31] D. M. Weinstein. The analytic 3-d transform for the least-squared fit of three pairs of corresponding points. Technical report, University of Utah, 1998.

[32] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *Int. J. Comput. Vision*, 13(2):119–152, 1994.

[33] X. Zhou, H. Seibert, C. Busch, and W. Funk. A 3d face recognition algorithm using histogram-based features. *Eurographics Workshop on 3D Object Retrieval*, pages 65–71, 2008.

[34] Multi-modal and other multi-biometric fusion. Technical report, ISO/IEC, 2007.

[35] A. J. Mansfield and J. L. Wayman. Best practices in testing and reporting performance of biometric devices. Technical report, National Physical Laboratory, 2002.

[36] Mathematical morphology. http://en.wikipedia.org/wiki/Mathematical_morphology. last visited: 5 April 2010.

[37] D. A. Field. Laplacian smoothing and delaunay triangulations. *Communications in Applied Numerical Methods*, 4:709–712, 1988.

Appendix A

Implementation in C++

The entire recognition pipeline was implemented in C++ language. For the graphic user interface, Qt toolkit is used and the graphic visualization is utilized by OpenGL. For the image processing openCV library is employed and the geometrical computation in three-dimensional space relies on the CGAL library.

Every face scan is implemented as an instance of class `Face`. This class provides methods for loading the pointcloud representation of the face and preprocessing, like landmark detection and orientation normalization.

The modularity of the system is achieved by the usage of inheritance. The base class `FeatureExtractorBase` defines interface that each recognition method should implement. The result of the feature extraction is stored in an instance of the class `TemplateVect`, that can be directly used for comparison.

The example of the entire comparison pipeline is in the code below. The probe scan (`probe`) is compared to the template stored in the database (`templateInTheDatabase`).

```
Template templateInTheDatabase;

// loading the probe scan
Face probe("/path/to/file");

// initial preprocessing
probe.calculateDepthMap();
probe.removeSpikes();
probe.calculateCurvatures();

// landmark detection
probe.landmarkDetection();

// orientation normalization
probe.orientationNormalization();
probe.additionalLandmarks();

// extraction of the histogram-based features
// area selection:
// 50 pixels on both sides from the nose tip
// 80 pixels up from the nose tip
```

```
// 20 pixels down from the nose tip
// 10 rows and 6 bins in each row
HistogramRectangleRange featureExtractor(50, 80, 20, 10, 6);
TemplateVect probeFeatureVector =
    featureExtractor.getFeatureVector(&probe);

// comparison
double score = Util::euclidianDistance(
    templateInTheDatabase.templateVect.vect,
    probeFeatureVector.vect,
    probeFeatureVector.len);
```

Appendix B

Complete Results of Performance Evaluation

B.1 Anatomical features

Table B.1: Evaluation of anatomical features extracted after iterative orientation normalization
Best results are marked with green color

	Feature	Value range		Intraclass var.			Interclass var.	Score
		min.	max.	μ	s	m	c	
1	Nose width	30.0000	54.0000	0.1381	0.0852	0.5000	0.3096	0.5863
2	Nose depth	17.2859	37.4124	0.0954	0.0690	0.3967	0.4179	0.8569
3	Nose height	32.0663	64.2384	0.1263	0.0730	0.5254	0.3023	0.5776
4	Nose sharpness	1.4580	2.3431	0.1601	0.0768	0.4958	0.2457	0.5131
5	Nose depth/Nose width	0.3850	0.9588	0.1184	0.0647	0.3874	0.4095	0.8390
6	Nose volume	9210.0700	46119.0000	0.0909	0.0695	0.4495	0.3253	0.7155
7	Bridge of the nose depth	2.8760	24.9746	0.0992	0.0524	0.2801	0.3771	0.9453
8	Profile (cross cor.)	0.6721	0.9979	0.0736	0.1015	0.6979	0.1164	0.2434
9	Profile (Pearsons prod.)	0.1768	0.9891	0.1197	0.1065	0.7168	0.1547	0.2117
10	Profile (Haus. dist.)	2.0014	32.6685	0.1479	0.1161	0.7919	0.1943	0.1384
11	Profile (sum of dist.)	-471.7160	585.6140	0.1425	0.0727	0.4453	0.3017	0.6413
12	Nose curve (cross cor.)	0.9840	0.9999	0.1884	0.1336	0.8265	0.2103	0.0618
13	Nose curve (Pearsons prod.)	0.9408	0.9995	0.2026	0.1481	0.9331	0.2092	-0.0747
14	Nose curve (Haus. dist.)	1.2102	27.9800	0.1679	0.0966	0.6637	0.2854	0.3573
15	Nose curve (sum of dist.)	-909.2180	946.7610	0.1322	0.0651	0.3414	0.4489	0.9101
16	Middle curve(cross cor.)	0.9702	0.9999	0.0621	0.0706	0.9785	0.1115	0.0004
17	Middle curve (Pearsons prod.)	0.6630	0.9993	0.0454	0.0692	0.9957	0.1020	-0.0083
18	Middle curve (Haus. dist.)	1.0901	24.2626	0.1330	0.0839	0.9137	0.3198	0.1894
19	Middle curve (sum of dist.)	-1002.2800	1158.6700	0.0862	0.0438	0.2327	0.4073	1.0446
20	Eyes curve (cross cor.)	0.9569	0.9998	0.1061	0.1101	0.9526	0.1311	-0.0377
21	Eyes curve (Pearsons prod.)	0.2463	0.9970	0.1551	0.1459	0.8982	0.1542	-0.0450
22	Eyes curve (Haus. dist.)	1.7857	21.0575	0.1958	0.1017	0.7186	0.3313	0.3152
23	Eyes curve (sum of dist.)	-1282.5600	1057.9100	0.1109	0.0470	0.2617	0.3572	0.9376
24	Profile (1^{st} der., cross cor.)	0.4467	0.9856	0.1429	0.1087	0.6757	0.1691	0.2417
25	Profile (1^{st} der., Pearsons prod.)	0.4460	0.9861	0.1347	0.1064	0.6613	0.1582	0.2557
26	Profile (1^{st} der., Haus. dist.)	0.3062	2.7675	0.1679	0.1401	0.9563	0.1509	-0.1134
27	Profile (1^{st} der., sum of dist.)	-41.4310	21.7782	0.1294	0.0685	0.4052	0.2288	0.6257
28	Nose curve (1^{st} der., cross cor.)	0.8195	0.9988	0.1796	0.1596	0.8874	0.2011	-0.0254
29	Nose curve (1^{st} der., Pearsons prod.)	0.8212	0.9991	0.1785	0.1594	0.8902	0.2020	-0.0260
30	Nose curve (1^{st} der., Haus. dist.)	0.1607	2.7890	0.1555	0.1573	0.8232	0.2008	0.0648
31	Nose curve (1^{st} der., sum of dist.)	-11.5453	17.1174	0.1735	0.0788	0.5849	0.1732	0.3359
32	Middle curve(1^{st} der., cross cor.)	0.5725	0.9982	0.0621	0.0890	0.9872	0.1082	-0.0301
33	Middle curve (1^{st} der., Pearsons prod.)	0.6174	0.9983	0.0654	0.0941	0.9864	0.1115	-0.0344
34	Middle curve(1^{st} der., Haus. dist.)	0.1170	2.2129	0.1295	0.1528	0.9644	0.1854	-0.0614
35	Middle curve (1^{st} der., sum of dist.)	-9.0598	30.1430	0.1311	0.0684	0.8117	0.1474	0.1361
36	Eyes curve (1^{st} der., cross cor.)	0.1159	0.9951	0.1203	0.1215	0.9723	0.1430	-0.0711
37	Eyes curve (1^{st} der., Pearsons prod.)	0.1094	0.9951	0.1162	0.1186	0.9768	0.1409	-0.0706
38	Eyes curve (1^{st} der., Haus. dist.)	0.1928	2.4904	0.1847	0.1848	0.8945	0.1713	-0.0927
39	Eyes curve (1^{st} der., sum of dist.)	-23.7641	26.2429	0.1422	0.0753	0.5108	0.1962	0.4679
40	Profile (2^{nd} der., cross cor.)	0.4725	0.9826	0.1885	0.1186	0.8318	0.1656	0.0268
41	Profile (2^{nd} der., Pearsons prod.)	0.4690	0.9825	0.1849	0.1156	0.8321	0.1668	0.0342
42	Profile (2^{nd} der., Haus. dist.)	0.0365	0.3455	0.1488	0.0910	0.8512	0.1289	0.0378
43	Profile (2^{nd} der., sum of dist.)	-2.2320	1.4703	0.1543	0.1217	0.7422	0.1351	0.1169
44	Nose curve (2^{nd} der., cross cor.)	0.4475	0.9931	0.1747	0.1678	0.9763	0.1797	-0.1391
45	Nose curve (2^{nd} der., Pearsons prod.)	0.4248	0.9936	0.1648	0.1520	0.8501	0.1787	0.0118
46	Nose curve (2^{nd} der., Haus. dist.)	0.0314	0.4648	0.1701	0.1760	0.9279	0.2077	-0.0662
47	Nose curve (2^{nd} der., sum of dist.)	-2.0216	4.4294	0.1031	0.1355	1.0000	0.1846	-0.0540
48	Middle curve(2^{nd} der., cross cor.)	0.5441	0.9887	0.1266	0.1555	0.8854	0.1398	-0.0277

Continued on next page

	Feature	min.	max.	μ	s	m	c	
49	Middle curve (2^{nd} der., Pearsons prod.)	0.5751	0.9905	0.1335	0.1530	0.8656	0.1519	-0.0003
50	Middle curve(2^{nd} der., Haus. dist.)	0.0357	0.2788	0.1253	0.1772	0.9612	0.1595	-0.1042
51	Middle curve (2^{nd} der., sum of dist.)	-0.7679	2.9125	0.0748	0.1266	0.8001	0.1914	0.1899
52	Eyes curve (2^{nd} der., cross cor.)	0.1918	0.9611	0.2044	0.1780	0.8881	0.1676	-0.1029
53	Eyes curve (2^{nd} der., Pearsons prod.)	0.1986	0.9604	0.2096	0.1653	0.8064	0.1854	0.0041
54	Eyes curve (2^{nd} der., Haus. dist.)	0.0454	0.3712	0.1466	0.1505	0.9064	0.1378	-0.0657
55	Eyes curve (2^{nd} der., sum of dist.)	-0.8291	3.2984	0.1431	0.1373	0.8451	0.1970	0.0715
56	Tip of the Nose horizontal curvature	-0.2761	-0.0426	0.1125	0.0919	0.6267	0.2851	0.4539
57	Tip of the Nose vertical curvature	-0.2196	-0.0738	0.1705	0.0895	0.5112	0.2904	0.5192
58	Nose horizontal curvature	-12.9980	-4.2981	0.1983	0.0932	0.6923	0.2600	0.2762
59	Nose vertical curvature	-0.0240	0.0273	0.1601	0.1007	0.7497	0.2598	0.2494
60	Bridge of the nose horiznontal curvature	-0.2399	-0.0203	0.1441	0.1038	0.6084	0.3843	0.5279
61	Bridge of the nose vertical curvature	0.0182	0.0942	0.1119	0.0724	0.6939	0.2610	0.3828

Table B.2: Evaluation of anatomical features extracted after non-iterative orientation normalization

	Feature	Value range		Intraclass var.			Interclass var.	Score
		min.	max.	μ	s	m	c	
1	Nose width	19.0000	53.0000	0.0966	0.0667	0.6765	0.2334	0.3937
2	Nose depth	5.7592	36.9078	0.0611	0.0771	0.8848	0.2690	0.2460
3	Nose height	32.6760	75.1553	0.1020	0.0730	0.7231	0.2416	0.3434
4	Nose sharpness	1.4788	2.3166	0.1696	0.0886	0.5915	0.2611	0.4114
5	Nose depth/Nose width	0.1516	0.9482	0.0876	0.0685	0.8554	0.2892	0.2777
6	Nose volume	1843.8200	42721.0000	0.0687	0.0668	0.6400	0.2913	0.5158
7	Bridge of the nose depth	0.1138	25.6314	0.0933	0.0720	0.7283	0.3320	0.4385
8	Profile (cross cor.)	0.7838	0.9975	0.0879	0.1054	0.7734	0.1318	0.1651
9	Profile (Pearsons prod.)	0.2550	0.9867	0.1199	0.1151	0.7990	0.1734	0.1394
10	Profile (Haus. dist.)	2.1719	25.9789	0.1534	0.1138	0.9148	0.2235	0.0415
11	Profile (sum of dist.)	-541.4310	853.0640	0.1044	0.0832	0.9912	0.2340	0.0551
12	Nose curve (cross cor.)	0.3636	0.9997	0.0237	0.0689	0.9866	0.1005	0.0213
13	Nose curve (Pearsons prod.)	0.1958	0.9988	0.0626	0.0837	0.9553	0.1055	0.0040
14	Nose curve (Haus. dist.)	1.2339	46.8718	0.1521	0.0857	0.8320	0.2150	0.1452
15	Nose curve (sum of dist.)	-913.6470	1698.5300	0.1004	0.0776	0.8112	0.2910	0.3018
16	Middle curve(cross cor.)	0.7804	0.9999	0.0437	0.0782	0.9765	0.1025	0.0040
17	Middle curve (Pearsons prod.)	-0.0226	0.9991	0.0822	0.0995	0.9459	0.1099	-0.0176
18	Middle curve (Haus. dist.)	1.0763	30.5597	0.2046	0.1050	0.7934	0.2947	0.1917
19	Middle curve (sum of dist.)	-1136.9000	1838.0000	0.0714	0.0734	0.7697	0.2945	0.3800
20	Eyes curve (cross cor.)	0.8256	0.9998	0.0487	0.0820	0.9943	0.1035	-0.0215
21	Eyes curve (Pearsons prod.)	-0.0435	0.9960	0.1775	0.1563	0.9582	0.1677	-0.1243
22	Eyes curve (Haus. dist.)	1.6653	32.3699	0.1729	0.0954	0.6307	0.2510	0.3520
23	Eyes curve (sum of dist.)	-1383.4500	981.4200	0.1065	0.0746	0.7895	0.3651	0.3945
24	Profile (1^{st} der., cross cor.)	0.4217	0.9795	0.1459	0.1228	0.9158	0.1690	-0.0155
25	Profile (1^{st} der., Pearsons prod.)	0.4237	0.9822	0.1410	0.1200	0.9113	0.1592	-0.0131
26	Profile (1^{st} der., Haus. dist.)	0.3253	2.8862	0.1490	0.1185	0.9664	0.1490	-0.0849
27	Profile (1^{st} der., sum of dist.)	-31.4570	24.0572	0.1332	0.0724	0.5233	0.2249	0.4961
28	Nose curve (1^{st} der., cross cor.)	0.2863	0.9974	0.0940	0.1206	0.9958	0.1194	-0.0910
29	Nose curve (1^{st} der., Pearsons prod.)	0.2869	0.9977	0.0902	0.1208	0.9960	0.1169	-0.0901
30	Nose curve (1^{st} der., Haus. dist.)	0.1992	4.4187	0.1926	0.1911	0.9187	0.1868	-0.1156
31	Nose curve (1^{st} der., sum of dist.)	-19.7394	35.0606	0.2390	0.1004	0.7145	0.2304	0.1764
32	Middle curve(1^{st} der., cross cor.)	0.0670	0.9980	0.0767	0.1254	0.9736	0.1094	-0.0663
33	Middle curve (1^{st} der., Pearsons prod.)	0.0684	0.9988	0.0695	0.1277	0.9846	0.1067	-0.0752
34	Middle curve(1^{st} der., Haus. dist.)	0.1287	3.6552	0.1642	0.1940	0.8762	0.1759	-0.0585
35	Middle curve (1^{st} der., sum of dist.)	-22.2797	30.1312	0.2649	0.1202	0.8665	0.2424	-0.0092
36	Eyes curve (1^{st} der., cross cor.)	-0.3260	0.9920	0.1353	0.1537	0.9607	0.1267	-0.1231
37	Eyes curve (1^{st} der., Pearsons prod.)	-0.3333	0.9922	0.1318	0.1564	0.9709	0.1240	-0.1351
38	Eyes curve (1^{st} der., Haus. dist.)	0.2081	3.4599	0.2418	0.2106	0.8865	0.1814	-0.1575
39	Eyes curve (1^{st} der., sum of dist.)	-26.6995	24.8295	0.2642	0.1027	0.6181	0.2311	0.2461
40	Profile (2^{nd} der., cross cor.)	0.5155	0.9794	0.2326	0.1397	0.9019	0.1743	-0.1000
41	Profile (2^{nd} der., Pearsons prod.)	0.5136	0.9794	0.2295	0.1368	0.9179	0.1721	-0.1120
42	Profile (2^{nd} der., Haus. dist.)	0.0459	0.2756	0.2003	0.1024	0.8161	0.1666	0.0477
43	Profile (2^{nd} der., sum of dist.)	-2.3440	1.6981	0.1274	0.1031	0.6419	0.1468	0.2743
44	Nose curve (2^{nd} der., cross cor.)	0.0573	0.9933	0.1878	0.1503	0.9861	0.1740	-0.1502
45	Nose curve (2^{nd} der., Pearsons prod.)	0.0474	0.9943	0.1775	0.1392	0.9846	0.1729	-0.1285
46	Nose curve (2^{nd} der., Haus. dist.)	0.0284	0.5787	0.2634	0.2112	0.9585	0.2429	-0.1902
47	Nose curve (2^{nd} der., sum of dist.)	-1.7917	4.2317	0.1900	0.1773	0.9204	0.1823	-0.1053
48	Middle curve(2^{nd} der., cross cor.)	-0.0376	0.9873	0.1192	0.1638	0.9812	0.1178	-0.1464
49	Middle curve (2^{nd} der., Pearsons prod.)	-0.0691	0.9884	0.1095	0.1492	0.9799	0.1171	-0.1215
50	Middle curve(2^{nd} der., Haus. dist.)	0.0356	0.4252	0.1844	0.2487	0.9573	0.1517	-0.2387
51	Middle curve (2^{nd} der., sum of dist.)	-0.7619	3.6253	0.1424	0.1900	0.7640	0.1840	0.0876
52	Eyes curve (2^{nd} der., cross cor.)	-0.4029	0.9525	0.1946	0.1663	0.9788	0.1687	-0.1710
53	Eyes curve (2^{nd} der., Pearsons prod.)	-0.3500	0.9572	0.1935	0.1566	0.9802	0.1789	-0.1515
54	Eyes curve (2^{nd} der., Haus. dist.)	0.0438	0.4618	0.2359	0.2176	0.8598	0.1731	-0.1402
55	Eyes curve (2^{nd} der., sum of dist.)	-0.4753	4.1919	0.2155	0.1790	0.8210	0.1921	-0.0234
56	Tip of the Nose horizontal curvature	-0.2666	0.0267	0.0880	0.0636	0.6221	0.2358	0.4622
57	Tip of the Nose vertical curvature	-0.2153	0.0006	0.1096	0.0791	0.9349	0.2086	0.0849
58	Nose horizontal curvature	-23.3556	-4.9102	0.1020	0.0772	0.9957	0.1722	-0.0027
59	Nose vertical curvature	-0.0227	0.0399	0.1439	0.1001	0.8775	0.2343	0.1127
60	Bridge of the nose horiznontal curvature	-0.2405	0.0277	0.1332	0.0999	0.5781	0.3162	0.5051
61	Bridge of the nose vertical curvature	0.0000	0.0958	0.0920	0.0692	0.5751	0.2120	0.4757

Table B.3: binning performanceBest results are marked with green color

Number of bins per dimension	Utilized binning candidates	Binning error (%)	Penetration rate (%)
5	7-15	0.37	55.07
6	7-15	0.94	45.13
7	7-15	2.06	38.60
8	7-15	3.30	31.02
9	7-15	4.36	27.32
10	7-15	6.05	22.69
11	7-15	8.35	19.59
12	7-15	9.23	17.51
13	7-15	11.78	15.25
14	7-15	13.65	13.59
15	7-15	16.40	12.03
5	7-19	0.00	58.17
6	7-19	0.25	48.08
7	7-19	1.06	39.65
8	7-19	1.56	32.92
9	7-19	2.43	28.37
10	7-19	3.49	24.28
11	7-19	4.86	20.78
12	7-19	5.05	18.68
13	7-19	7.73	16.30
14	7-19	8.73	14.49
15	7-19	10.29	12.81
5	7-23	0.06	60.88
6	7-23	0.56	50.63
7	7-23	1.00	41.76
8	7-23	2.56	35.69
9	7-23	3.43	31.56
10	7-23	4.93	26.94
11	7-23	7.36	23.56
12	7-23	7.86	20.49
13	7-23	11.66	18.71
14	7-23	13.47	16.00
15	7-23	16.15	14.50
5	15-19	0.37	60.24
6	15-19	0.69	49.92
7	15-19	1.56	43.65
8	15-19	2.37	37.44
9	15-19	4.18	31.45
10	15-19	5.11	28.01
11	15-19	6.61	23.81
12	15-19	7.73	22.00
13	15-19	8.92	19.62
14	15-19	9.66	17.62
15	15-19	12.09	15.91
5	15-23	0.44	58.40
6	15-23	1.00	48.11
7	15-23	1.68	41.03
8	15-23	3.30	34.23
9	15-23	5.42	29.62
10	15-23	6.73	25.16
11	15-23	9.98	21.53
12	15-23	10.85	19.08
13	15-23	14.03	16.95
14	15-23	15.27	14.96
15	15-23	19.39	13.38
5	19-23	0.06	62.67
6	19-23	0.31	52.40
7	19-23	0.69	44.25
8	19-23	1.56	38.00
9	19-23	3.43	33.11
10	19-23	4.11	28.51
11	19-23	6.42	24.60
12	19-23	6.55	22.46
13	19-23	9.66	19.63
14	19-23	10.54	17.49
15	19-23	13.09	15.55
5	7-15-19	0.37	51.36
6	7-15-19	0.94	40.76
7	7-15-19	2.31	33.43
8	7-15-19	3.55	26.27
9	7-15-19	5.42	21.78
10	7-15-19	7.04	17.81
11	7-15-19	9.29	14.42
12	7-15-19	10.35	12.68
13	7-15-19	13.09	10.67
14	7-15-19	14.90	9.14
15	7-15-19	17.77	7.78
5	7-15-23	0.44	50.85
6	7-15-23	1.25	40.28
7	7-15-23	2.31	32.59
8	7-15-23	4.55	25.48
9	7-15-23	6.48	21.56
10	7-15-23	8.54	17.12
11	7-15-23	12.34	14.09
12	7-15-23	13.40	11.87

Continued on next page

Number of bins per dimension	Utilized binning candidates	Binning error (%)	Penetration rate (%)
13	7-15-23	17.83	10.14
14	7-15-23	19.95	8.38
15	7-15-23	24.38	7.21
5	7-19-23	0.06	53.93
6	7-19-23	0.56	43.04
7	7-19-23	1.31	34.34
8	7-19-23	2.81	27.82
9	7-19-23	4.43	23.32
10	7-19-23	6.05	18.99
11	7-19-23	8.92	15.66
12	7-19-23	9.35	13.47
13	7-19-23	13.84	11.45
14	7-19-23	15.40	9.52
15	7-19-23	18.58	8.15
5	15-19-23	0.44	54.99
6	15-19-23	1.00	43.76
7	15-19-23	1.93	36.33
8	15-19-23	3.62	29.98
9	15-19-23	6.42	24.52
10	15-19-23	7.86	20.57
11	15-19-23	11.03	16.64
12	15-19-23	12.03	14.79
13	15-19-23	15.34	12.52
14	15-19-23	16.46	10.72
15	15-19-23	20.76	9.22
5	7-15-19-23	0.44	48.25
6	7-15-19-23	1.25	37.13
7	7-15-19-23	2.56	29.23
8	7-15-19-23	4.80	22.48
9	7-15-19-23	7.42	18.07
10	7-15-19-23	9.54	14.12
11	7-15-19-23	13.28	10.98
12	7-15-19-23	14.40	9.27
13	7-15-19-23	19.08	7.55
14	7-15-19-23	20.95	6.09
15	7-15-19-23	25.50	5.00

Table B.4: Performance evaluation of recognition based on anatomical features
Best results are marked with green color

Feature selection threshold	Binning candidates	Number of bins per dimension	Distance function	EER
0	-	-	Euclidian	0.1077
0.1	-	-	Euclidian	0.0896
0.2	-	-	Euclidian	0.0805
0.3	-	-	Euclidian	0.0746
0.4	-	-	Euclidian	0.0749
0.5	-	-	Euclidian	0.0777
0.6	-	-	Euclidian	0.0806
0.7	-	-	Euclidian	0.0826
0.8	-	-	Euclidian	0.0937
0.9	-	-	Euclidian	0.1075
1	-	-	Euclidian	0.1750
0	-	-	city block	0.0888
0.1	-	-	city block	0.0782
0.2	-	-	city block	0.0713
0.3	-	-	city block	0.0642
0.4	-	-	city block	0.0708
0.5	-	-	city block	0.0752
0.6	-	-	city block	0.0769
0.7	-	-	city block	0.0836
0.8	-	-	city block	0.0871
0.9	-	-	city block	0.1043
1	-	-	city block	0.1586
0	7-19	5	city block	0.1070
0.1	7-19	5	city block	0.0893
0.2	7-19	5	city block	0.0804
0.3	7-19	5	city block	0.0745
0.4	7-19	5	city block	0.0749
0.5	7-19	5	city block	0.0778
0.6	7-19	5	city block	0.0807
0.7	7-19	5	city block	0.0828
0.8	7-19	5	city block	0.0938
0.9	7-19	5	city block	0.1075
1	7-19	5	city block	0.1686
0	7-19	7	city block	0.1078
0.1	7-19	7	city block	0.0888
0.2	7-19	7	city block	0.0803
0.3	7-19	7	city block	0.0741
0.4	7-19	7	city block	0.0755
0.5	7-19	7	city block	0.0782
0.6	7-19	7	city block	0.0815
0.7	7-19	7	city block	0.0830
0.8	7-19	7	city block	0.0937
0.9	7-19	7	city block	0.1070
1	7-19	7	city block	0.1625
0	7-19	10	city block	0.1048

Continued on next page

Feature selection threshold	Binning candidates	Number of bins per dimension	Distance function	EER
0.1	7-19	10	city block	0.0927
0.2	7-19	10	city block	0.0865
0.3	7-19	10	city block	0.0829
0.4	7-19	10	city block	0.0821
0.5	7-19	10	city block	0.0840
0.6	7-19	10	city block	0.0903
0.7	7-19	10	city block	0.0858
0.8	7-19	10	city block	0.0944
0.9	7-19	10	city block	0.1059
1	7-19	10	city block	0.1162
0	7-15-19-23	5	city block	0.1115
0.1	7-15-19-23	5	city block	0.0892
0.2	7-15-19-23	5	city block	0.0802
0.3	7-15-19-23	5	city block	0.0742
0.4	7-15-19-23	5	city block	0.0747
0.5	7-15-19-23	5	city block	0.0777
0.6	7-15-19-23	5	city block	0.0807
0.7	7-15-19-23	5	city block	0.0827
0.8	7-15-19-23	5	city block	0.0937
0.9	7-15-19-23	5	city block	0.1075
1	7-15-19-23	5	city block	0.1667
0	7-15-19-23	7	city block	0.1094
0.1	7-15-19-23	7	city block	0.0970
0.2	7-15-19-23	7	city block	0.0899
0.3	7-15-19-23	7	city block	0.0776
0.4	7-15-19-23	7	city block	0.0778
0.5	7-15-19-23	7	city block	0.0851
0.6	7-15-19-23	7	city block	0.0834
0.7	7-15-19-23	7	city block	0.0841
0.8	7-15-19-23	7	city block	0.0947
0.9	7-15-19-23	7	city block	0.1082
1	7-15-19-23	7	city block	0.1222
0	7-15-19-23	10	city block	0.1247
0.1	7-15-19-23	10	city block	0.1236
0.2	7-15-19-23	10	city block	0.1215
0.3	7-15-19-23	10	city block	0.1171
0.4	7-15-19-23	10	city block	0.1183
0.5	7-15-19-23	10	city block	0.1185
0.6	7-15-19-23	10	city block	0.1159
0.7	7-15-19-23	10	city block	0.1192
0.8	7-15-19-23	10	city block	0.1209
0.9	7-15-19-23	10	city block	0.1184
1	7-15-19-23	10	city block	0.1240

B.2 Recognition using histogram-based features

Table B.5: Performance evaluation of recognition using histogram-based features
Best results are marked with green color

Distance function	Source for feature extraction	Number of rows	Number of bins in each row	EER
Euclidian	Smoothed range image	10	2	0.163374
city block	Smoothed range image	10	2	0.135278
Euclidian	Smoothed range image	10	6	0.158218
city block	Smoothed range image	10	6	0.119346
Euclidian	Smoothed range image	10	10	0.164365
city block	Smoothed range image	10	10	0.127348
Euclidian	Smoothed range image	20	2	0.157255
city block	Smoothed range image	20	2	0.139012
Euclidian	Smoothed range image	20	6	0.155798
city block	Smoothed range image	20	6	0.123463
Euclidian	Smoothed range image	20	10	0.173966
city block	Smoothed range image	20	10	0.137883
Euclidian	Smoothed range image	25	2	0.166792
city block	Smoothed range image	25	2	0.154693
Euclidian	Smoothed range image	25	6	0.163948
city block	Smoothed range image	25	6	0.135443
Euclidian	Smoothed range image	25	10	0.183255
city block	Smoothed range image	25	10	0.146448
Euclidian	Smoothed range image	50	2	0.183378
city block	Smoothed range image	50	2	0.169648
Euclidian	Smoothed range image	50	6	0.179418
city block	Smoothed range image	50	6	0.151334
Euclidian	Smoothed range image	50	10	0.1987
city block	Smoothed range image	50	10	0.161257
Euclidian	Range image	10	2	0.192189
city block	Range image	10	2	0.166107
Euclidian	Range image	10	6	0.163911
city block	Range image	10	6	0.122306
Euclidian	Range image	10	10	0.163631
city block	Range image	10	10	0.122799
Euclidian	Range image	20	2	0.151964
city block	Range image	20	2	0.133388
Euclidian	Range image	20	6	0.154392

Continued on next page

Distance function	Source for feature extraction	Number of rows	Number of bins in each row	EER
city block	Range image	20	6	0.122384
Euclidian	Range image	20	10	0.175804
city block	Range image	20	10	0.135073
Euclidian	Range image	25	2	0.162756
city block	Range image	25	2	0.149082
Euclidian	Range image	25	6	0.16013
city block	Range image	25	6	0.130887
Euclidian	Range image	25	10	0.181751
city block	Range image	25	10	0.14628
Euclidian	Range image	50	2	0.178168
city block	Range image	50	2	0.164819
Euclidian	Range image	50	6	0.173421
city block	Range image	50	6	0.145446
Euclidian	Range image	50	10	0.194846
city block	Range image	50	10	0.15846
Euclidian	Pointcloud	20	2	0.194209
city block	Pointcloud	20	2	0.151375
Euclidian	Pointcloud	20	6	0.206094
city block	Pointcloud	20	6	0.151322
Euclidian	Pointcloud	20	10	0.202375
city block	Pointcloud	20	10	0.148557
Euclidian	Pointcloud	30	2	0.182735
city block	Pointcloud	30	2	0.136324
Euclidian	Pointcloud	30	6	0.199291
city block	Pointcloud	30	6	0.144684
Euclidian	Pointcloud	30	10	0.200245
city block	Pointcloud	30	10	0.147181
Euclidian	Pointcloud	40	2	0.173546
city block	Pointcloud	40	2	0.13442
Euclidian	Pointcloud	40	6	0.193737
city block	Pointcloud	40	6	0.142263
Euclidian	Pointcloud	40	10	0.198977
city block	Pointcloud	40	10	0.147853
Euclidian	Pointcloud	50	2	0.170278
city block	Pointcloud	50	2	0.132085
Euclidian	Pointcloud	50	6	0.189669
city block	Pointcloud	50	6	0.145704
Euclidian	Pointcloud	50	10	0.19981
city block	Pointcloud	50	10	0.147587
Euclidian	Pointcloud	60	2	0.16869
city block	Pointcloud	60	2	0.128548
Euclidian	Pointcloud	60	6	0.192755
city block	Pointcloud	60	6	0.14321
Euclidian	Pointcloud	60	10	0.200224
city block	Pointcloud	60	10	0.149098

B.3 Principal component analysis

Table B.6: Performance evaluation of PCA(Best results are marked with green color)

Input image size	Source for PCA	Number of eigenvectors	Distance function	EER
small	range image	49	city block	0.135845
small	range image	49	Euclidian	0.12057
small	principal curvature k_1	49	city block	0.156621
small	principal curvature k_1	49	Euclidian	0.148458
small	principal curvature k_2	49	city block	0.155467
small	principal curvature k_2	49	Euclidian	0.144816
small	mean curvature map	49	city block	0.163487
small	mean curvature map	49	Euclidian	0.15848
small	Gaussian curvature map	49	city block	0.189576
small	Gaussian curvature map	49	Euclidian	0.194101
small	shape index map	49	city block	0.0927494
small	shape index map	49	Euclidian	0.0886528
small	smoothed range image	49	city block	0.146314
small	smoothed range image	49	Euclidian	0.125976
big	range image	49	city block	0.14554
big	range image	49	Euclidian	0.125457
big	principal curvature k_1	49	city block	0.131512
big	principal curvature k_1	49	Euclidian	0.129577
big	principal curvature k_2	49	city block	0.15098
big	principal curvature k_2	49	Euclidian	0.146695
big	mean curvature map	49	city block	0.151014
big	mean curvature map	49	Euclidian	0.148456
big	Gaussian curvature map	49	city block	0.163436
big	Gaussian curvature map	49	Euclidian	0.169794
big	shape index map	49	city block	0.0927679
big	shape index map	49	Euclidian	0.0933407
big	smoothed range image	49	city block	0.156626
big	smoothed range image	49	Euclidian	0.128717
small	range image	199	city block	0.165655
small	range image	199	Euclidian	0.132189
small	principal curvature k_1	199	city block	0.225893
small	principal curvature k_1	199	Euclidian	0.189695

Continued on next page

Input image size	Source for PCA	Number of eigenvectors	Distance function	EER
small	principal curvature k_2	199	city block	0.220108
small	principal curvature k_2	199	Euclidian	0.189253
small	mean curvature map	199	city block	0.241179
small	mean curvature map	199	Euclidian	0.207171
small	Gaussian curvature map	199	city block	0.259375
small	Gaussian curvature map	199	Euclidian	0.240743
small	shape index map	199	city block	0.150914
small	shape index map	199	Euclidian	0.120256
small	smoothed range image	199	city block	0.183103
small	smoothed range image	199	Euclidian	0.136623
big	range image	199	city block	0.198561
big	range image	199	Euclidian	0.144185
big	principal curvature k_1	199	city block	0.194373
big	principal curvature k_1	199	Euclidian	0.158947
big	principal curvature k_2	199	city block	0.229522
big	principal curvature k_2	199	Euclidian	0.188079
big	mean curvature map	199	city block	0.225609
big	mean curvature map	199	Euclidian	0.180418
big	Gaussian curvature map	199	city block	0.236113
big	Gaussian curvature map	199	Euclidian	0.197005
big	shape index map	199	city block	0.144321
big	shape index map	199	Euclidian	0.125455
big	smoothed range image	199	city block	0.196309
big	smoothed range image	199	Euclidian	0.147406