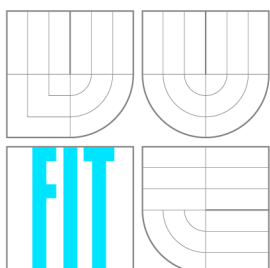


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

HYBRIDNÍ ROZPOZNÁVÁNÍ 3D OBLIČEJE

HYBRID 3D FACE RECOGNITION

DISERTAČNÍ PRÁCE

PHD THESIS

AUTOR PRÁCE

AUTHOR

ŠTĚPÁN MRÁČEK

VEDOUCÍ PRÁCE

SUPERVISOR

MARTIN DRAHANSKÝ

BRNO 2015

Abstrakt

Tato disertační práce se zabývá biometrickým rozpoznáváním 3D obličejů. V úvodu práce jsou prezentovány současné metody a techniky pro rozpoznávání. Následně je navržen nový algoritmus, který využívá tzv. multialgoritmickou biometrickou fúzi. Vstupní snímek 3D obličeje je paralelně zpracován dílčími rozpoznávacími podalgoritmy a celkové rozhodnutí o identitě nebo verifikaci identity uživatele je výsledkem sloučení výstupu těchto podalgoritmů. Rozpoznávací algoritmus byl testován na veřejně přístupné databázi 3D obličejů FRGC v 2.0 i vlastních databázích, které byly pořízeny pomocí senzorů Microsoft Kinect a SoftKinetic DS325.

Abstract

This Ph.D. thesis deals with the biometric recognition of 3D faces. Contemporary recognition methods and techniques are presented first. After that, the new recognition algorithm is proposed. It is based on the multialgorithmic fusion. The input 3D face scan is processed by the individual recognition units and the final decision about the subject identity is the result of combination of involved recognition unit outputs. Proposed approach has been tested on the publicly available FRGC v 2.0 database as well as on our own databases acquired with the Microsoft Kinect and SoftKinetic DS325 sensors.

Klíčová slova

biometrie, 3D obličej, multi-algoritmická fúze.

Keywords

biometrics, 3D face, multi-algorithmic fusion.

Citace

Štěpán Mráček: Hybrid 3D Face Recognition, disertační práce, Brno, FIT VUT v Brně, 2015

Hybrid 3D Face Recognition

Prohlášení

Prohlašuji, že jsem tuto disertační práci vypracoval samostatně pod vedením pana doc. Martina Drahanského.

.....
Štěpán Mráček
March 12, 2015

Poděkování

First of all I would like to thank my Ph.D. supervisor Martin Drahanský for his guidance, support and encouragement during my Ph.D. study. I have enjoyed valuable discussions on my work with several friends and colleagues. Therefore, I would like to thank to Jan Váňa, Radim Dvořák, Christoph Busch, and Svetlana Yanushkevich.

© Štěpán Mráček, 2015.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Contents

| | | |
|----------|--------------------------------------------------------------|-----------|
| 1 | Introduction | 3 |
| 1.1 | Chapters Overview | 3 |
| 2 | Biometrics | 4 |
| 2.1 | Fundamental Biometric Terms | 4 |
| 2.2 | Evaluating Performance of Biometric Systems | 6 |
| 2.3 | Multibiometrics | 8 |
| 3 | Obtaining Three Dimensional Data | 11 |
| 3.1 | Structured light | 11 |
| 3.2 | Commercial Solutions | 12 |
| 3.3 | Available 3D Face Databases | 14 |
| 3.3.1 | Face Recognition Grand Challenge Database | 15 |
| 3.3.2 | GavabDB | 15 |
| 3.3.3 | The University of Western Australia Face Database | 16 |
| 4 | Overview of Face Recognition Techniques | 17 |
| 4.1 | Face Recognition Difficulties | 17 |
| 4.2 | State-of-the-art | 18 |
| 4.3 | Classification of Face Recognition Methods | 19 |
| 4.4 | Projection-based Holistic Face Recognition Methods | 19 |
| 4.4.1 | Principal Component Analysis | 20 |
| 4.4.2 | Linear Discriminant Analysis | 23 |
| 4.4.3 | Independent Component Analysis | 24 |
| 4.5 | Local Binary Patterns | 27 |
| 4.6 | Active Models | 28 |
| 4.6.1 | Active Shape Models | 29 |
| 4.6.2 | Active Appearance Models | 32 |
| 4.6.3 | Active Models and Face Recognition | 34 |
| 4.7 | 3D Face Recognition | 34 |
| 4.7.1 | Representations of the 3D Face Model | 34 |
| 4.7.2 | Curvature Analysis | 35 |
| 4.7.3 | Facial Landmarks Detection | 36 |
| 4.7.4 | Face Orientation Normalization | 39 |
| 4.7.5 | Eigenfaces in Three-Dimensional Face Recognition | 39 |
| 4.7.6 | Model Based 3D Face Recognition | 40 |
| 4.7.7 | Face Recognition Using Histogram Based Features | 41 |
| 4.7.8 | Recognition Based on Facial Curves | 42 |

| | | |
|----------|-----------------------------------------------------------------|-----------|
| 5 | Proposal of the Recognition Algorithm | 44 |
| 5.1 | Generalized Recognition Pipeline for Face Recognition | 44 |
| 5.2 | Face Alignment | 45 |
| 5.2.1 | Reference Template Creation | 45 |
| 5.2.2 | Iterative Closest Point Alignment | 46 |
| 5.3 | Source Image Data | 47 |
| 5.4 | Filter Banks | 48 |
| 5.4.1 | Gabor Filter Bank | 49 |
| 5.4.2 | Gauss-Laguerre Filter Bank | 49 |
| 5.4.3 | Other Filters | 50 |
| 5.5 | Iso-geodesic curves | 52 |
| 5.6 | Feature Extraction and Metric Selection | 52 |
| 5.7 | Multi-algorithmic Score-level Fusion | 53 |
| 5.7.1 | Score normalization | 54 |
| 5.7.2 | Classifier-based fusion | 55 |
| 5.7.3 | Hill-climbing Unit Selection | 56 |
| 6 | Evaluation | 57 |
| 6.1 | Database Description and Evaluation Methodology | 57 |
| 6.2 | Face Alignment | 58 |
| 6.3 | Particular Parameters of the Recognition Algorithm | 59 |
| 6.3.1 | PCA parameters | 59 |
| 6.3.2 | Region of Interest | 60 |
| 6.3.3 | Projection and Metric Selection | 60 |
| 6.4 | Evaluation of Individual Recognition Units | 61 |
| 6.5 | Multi-algorithmic Fusion | 64 |
| 6.5.1 | Score Normalization Techniques | 64 |
| 6.5.2 | Greedy Hill-climbing Unit Selection | 64 |
| 6.5.3 | Comparison of Fusion Techniques | 66 |
| 6.6 | Comparison with the State-of-the-art | 66 |
| 6.7 | Evaluation on SoftKinetic Database | 67 |
| 6.7.1 | Finding Suitable Smoothing and Denoising Algorithm | 68 |
| 6.7.2 | Multi-Algorithmic Fusion | 69 |
| 6.7.3 | Real-World Scenarios | 71 |
| 6.8 | Kinect Evaluation | 72 |
| 6.9 | Limitations of Face Biometrics | 74 |
| 6.9.1 | Analysis of Facial Mimics | 74 |
| 6.9.2 | Tampering a Face Recognition System | 77 |
| 7 | Conclusion | 79 |
| A | Implementation | 88 |

Chapter 1

Introduction

Face recognition is one of the most frequently 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, does not 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 do 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 2D 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*). Another biometric modality that is used in the biometric passports is the iris. The face was recommended as the primary biometrics, mandatory for global interoperability in passport inspection systems, while the finger and iris were recommended as secondary biometrics to be used at the discretion of the passport-issuing state [44].

The biometric face recognition, which is the main focus of this work, has a 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 the 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 Chapters Overview

This work is about the biometric face recognition and all connected matters. In the second chapter, basic terms related to the biometrics are explained and a general biometric system is described. The methods of evaluating the biometric system performance as well as general classification of multibiometric systems are also presented in the second chapter. The third chapter describes the process of obtaining a three dimensional data. Commercial devices that are able to scan human faces are mentioned. Finally, the available three dimensional databases are described. The fourth chapter brings an overview of face recognition techniques. In the fifth chapter, a proposal of the 3D face recognition algorithm, which is the main goal of this work, is described. The sixth chapter contains an evaluation of proposed algorithm and describes the achieved results.

Chapter 2

Biometrics

In this chapter, biometrics and related terms will be explained. The definitions provided here are from the ISO standard Harmonized Biometric Vocabulary [29], the paper [33] by Jain and Ross, and the first chapter in their book Handbook of Biometrics [32].

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, refer to the characteristics that are always present on data subject's action independently. Biometric methods based on the physiological 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. Some examples of the physiological and the behavioral characteristics are in the Table 2.1.

2.1 Fundamental Biometric Terms

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

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 electronic identity.

Identification – The identification is the process when the biometric system recognizes an individual by comparing his/her characteristics with all templates stored in the biometric database. The result of the identification is the data subject's identity or “not recognized”.

Table 2.1: Examples of physiological and behavioral biometric characteristics.

| Physiological characteristics | Behavioral characteristics |
|-------------------------------|----------------------------|
| Fingerprints | Voice |
| Face | Gait (walk) |
| Iris | Lips motions |
| DNA | Signature dynamics |
| Palm veins | Keystroke dynamics |

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

Every biometric feature 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.

Regarding the 3D face recognition, the universality and collectability is fulfilled. But it should be taken into account that hairstyle, clothes covering a part of the face or glasses may be quite a challenging problem for a face-based biometric system. The uniqueness holds to some extent. Although recognition and identification based only on photographs pose no problem to humans, some people, especially twins or siblings are look-alikes. The permanence for 3D face is also not ideal. The overall appearance of the face model is stable in perspective of several months, however, it rapidly changes during the first years of life and during the late age [40].

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

Performance – Refers to the achievable identification accuracy.

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.

From many points of view, face recognition seems to be problematic (uniqueness, permanence). What makes it popular is high acceptability. Face recognition is touch-less, it can be performed with just a minimal cooperation of the user and the recognition itself is similar to the way we recognize each other. The circumvention is a problem especially for classical 2D face recognition. Many utilizations of this approach, e.g. the first implementation of Face Unlock on some Android smart-phones, does not have any liveness detection, and the biometric system can be deceived with just a printed photograph. Fooling a 3D sensor is much harder as we need precise real-size model with texture.

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 among individual scans of the same person, while the inter-class variance refers to the diversity among various persons. It is good to choose a 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 serves to 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 repeatable and distinctive numbers

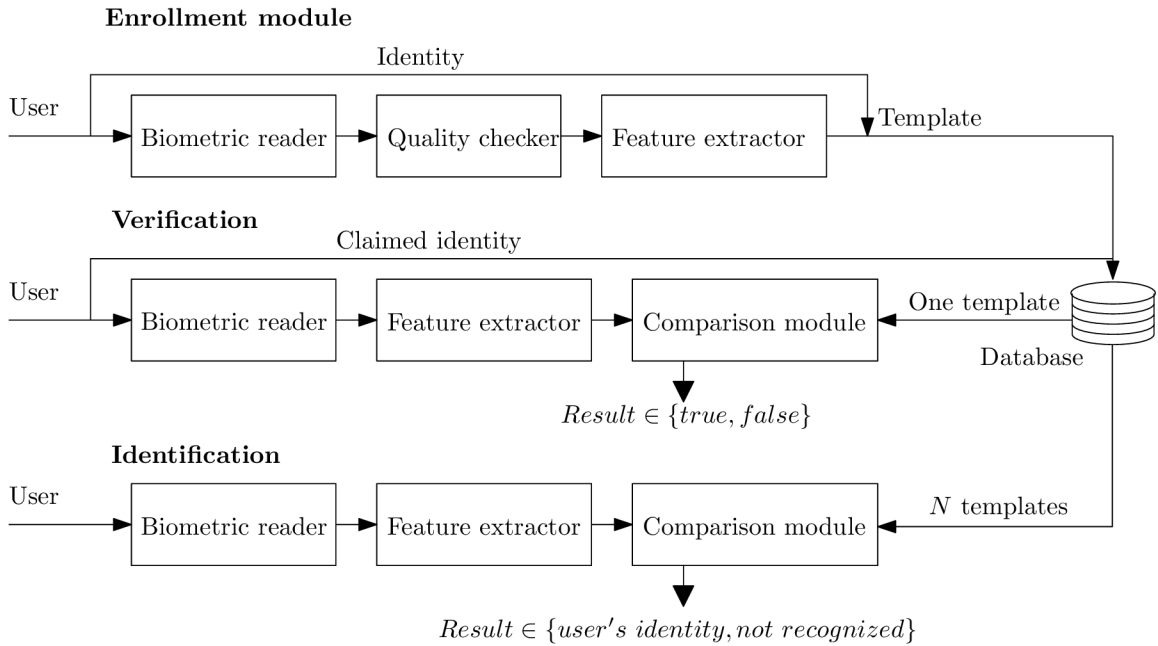


Figure 2.1: Generic biometric system.

or labels (*biometric features*) are subsequently extracted and stored in the database as a template.

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/her identity and the biometric system has to decide whether it is true or not. On the other hand, in the case of identification, the recognition system has to decide whether the data subject is registered, and if so, which template in the database belongs to him/her. While the verification is 1 : 1 comparison, the identification is 1 : N comparison. A generic biometric system is illustrated in Figure 2.1.

Comparing a template with features extracted from a scan provided by a data subject produces the comparison score denoting how the template and the extracted features are 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.

2.2 Evaluating Performance of Biometric Systems

One of the most important properties of a biometric system is how successful in recognition it is. There are two main errors that the 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 a registered person is rejected from the biometric system.

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 the distance between the gallery and the probe scan, the decision algorithm is as follows:

```

score ← getComparisonScore(probe, gallery)
if score ≥ Threshold then
    reject
else
    accept
end if

```

Four outcomes might occur:

True acceptance – The genuine person is truly recognized.

True rejection – The impostor is truly rejected.

False acceptance – The impostor is falsely 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 and also comfortable for users. This means that the goal is to minimize the false rejection and false acceptance cases. The False acceptance rate (FAR) indicates what proportion of attempts resulted in a false recognition.

$$\text{FAR} = \frac{\sum \text{different measures classified as the same}}{\sum \text{measures of various persons or instances}} \quad (2.1)$$

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

$$\text{FRR} = \frac{\sum \text{misclassified measures of the same person or instance}}{\sum \text{measures of the same person or instance}} \quad (2.2)$$

The FAR and FRR are joined together by the threshold that decides whether the data subject is accepted or not. However, a higher threshold leads to a more secure system, where the impostors are refused, but genuine data subjects are sometimes refused as well. On the other hand, a 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 a given threshold are equal. It is often used as a criteria for evaluating performance of the biometric systems. The lower the value, the better the system compared to another. The relation between the FAR and FRR is illustrated in Figure 2.2. The decision if the data subject is accepted or not is strictly based on the retrieved comparison score and the given threshold.

The relation between the FAR and FRR values at the different thresholds depicted as a curve is referred to as Detection Error Trade-off (DET). The example of DET curve is in Figure 2.3. FRR at any given FAR can be easily read from DET curve for example and thus DET provides much more information about characteristics of the biometric system than just the EER value.

There are some more terms related to the evaluation 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 refers to the biometric sensor and its quality checker, especially with the defined quality that 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.

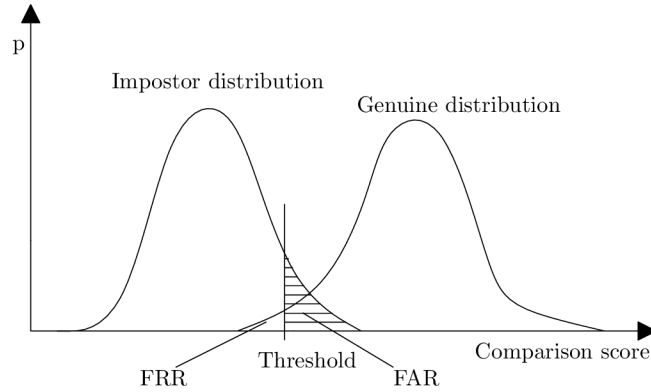


Figure 2.2: False acceptance rate and false rejection rate.

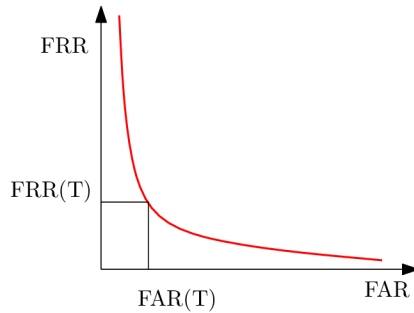


Figure 2.3: Detection error trade-off curve.

FNMR – The false non-match rate refers to the portion of the false rejected persons. Contrary to the FRR the FNMR does not include attempts that had been unsuccessful before the comparison started.

FMR – The false match rate refers to the portion of the false accepted persons. As with the FNMR, the 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.

2.3 Multibiometrics

Multibiometric systems combine the information presented by multiple biometric sensors, algorithms, samples, or units. Besides enhancing recognition performance, these systems are expected to improve population coverage and decrease the possibility of spoofing [71].

In general, a multibiometric system may be classified into one of the following six categories:

Multisensor systems employ multiple sensors to capture a single biometric trait of a data subject.

Multialgorithm systems involve multiple feature extractors and/or multiple comparison algorithms.

Multiinstance systems use multiple instances of the same body trait. For example, fingerprints of the left and right index finger are used.

Multisample systems use a single sensor in order to acquire more samples of the same biometric trait. These samples are subsequently fused together.

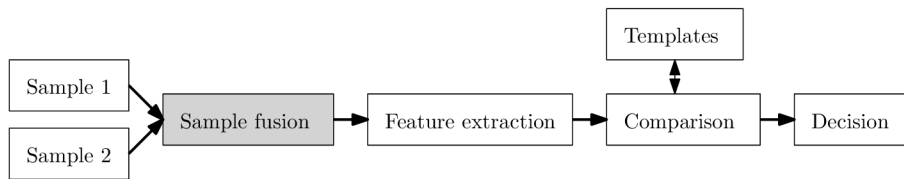
In multimodal systems, the identity is established by the evidence of multiple biometric traits, for example face and fingerprints.

Hybrid systems may combine two and more categories mentioned above.

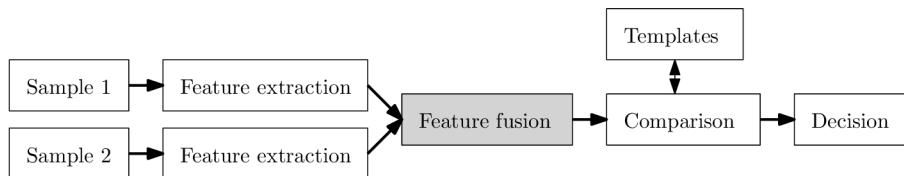
In this thesis, the emphasis is put on the multi-algorithmic systems. The single modality, 3D model of the face, is processed with various feature extraction algorithms and the resulting feature vectors are mutually compared using several different metrics. The chapter 6 shows that the utilization of multi-algorithmic approach outperforms unimodal approach significantly.

Once a multibiometric system is employed, a biometric fusion of the obtained information should be used. Based on the type of the obtained information, different levels of fusion may be defined [58]:

Sample level fusion : The fusion process fuses the collection of obtained scans from multiple sensors into a single sample:

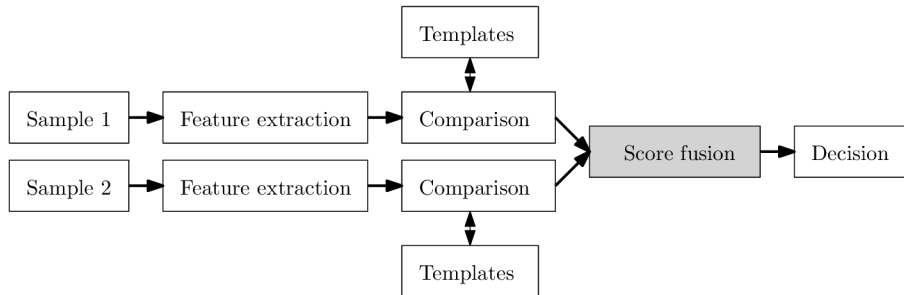


Feature level fusion Each of the employed feature extractors outputs the collection of features. The fusion process fuses obtained feature vectors into a single vector:



This fusion approach may also be applied to the multi-algorithmic systems, where each employed algorithm processes the input scan and produces a feature vector. The easiest solution is when the individual feature vectors are concatenated. However, when the feature vectors have different lengths, this yields to neglect of short feature vectors. Usually, the feature-level fusion is accomplished by some linear projection to a lower-dimensional space where the variability is preserved. More information may be found in Sections 4.4.1, 4.4.2, and 4.4.3.

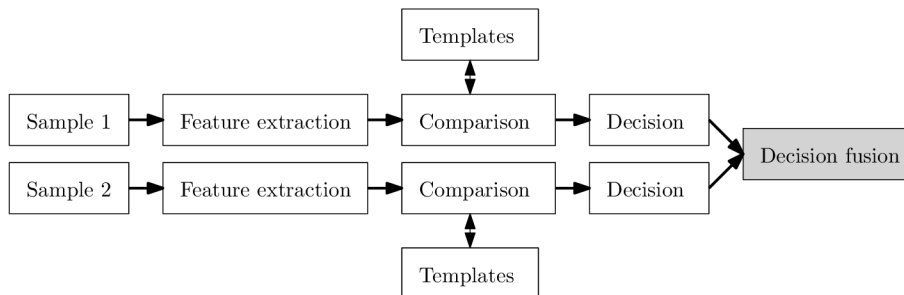
Score level fusion Each individual biometric process provides a comparison score and these scores are subsequently fused. Even this approach may be applied to the unimodal multi-algorithmic systems. The problem that comes up, when the score-level fusion is involved, is the necessity of score-normalization:



According to [59], score fusion techniques can be divided into the following three categories:

- *Transformation-based fusion* – The scores are first normalized (transformed) to a common domain and then combined. A weighted sum or just a simple product are the representatives of the transformation-based fusion.
- *Classifier-based fusion* – Scores from multiple comparison modules are treated as a feature vector and a binary classifier is constructed to discriminate genuine and impostor scores. The classifier-based fusion may be provided by the Support Vector Machines (SVM) classifier with linear kernel, Linear Discriminant Analysis (LDA), or logistic regression, for example.
- *Density-based score fusion* – This approach is based on the likelihood ratio test and it requires an explicit estimation of genuine and impostor comparison score densities, for example using Gaussian Mixture Model (GMM).

Decision level fusion Each involved biometric system provides boolean results whether the data subject is accepted or not. The fusion process fuses the output results together by boolean operators AND or OR. It may also take into account additional parameters, such as quality of samples or obtained comparison scores:



Chapter 3

Obtaining Three Dimensional Data

The classic face recognition approach utilizing 2D photographs has to deal with illumination and pose variation. This can be solved when the 3D face recognition is used, however, the biggest disadvantage of this approach is much higher acquisition costs.

On the other hand, the expansion of personal depth sensors related to the new ways of the human-computer interaction in recent years has markedly lowered the price of 3D acquiring devices for personal use, such as Microsoft Kinect 360¹ or SoftKinetic DS325² sensors.

The biggest challenge of the face recognition based on the low-cost depth sensors is the quality of acquired scans. While, for example, the Minolta Vivid or Artec 3D M scanners provide a highly precise geometry with an outstanding resolution and level of detail, the scans retrieved from the Kinect or DS325 sensors are noisy, have a low resolution and sometimes contain holes.

3.1 Structured light

3D sensors utilizing a structured light approach project a known pattern on a scanned object. The depth is calculated based on the deformation of the pattern. The most common pattern used in the structured light 3D scanning is many narrow stripes, although other strategies of pattern codification may be used [64, 75]. The pattern can be projected either in visible light or in infra-red spectra. An example of the reconstruction process is, for instance, proposed in [19], where many coloured stripes in visible light spectra are projected on the face surface. In order to minimize the misclassification between the projected lines and lines observed from the camera, De Bruijn sequence consisting of seven colors is used (see Figure 3.1). This pattern forms image $P_{pattern}$.

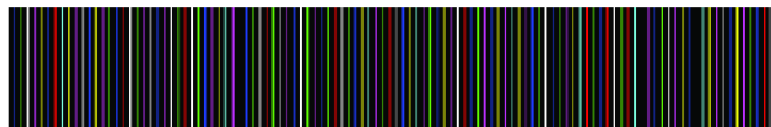


Figure 3.1: De Bruijn sequence of coloured vertical lines used in the structured light 3D scanning [19].

¹<http://www.xbox.com/kinect/>

²<http://www.softkinetic.com/products/depthsensecameras.aspx>

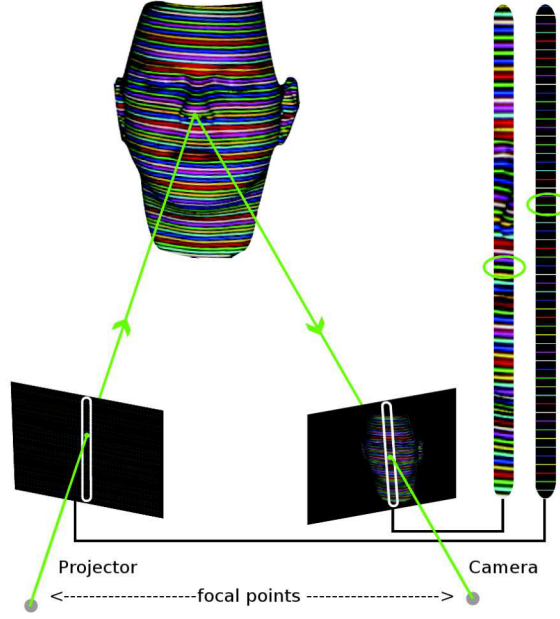


Figure 3.2: Principle of calculation of depth information with the usage of trigonometry [19].

The algorithm of retrieving 3D data from the scanned face surface is as follows:

1. Take one picture of a scanned subject illuminated with stripes $P_{colored}$ and one without stripes P_{clean} .
2. Extract projected stripes: $P_{colored} \leftarrow P_{colored} - P_{clean}$.
3. Match the stripes in $P_{colored}$ with the original pattern $P_{pattern}$. At this point, dynamic programming is used. The cost matching function that we are trying to minimize is the sum of possible color misclassification and gaps caused by the topology of the scanned surface [19]:

$$Cost = \sum_{i=1}^m colorDiff(i) + jumpWeight(i, i-1) \quad (3.1)$$

where $colorDiff(i)$ is the possible difference between i^{th} stripe in $P_{colored}$ and assigned stripe from $P_{pattern}$, $jumpWeight(i, i-1)$ returns the penalty of gap, and m is number of observed stripes in $P_{colored}$.

4. Calculate the depth information from the shift between the observed stripes and the original stripes in $P_{pattern}$. (See Figure 3.2.)

3.2 Commercial Solutions

Minolta Vivid

Minolta Vivid³ is a laser 3D scanner. Light reflected from the scanned object is acquired by CCD camera. After that the final model is calculated using the standard triangulation

³<http://www.konicaminolta.com/instruments/products/3d/non-contact/vivid910/index.html>



Figure 3.3: The examples of scans obtained with the Artec 3D M scanner. The structured light scanners have a problem with scanning shiny objects or a structure where the projected pattern is highly distorted. This problem leads to the impossibility of capturing glasses (middle scan) or a beard (right scan).

method. This scanner was, for example, used for acquiring of the FRGC database (see Section 3.3.1).

Artec 3D scanner

The Artec 3D M scanner⁴ has a flash bulb and a camera. The bulb flashes a light pattern onto the object and the CCD camera records this pattern. The distortion in the light pattern, due to the specific curvature of the object, is then translated into a 3D image by Artec software. As the user moves around the object, the light pattern changes and the software provided together with the camera recognizes these changes. The light pattern is projected onto the object with the frame rate 15 frames per second. Individual scans are joined together and subsequently form the resulting 3D model. Examples of scans obtained with the Artec 3D M scanner are in Figure 3.3.

A4 Vision Enrollment Station

The A4 Vision Enrolment Station⁵ is a specialized 3D face acquiring device that is used for enrolling users to the face recognition biometric system. It operates in infra-red spectra – it projects horizontal lines on the scanned face. Although the software and API provided together with the camera supports the BioAPI specification⁶, there is not direct access to the scanned 3D data and thus it cannot be used for developing own 3D face recognition algorithm.

Microsoft Kinect 360

The Microsoft Kinect 360 is a structured light depth sensor operating in infra-red spectra utilizing technology developed by PrimeSense, Ltd.⁷. It also contains a RGB sensor and a directional microphone. Its main purpose is to serve as the motion-sensing input device for Microsoft Xbox 360. It enables users to control and interact with their gaming console/computer without the need for a game controller through a natural user interface using

⁴http://www.artec3d.com/3d_scanners/artec-m

⁵<http://www.lid.com/pages/404-3d-face-reader>

⁶BioAPI Consortium – <http://www.bioapi.org/>

⁷<http://www.primesense.com/>

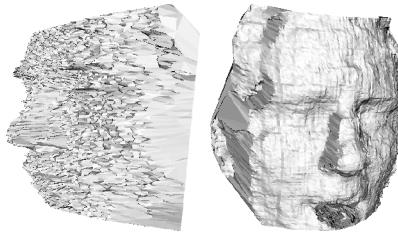


Figure 3.4: Example scans from SoftKinetic (left) and Kinect (right) sensors.

| Sensor name | Resolution | Technology | Range | Texture | Price |
|------------------------------|-----------------------------------------------|--------------------------------|-------------|---------|---------------------|
| Minolta Vivid 9i/910 | 640×480 | structured light (non-visible) | 0.6 – 1.2 m | yes | \$25,000 – \$55,000 |
| Artec 3D M scanner | uses fusion of several consecutive scans | structured light (visible) | 0.4 – 1 m | no | \$12,000 – \$22,000 |
| A4 Vision Enrollment Station | manufacturer does not provide any information | structured light (non-visible) | | yes | \$21,150 |
| Microsoft Kinect 360 | 640×480 | structured light (non-visible) | 1.2 – 3.5 m | yes | \$100 |
| Occipital Structure Sensor | 320×240 | structured light (non-visible) | 0.4 – 3.5 m | no | \$379 |
| Microsoft Kinect 2 | 512×424 | time of flight | 0.8 – 4 m | yes | \$199 |
| DepthSense SoftKinetic 325 | 320×240 | time of flight | 0.15 – 1 m | yes | \$249 |

Table 3.1: Overview of some commercial 3D sensors on the market.

gestures and spoken commands. Example of a scan acquired with the Kinect is in Figure 3.4. The new updated version of Kinect uses a wide-angle time of flight (ToF) camera.

DepthSense SoftKinetic 325

The SoftKinetic 325 is the only sensor in this list utilizing ToF technology. It resolves distance based on the known speed of light, measuring the time-of-flight of a light signal between the camera and the subject for each point of the image [21]. The primary purpose of the device is hand and finger tracking and thus interacting with devices without touching a screen, keyboard, trackball or mouse. An example scan from the SoftKinetic sensor is in Figure 3.4. The overview of all mentioned sensors is in Table 3.1.

3.3 Available 3D Face Databases

Developing a face recognition method also includes evaluating performance of the system, thus testing data are needed. Several available 3D face databases will be described in this section.

The European Association for Biometrics provides for their members an overview of publicly available 2D and 3D face databases. Individual databases are varying in many parameters, e.g. number of participating subjects, number of capturing sessions, number of samples per each subject, and total number of samples. The overview is in Table 3.2. Some of the mentioned databases are more focused on varying facial expressions (ND-2006), while others contain subjects captured from varying angles (NKCU, CASIA-3D). Interesting is also ND-TWINS which was captured at the Twins Days Festivals in Twinsburg, Ohio in 2009 and 2010.

| Type | Name | No. of subjects | No. of sessions | No. of samples | Samples/ session | Samples/ subject | Reference |
|------|------------------------|-----------------|-----------------|----------------|------------------|------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| 2D | BioSecure | 667 | 2 | | | | http://biosecure.it-sudparis.eu |
| 2D | CASIA-FaceV5 | 500 | | 2,500 | | 5 | Chinese Academy of Sciences |
| 2D | FERET Face Dataset | 1,199 | 15 | 14,126 | varies | varies | http://www.nist.gov/itl/iad/ig/feret.cfm |
| 2D | FRGC v 2.0 (2D) | 4,003 | 1 | 50,000 | 6 | | http://www.nist.gov/itl/iad/ig/frgc.cfm |
| 2D | ND-2006 Dataset | 888 | | 13,450 | | up to 63 | http://www.nd.edu/~cvrl/ |
| 2D | ND-TWINS | 435 | | 24,050 | | | CVRL/Data_Sets.html |
| 2D | NKCU | 89 | 2 | 6,589 | 37 | 74 | National Cheng Kung University |
| 2D | Sheffield Face Dataset | 20 | | 564 | | up to 64 | http://www.sheffield.ac.uk/eee/research/iel/research/face |
| 3D | CASIA-3D FaceV1 | 123 | 3 | 4,624 | | 37 or 38 | http://biometrics.idealtest.org |
| 3D | FRGC 1.0 (3D) | 557 | 3 | 4,059 | 3 | | http://www.nist.gov/itl/iad/ig/frgc.cfm |

Table 3.2: List of publicly available face databases according to EAB.

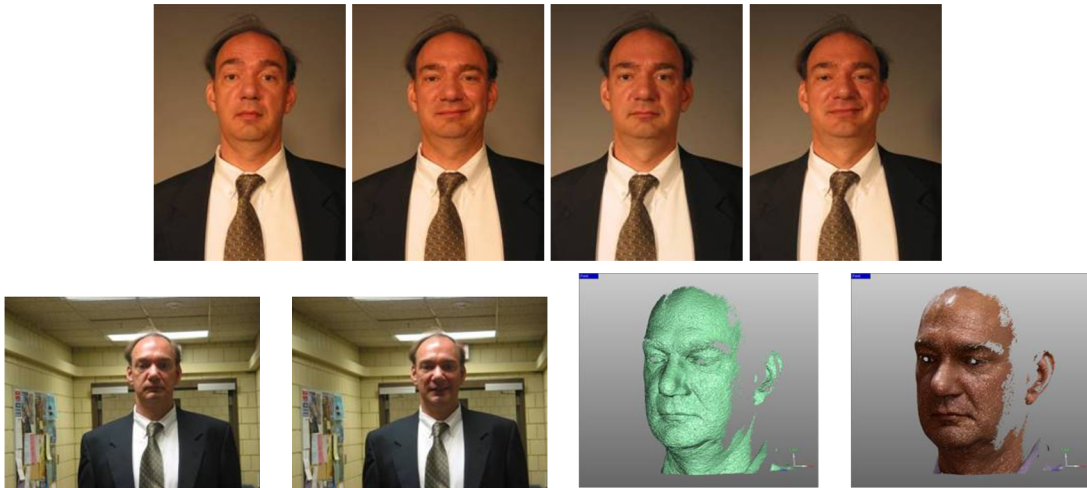


Figure 3.5: Acquired scans during one session in FRGC [66].

3.3.1 Face Recognition Grand Challenge Database

Face Recognition Grand Challenge (FRGC) database [66] 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.

Data for the FRGC were collected at the University of Notre Dame between autumn 2003 and spring 2004. Each subject had several sessions. Four images taken under controlled lighting conditions, two images at uncontrolled conditions, and one 3D scan has been acquired at each session. The example of one subject session is in Figure 3.5. The three dimensional scans were acquired by the Minolta Vivid 910 scanner.

3.3.2 GavabDB

GavabDB [49] 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 3.6.

The data in the database contains, except for the facial scans, also noise, spikes, and the data that is 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 in order to extract the face and eliminate the impact of the noise on the recognition performance.



Figure 3.6: GavabDB facial scans examples (a) and from The University of Western Australia Face Database (b).

Each subject 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.

3.3.3 The University of Western Australia Face Database

This database contains scans from 106 subjects, but 3D models of some subjects are missing because they did not allow their images to be distributed [47]. Example of some scans is in Figure 3.6.

Chapter 4

Overview of Face Recognition Techniques

In this chapter, the overview of face recognition methods will be provided. First, some essential terms related to the face recognition will be explained. Then various techniques of classical two-dimensional as well as three-dimensional approaches will be described.

The biometric recognition of faces includes the methods and algorithms for the detection of 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 [25], a special device for acquiring scans from the data subject should be purchased. This fact leads to a much higher acquisition cost. However, the enrollment may be conducted using a 3D scanner and latter verification would be fulfilled by conventional 2D cameras [10].

The face recognition is, along with the fingerprint and iris recognition, one of the most commonly used biometric techniques. It is well accepted by users due to its non-invasive 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, e.g., the localization of the nose tip, are not easy for computers. 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.

4.1 Face Recognition Difficulties

The biggest challenge of the biometric face recognition is to deal with relatively big intra-class variation, which is related to many factors, mostly varying lighting conditions, face orientation, and facial expressions. Light direction, color, and intensity have a 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 a 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 in the following text. Two pictures with varying lighting conditions of the same person are shown in Figure 4.1.



Figure 4.1: Various lighting conditions [4].

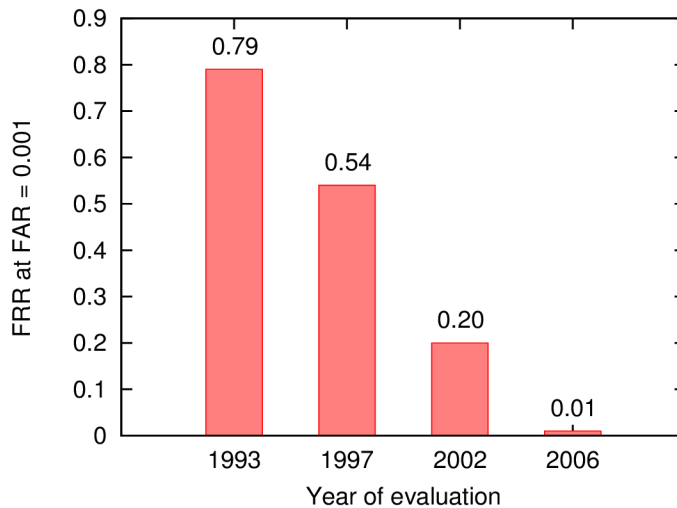


Figure 4.2: The reduction in error rate for state-of-the-art face recognition algorithms as documented through the FERET, the FRVT 2002, and the FRVT 2006 evaluations.

4.2 State-of-the-art

The developed face recognition system should be compared with other actual face recognition systems on the market. In 2006 the National Institute of Standards and Technology in the USA realized the Face Recognition Vendor Test (FRVT) [66]. It has been so far the latest in a series of large scale independent evaluations for face recognition systems. Previous evaluations in the series were the FERET, FRVT 2000, and FRVT 2002. The primary goal of the FRVT 2006 was to measure progress of prototype systems/algorithms and commercial face recognition systems since FRVT 2002. FRVT 2006 evaluated performance on high resolution still images (5 to 6 mega-pixels) and 3D facial scans.

The comprehensive report of achieved results and used evaluation methodology is described in [65]. The progress that has been achieved during the last years is depicted in Figure 4.2. The results show the achieved false rejection rate at false acceptance rate 0.001 for the best face recognition algorithms in specific years. This means that in 2006, if we admit that 0.1% will be falsely accepted as genuine persons, only 1% of users will be incorrectly rejected.

The best 3D face recognition algorithm that has been evaluated in FRVT 2006 was Vi-
 sage from the commercial portion of participated organizations [65]. The plot in Figure 4.3 shows the evaluation results of the participated organizations and their 3D face recognition algorithms. The entire FRGC dataset was divided into several subsets. The algorithms were evaluated for each subset and the results reported to the box-plot.

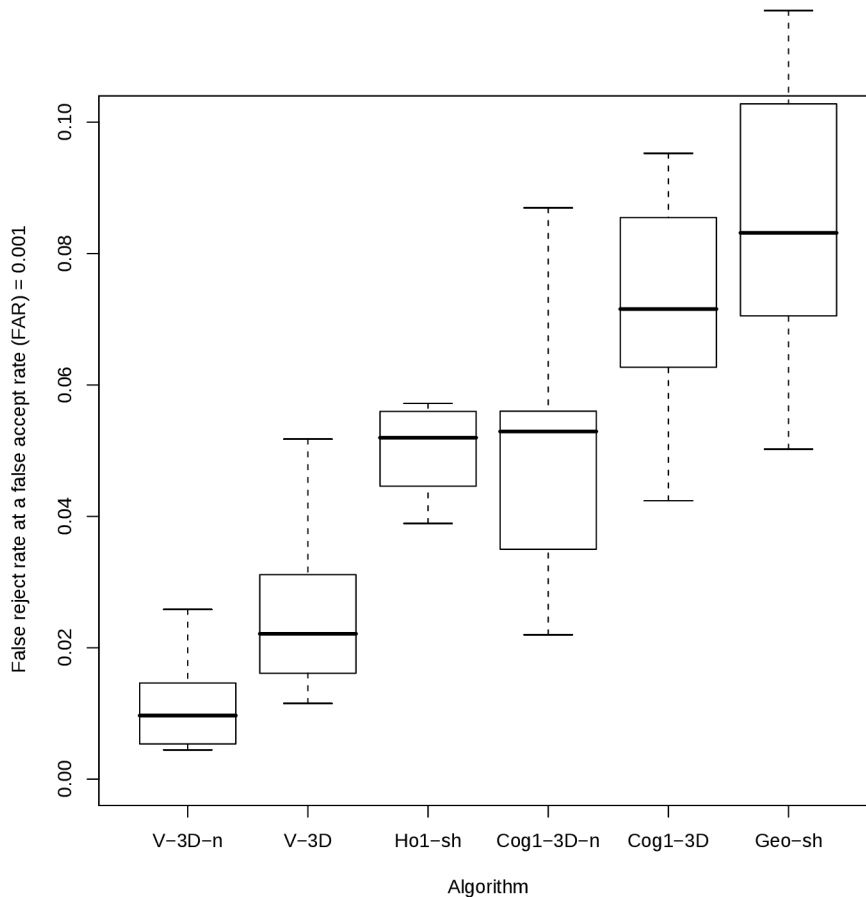


Figure 4.3: The box-plots of the evaluation of 3D face recognition algorithms participating in FRVT [65].

4.3 Classification of Face Recognition Methods

The two-dimensional face recognition as well as the three-dimensional approach can be divided into three categories – holistic, feature based, and hybrid [88]. The holistic recognition methods utilize global information from faces in order to perform face recognition. The global information is directly derived from the face representations. The feature based face recognition, conversely, uses a priori information or local features of faces to select a number of features to uniquely identify individuals. Local features may include eyes, nose, mouth, chin and head outline. The hybrid approach combines both holistic and feature based methods.

4.4 Projection-based Holistic Face Recognition Methods

Face recognition is in principle a pattern recognition. Each face is represented as a vector that could be located in a multi-dimensional face space, e.g. in the two-dimensional face recognition a face could be represented as an image with resolution 150×100 pixels. 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 away. Calculating distances between the face scans in this multi-dimensional space,

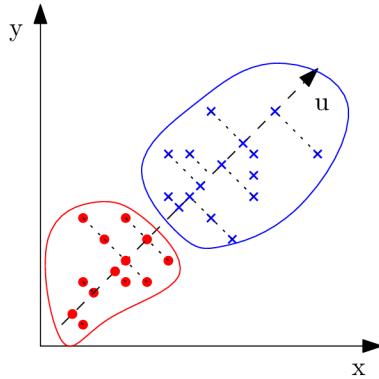


Figure 4.4: Principal component analysis. The points from the two classes are projected on the new axis u .

thus comparing faces, is very time consuming due to multi-dimensionality of the space. Moreover, there is a lot of unwanted information stored, such as a background, hair or clothes. Therefore, various techniques that decrease number of dimensions were invented. The best known are the *Principal Component Analysis* (PCA), the *Linear Discriminant Analysis* (LDA), and the *Independent Component Analysis* (ICA).

4.4.1 Principal Component Analysis

Principal component analysis was first introduced by Karl Pearson [63] 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 describe as much of the remaining variability as possible.

Model situation is shown in Figure 4.4. Two-dimensional space containing data in two classes is reduced to one-dimensional space. Each 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 the learning process. The principal component analysis seeks for direction in which the data vary the most. This could cause in some cases wrong classification. This problem is illustrated in Figure 4.5.

The eigenface method [82] is an example of the 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.

Each image is represented as column vector \mathbf{x} . First, the mean face from the set of training images is calculated. We take the set of p training images $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p$ and the mean face $\bar{\mathbf{x}}$ is calculated:

$$\bar{\mathbf{x}} = \frac{1}{p} \sum_{i=1}^p \mathbf{x}_i \quad (4.1)$$

Then the mean face image $\bar{\mathbf{x}}$ is subtracted from each training image \mathbf{x}_i

$$\mathbf{x}_i \leftarrow \mathbf{x}_i - \bar{\mathbf{x}} \quad \forall i \in (1, 2, \dots, p) \quad (4.2)$$

After that, the covariance matrix C is constructed:

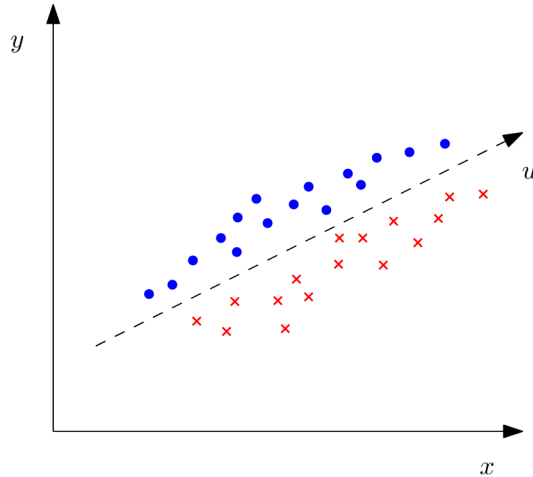


Figure 4.5: Wrong class separation with principal component analysis – all data points are projected to the new axis u and therefore the classes separation is lost.

$$\mathbf{C} = \mathbf{A} \mathbf{A}^T = [\mathbf{x}_1 \mathbf{x}_2 \dots \mathbf{x}_p] [\mathbf{x}_1 \mathbf{x}_2 \dots \mathbf{x}_p]^T \quad (4.3)$$

where \mathbf{A} stands for a matrix where each column i contains a corresponding vector \mathbf{x}_i and \mathbf{A}^T stands for transposed matrix \mathbf{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 [22]. Given a matrix \mathbf{M} , a non-zero vector \mathbf{v} is defined to be an eigenvector of the matrix if it satisfies the eigenvalue equation

$$\mathbf{M} \mathbf{x} = \lambda \mathbf{v} \quad (4.4)$$

for some scalar λ . In this situation, the scalar λ is called an eigenvalue of \mathbf{M} corresponding to the eigenvector \mathbf{v} [38].

However, the covariance matrix might be very large and thus the computation of its eigenvectors and eigenvalues would be time and memory consuming. If the amount of training images p is sufficiently smaller than the size (dimensionality) n of training images, eigenvectors and eigenvalues could be retrieved from matrix \mathbf{C}' .

$$\mathbf{C}' = \mathbf{A}^T \mathbf{A} \quad (4.5)$$

The size of matrix \mathbf{C}' is determined by the size of the training set and it is $p \times p$. The first p sorted eigenvalues of matrix \mathbf{C}' are also eigenvalues of matrix \mathbf{C} . The eigenvectors of matrix \mathbf{C} are calculated by multiplying matrix \mathbf{A}^T by matrix \mathbf{W}' . \mathbf{W}' is the matrix containing in each row one eigenvector \mathbf{w}' of the matrix \mathbf{C}' .

$$\mathbf{W} = \mathbf{A}^T \mathbf{W}' = \mathbf{A}^T \begin{bmatrix} w'_{11} & w'_{12} & \dots & w'_{1p} \\ w'_{21} & w'_{22} & \dots & w'_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ w'_{p1} & w'_{p2} & \dots & w'_{pp} \end{bmatrix} \quad (4.6)$$

The resulting matrix \mathbf{W} contains one eigenvector of the covariance matrix \mathbf{C} in each row. These eigenvectors define a set of mutually orthogonal axes within facial space, along

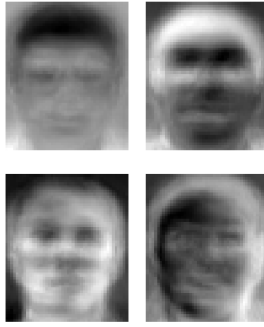


Figure 4.6: Eigenfaces obtained from AT&T face database [4].

which there is the most variance. The corresponding eigenvalues represent the degree of variance along these axes. In Figure 4.6, there are displayed some eigenvectors from AT&T face database. Due to the likeness to faces, Turk and Pentland refer them as eigenfaces [25].

Projection of the facial image \mathbf{I} to the face space is as follows: first the image is transformed to the column vector \mathbf{x} and the precomputed mean face $\bar{\mathbf{x}}$ is subtracted from the input face \mathbf{x} . Then each component ω_i is calculated by multiplying the corresponding eigenvector \mathbf{w}_i by the modified input vector.

$$y_i = \mathbf{w}_i (\mathbf{x} - \bar{\mathbf{x}}) \quad (4.7)$$

$$\mathbf{y} = \mathbf{W}^T (\mathbf{x} - \bar{\mathbf{x}}) \quad (4.8)$$

The comparison of two faces in this face space is performed by the calculation of the distance between these two faces. Various distance calculation could be used [25] on projected face images \mathbf{y}_A and \mathbf{y}_B , such as Euclidean distance:

$$d(\mathbf{y}_A, \mathbf{y}_B) = \sqrt{\sum_{i=1}^P (y_{Ai} - y_{Bi})^2} \quad (4.9)$$

the city block distance:

$$d(\mathbf{y}_A, \mathbf{y}_B) = \sum_{i=1}^P |y_{Ai} - y_{Bi}| \quad (4.10)$$

the cosine distance:

$$d(\mathbf{y}_A, \mathbf{y}_B) = 1 - \frac{\mathbf{y}_A^T \mathbf{y}_B}{\|\mathbf{y}_A\| \|\mathbf{y}_B\|} \quad (4.11)$$

or correlation distance:

$$d(\mathbf{y}_A, \mathbf{y}_B) = 1 - \frac{\sum_{i=1}^N (y_{Ai} - \bar{y}_A)(y_{Bi} - \bar{y}_B)}{\sqrt{\sum_{i=1}^N (y_{Ai} - \bar{y}_A)^2}} \quad (4.12)$$

where N is the size of input column vectors and $\|\mathbf{a}\|$ stands for the norm of the vector \mathbf{a} .

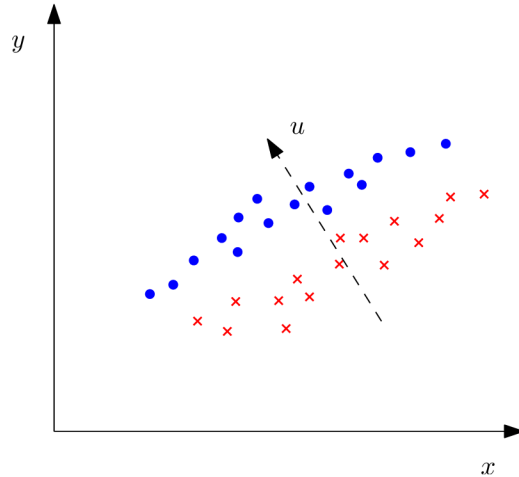


Figure 4.7: Linear Discriminant Analysis – data points are projected to the new axis u that separates both classes.

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 [1].

4.4.2 Linear Discriminant Analysis

Linear discriminant analysis (LDA), introduced by Ronald Aylmer Fisher [20], is an example of supervised learning. The class membership is taken into account during learning. LDA seeks for vectors that provide the best discrimination between classes after the projection. Therefore, the LDA is applicable to the classification problems where PCA fails (Figure 4.7).

Fisherface method is a 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. [6].

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

$$\tau = \{X_1, X_2, \dots, X_K\} \quad (4.13)$$

where K is the number of classes and $X_i = \{\mathbf{x}_1, \mathbf{x}_2, \dots\}$, where \mathbf{x}_j stands for individual picture of the person from class i with different facial expressions or taken under various lighting conditions.

First, the intra-class (within-class) distribution matrix S_W describing variation inside the classes is calculated.

$$\mathbf{S}_W = \sum_{k=1}^K \mathbf{S}_k \quad (4.14)$$

where

$$\mathbf{S}_k = \sum_{n \in C_k} (\mathbf{x}_n - \bar{\mathbf{x}}_k)(\mathbf{x}_n - \bar{\mathbf{x}}_k)^T \quad (4.15)$$

$\bar{\mathbf{x}}_k$ is the mean of class k :

$$\bar{\mathbf{x}}_k = \frac{1}{N_k} \sum_{n \in C_k} \mathbf{x}_n \quad (4.16)$$

and N_k is the number of scans in class C_k .

After that, the inter-class (between-class) distribution matrix \mathbf{S}_B is calculated. This matrix describes variation among individual persons from the training set.

$$\mathbf{S}_B = \sum_{k=1}^K N_k (\bar{\mathbf{x}}_k - \bar{\mathbf{x}})(\bar{\mathbf{x}}_k - \bar{\mathbf{x}})^T \quad (4.17)$$

where $\bar{\mathbf{x}}_k$ is the average of class X_k and $\bar{\mathbf{x}}$ is the mean of the entire dataset:

$$\bar{\mathbf{x}} = \frac{1}{N} \sum_{n=1}^N \mathbf{x}_n \quad (4.18)$$

The objective optimization criterion (Fisher criterion) is given by [9]:

$$J(\mathbf{W}) = \frac{\mathbf{W}^T \mathbf{S}_B \mathbf{S}_W}{\mathbf{W}^T \mathbf{S}_W \mathbf{S}_W} \quad (4.19)$$

The projection matrix \mathbf{W} is determined by the eigenvectors of $\mathbf{S}_W^{-1} \mathbf{S}_B$. The projection of the input image \mathbf{x} in LDA is similar to the projection in PCA:

$$\mathbf{y} = \mathbf{W}^T (\mathbf{x} - \bar{\mathbf{x}}) \quad (4.20)$$

It has been shown that fisherface method may provide better results than eigenface method [6, 25]. This is caused mainly because fisherface method takes the advantage of inter-class variation. To achieve significantly better results compared to eigenface method, good training set containing various facial expression and lighting conditions is required. On the other hand, some researches point out that LDA suffers from overtraining.

4.4.3 Independent Component Analysis

Another data projection method is *Independent Component Analysis* (ICA) first introduced by Piere Comon [15]. The definitions and equation presented in this chapter are from paper by Hyvärinen and Oja [28]. Contrary to the PCA, which seeks for the dimensions in which data vary the most, ICA is looking for the transformation of the input data that maximizes non-gaussianity. An example of this process is illustrated in Figure 4.8.

The origin of ICA comes from *Blind Signal Separation* (BSS) or *Cocktail Party Problem* [11]. Suppose that we have two statistically independent sources of speech signal at different locations $s_1(t)$ and $s_2(t)$. Both signals are recorded by two microphones situated somewhere else in the space. Each of these recorded signals is therefore a weighted sum of the original source signals:

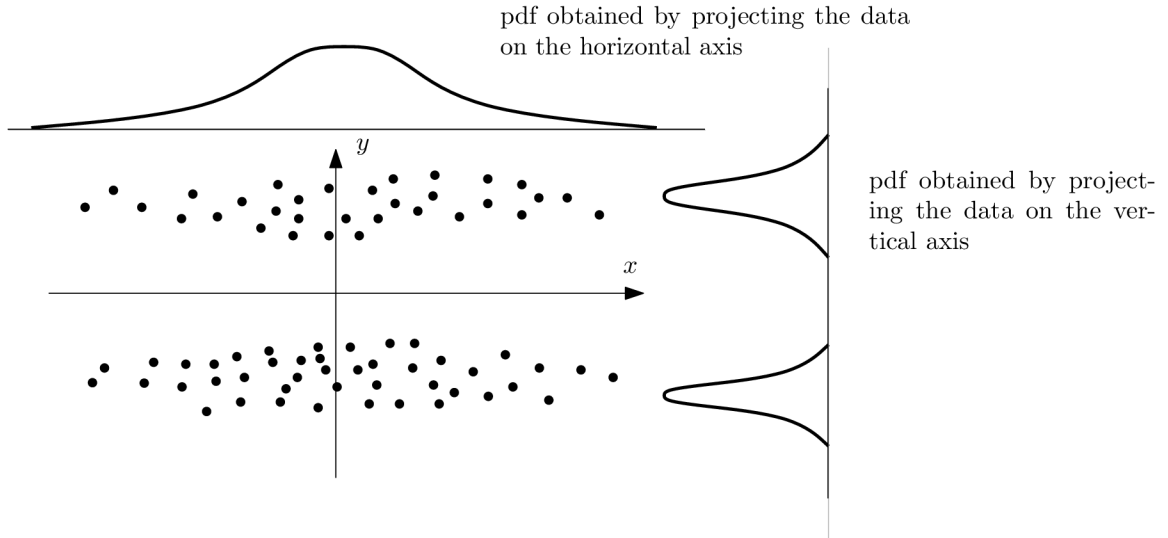


Figure 4.8: While the PCA analysis of a given example would yield into the projection on the vertical axis, ICA provides projection on the horizontal axis which offers non-gaussian probability distribution function (pdf) and thus better cluster separation.

$$\begin{aligned} x_1(t) &= a_{11}s_1(t) + a_{12}s_2(t) \\ x_2(t) &= a_{21}s_1(t) + a_{22}s_2(t) \end{aligned} \quad (4.21)$$

The main task in the Cocktail Party Problem is to reveal coefficients a_{11} , a_{12} , a_{21} , and a_{22} and estimate the original source signals $s_1(t)$ and $s_2(t)$. This problem is solvable only if the original sources are statistically independent and non-gaussian [28].

The coefficients a_{ij} form a matrix \mathbf{A} that describes an ICA model. Thus, any data vector \mathbf{x} can be reconstructed from the independent sources as:

$$\mathbf{x} = \mathbf{A}\mathbf{s}. \quad (4.22)$$

Preprocessing Steps Prior to the Computation of ICA

As well as PCA, ICA expects that data vectors have zero mean. If we have m vectors $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m$ in \mathbb{R}^N , the mean can be computed as:

$$\bar{\mathbf{x}} = E\{\mathbf{x}\} = \frac{1}{m} \sum_{i=1}^m \mathbf{x}_i \quad (4.23)$$

The mean is subtracted from each vector subsequently:

$$\mathbf{x} \leftarrow \mathbf{x} - \bar{\mathbf{x}} \quad (4.24)$$

The next step in the preprocessing for ICA is whitening the data. Each vector \mathbf{x}_i has to be linearly transformed in such a way that its components are uncorrelated and their variance equal unity. This means that the covariance matrix should be the identity matrix:

$$E\{\mathbf{x}\mathbf{x}^T\} = \mathbf{I} \quad (4.25)$$

This is possible with the usage of the eigenvalue decomposition of the covariance matrix $\Sigma = \mathbf{x}\mathbf{x}^T$ that is also the core of PCA:

$$\Sigma = \mathbf{E}\mathbf{D}\mathbf{E}^T \quad (4.26)$$

where \mathbf{E} is the orthogonal matrix containing computed eigenvectors and \mathbf{D} is diagonal matrix of corresponding eigenvalues. Each data vector \mathbf{x} is modified in the following way:

$$\mathbf{x} \leftarrow \mathbf{E}\sqrt{\mathbf{D}}\mathbf{E}^T\mathbf{x} \quad (4.27)$$

where $\sqrt{\mathbf{D}}$ stands for the piece-wise square root of the matrix components.

FastICA algorithm

The FastICA algorithm [28] iteratively seeks for the weight vector \mathbf{w} such that the projection of the input vector $\mathbf{w}^T\mathbf{x}$ maximizes non-gaussianity. Non-gaussianity is measured by the approximation of negentropy. The differential entropy for a random vector \mathbf{x} is defined as [28]:

$$H(\mathbf{x}) = - \int p(\mathbf{x}) \log p(\mathbf{x}) d\mathbf{x} \quad (4.28)$$

The highest entropy among all random variables of equal variance has gaussian variable. This means that any other random variable has a lower entropy. The measurement of the non-gaussianity can be therefore expressed as:

$$J(\mathbf{x}) = H(\mathbf{y}_{gauss}) - H(\mathbf{y}) \quad (4.29)$$

where \mathbf{y}_{gauss} is a gaussian random variable with the same covariance matrix as \mathbf{y} . The Equation 4.29 is called negentropy [28]. In optimization tasks, it is very useful to have approximation of negentropy that can be easily and quickly calculated. One such an approximation is:

$$J(x) \propto (E\{G(x)\} - E\{G(\nu)\})^2 \quad (4.30)$$

where ν is random gaussian variable with zero mean and unit variance and G is an non-quadratic function. Hyvärinen and Oja [28, 27] recommend:

$$\begin{aligned} G_1(u) &= \frac{1}{a_1} \log \cosh a_1 u \\ G_2(u) &= -\exp(-u^2/2) \end{aligned} \quad (4.31)$$

The FastICA algorithm for the computation of one independent component is as follows:

1. Choose a random initial weight/projection vector \mathbf{w}
2. $\mathbf{w}^+ \leftarrow E\{\mathbf{x}g(\mathbf{w}^T\mathbf{x})\} - E\{g'(\mathbf{w}^T\mathbf{x})\}\mathbf{w}$
3. $\mathbf{w} \leftarrow \mathbf{w}^+ / \|\mathbf{w}^+\|$
4. If not converged, return to 2.



Figure 4.9: The example of facial image independent components whose linear combination yields a face space [87].

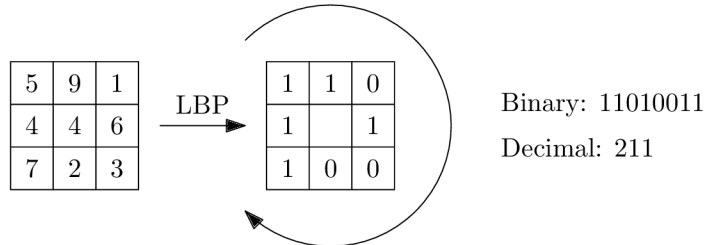


Figure 4.10: LBP operator. The left part of the figure shows a pixel with intensity value 4 and its 8 neighbors. Each neighbor value is compared with the center value. If it is greater or equal the 1 is added to the LBP, 0 otherwise. The resulting binary pattern – 11010011 is thus extracted.

The g function in algorithm above is a derivative of function from Equation 4.31.

In order to find n weight vectors \mathbf{w}_i we have to run the algorithm above n times. To prevent different initial weight vectors converging to the same maxima, weight vectors have to be decorrelated. If we have computed p independent components, the following procedure is repeated after every iteration of the FastICA algorithm when we are computing independent component $p + 1$ [28]:

1. $\mathbf{w}_{p+1} \leftarrow \mathbf{w}_{p+1} - \sum_{j=1}^p \mathbf{w}_{p+1}^T \mathbf{w}_j^2$
2. $\mathbf{w}_{p+1} \leftarrow \mathbf{w}_{p+1} / \sqrt{\mathbf{w}_{p+1}^T \mathbf{w}_{p+1}}$

Usage of ICA in Face Recognition

The independent component analysis has been widely used in the area of 2D face recognition. Many researchers have shown that it achieves significantly better results than PCA on the same dataset [87]. The example of the independent components whose linear combination forms a face space is shown in Figure 4.9.

4.5 Local Binary Patterns

The Local Binary Pattern operator (LBP) is a per-pixel operator used for texture description [60], face recognition [3], face detection, and facial expression recognition [77]. It takes the intensity value of a particular pixel and compares it to its 8 neighbors - see Figure 4.10 for more details. The main advantage of the LBP operator is that it is invariant to global brightness in the image since it is based on the relative comparisons. The generalization of LBP approach to Local Ternary Patterns (LTP) was introduced in [80]. LBP were also employed in the 3D face recognition [81].

The facial image is usually divided into small cells. The histogram of LBP values is calculated within each cell and all histograms are concatenated into one feature vector.

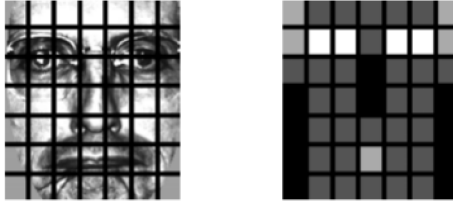


Figure 4.11: Facial image divided into 7×7 grid (left) and the weights set for weighted χ^2 dissimilarity measure (right). Black squares indicate weight 0.0, dark grey 1.0, light grey 2.0 and white 4.0 [3].

Dividing the image into the grid helps to handle even local changes of the brightness.

Once the histograms of the LBP values are calculated, and the feature vector is thus extracted, there are several options how to compare them. Ahonen [3] suggests histogram intersection metric:

$$d_{\text{intersection}}(\mathbf{P}, \mathbf{R}) = \sum_i \min(P_i, R_i) \quad (4.32)$$

Log-likelihood statistics:

$$d_{\log}(\mathbf{P}, \mathbf{R}) = - \sum_i P_i \log R_i \quad (4.33)$$

or Chi-square statistic:

$$\chi^2(\mathbf{P}, \mathbf{R}) = \sum_i \frac{(P_i - R_i)^2}{P_i + R_i} \quad (4.34)$$

where \mathbf{P} is the input probe histogram and \mathbf{R} is the reference. Ahonen further suggests weighting of individual grid cells from which the histograms are created. It can be expected that some of the regions contain more useful information than others in terms of distinguishing among individuals. For example, eyes seem to be an important cue in human face recognition. Weighted Chi-square statistic therefore becomes:

$$\chi_w^2(\mathbf{P}, \mathbf{R}) = \sum_{i,j} w_j \frac{(P_{i,j} - R_{i,j})^2}{P_{i,j} + R_{i,j}} \quad (4.35)$$

The weights computed by Ahonen are illustrated in Figure 4.11. To find the weights w_j for the weighted χ_2 , the following procedure was adopted: a training set was classified using only one of the cell grid at a time. The recognition rates of corresponding windows on the left and right half of the face were averaged. Then the windows whose rate lay below the 0.2 percentile of the rates got weight 0 and windows whose rate lay above the 0.8 and 0.9 percentile got weights 2.0 and 4.0, respectively. The other windows got weight 1.0.

4.6 Active Models

Active models are a branch of computer vision algorithms that try to match previously learned statistical model to a new image. There are two main concepts involved in this process. The first one is the principal component analysis (see Section 4.4.1), the latter involved concept is an iterative process of finding the optimal model deformation (instance)

that best matches the new image. Active models were introduced by Cootes, Edwards, and Taylor [16].

There are two main approaches to the active models – Active Shape Models (ASM) and Active Appearance Models (AAM).

4.6.1 Active Shape Models

The active shape model is a statistical model of contours within the image. During the iteration, when the model is deformed in order to match the input image, a relatively small neighbourhood around each control point of the contour is investigated. Subsequently, each control point is shifted to the position which matches best the training data. In order to do that, we must represent a shape and learn how to warp and model the shape variation.

Procrustes Analysis

The common representation of the shape formed from d points in n -dimensional space \mathbb{R}^n is column vector \mathbf{x} . For instance, a shape that consists from 4 points in 2D space is represented as:

$$\mathbf{x}^T = (x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4) \quad (4.36)$$

In order to compensate the influence of the global shape transformation that is not directly connected with the shape (e.g. scale and rotation), the shapes from the training set have to be aligned in a common co-ordinate frame. One such an approach is Procrustes Analysis [35]. It aligns each shape such that the sum of distances of each shape to the mean shape is minimized. The objective function d where we try to minimize the distance between the shape \mathbf{x}_i and mean shape $\bar{\mathbf{x}}$ is defined as:

$$d = \sum_{i=1}^m (\bar{\mathbf{x}} - \mathbf{x}_i)^T (\bar{\mathbf{x}} - \mathbf{x}_i) \quad (4.37)$$

Iterative Procrustes Analysis algorithm is as follows [16]:

Repeat until there is no significant change of $\bar{\mathbf{x}}$ after the iteration

1. Translate each shape so that the sum of its components in every dimension is zero.
2. Choose the first shape as the initial mean $\bar{\mathbf{x}}$ and scale it, such that $|\bar{\mathbf{x}}| = 1$.
3. Align all shapes with current $\bar{\mathbf{x}}$.
4. Recalculate the mean from the aligned shapes.
5. Scale the mean, so that $|\bar{\mathbf{x}}| = 1$.
6. If not converged, return to 3.

The key part of the algorithm above is the alignment of shape \mathbf{x}_i to the current mean $\bar{\mathbf{x}}$. This involves scaling and rotation. Since our mean estimation is unit vector, for scaling it is sufficient if the \mathbf{x}_i is also normalized.

Removing the rotational component is more complex. Suppose that we have two shapes $\mathbf{a} = (x_1, x_2, \dots, x_d, y_1, y_2, \dots, y_d)$ and $\mathbf{b} = (w_1, w_2, \dots, w_d, z_1, z_2, \dots, z_d)$ in two dimensional space and we are trying to rotate \mathbf{b} by an angle θ in order to minimize the sum of squares error between \mathbf{a} and rotation $T_\theta(\mathbf{b}) = (u_1, u_2, \dots, u_d, v_1, v_2, \dots, v_d)$, where $u_i = \cos \theta w_i - \sin \theta z_i$ and $v_i = \sin \theta w_i + \cos \theta z_i$.

The error function is then:

$$E(\theta) = (u_1 - x_1)^2 + (u_1 - x_1)^2 + \dots + (u_d - x_d)^2 + (v_1 - y_1)^2 + (v_2 - y_2)^2 + \dots + (v_d - y_d)^2 \quad (4.38)$$

Taking the derivative of function $E(\theta)$ with respect to θ and solving for $\theta = 0$ gives:

$$\theta = \tan^{-1} \left(\frac{\sum_{i=1}^d (w_i y_i - z_i x_i)}{\sum_{i=1}^d (w_i x_i - z_i y_i)} \right) \quad (4.39)$$

PCA and Modelling Shape Variation

The key concept of modelling shape variation is PCA that is performed on the aligned shapes $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_s$. The PCA produces the projection matrix Φ . The number of chosen eigenvectors (columns of the matrix Φ) usually depends on the problem being solved. It has been suggested that the number of eigenvectors should catch 98% of the variance in the training set [16]. The optimal number of eigenvectors has also been examined in [66].

Generating plausible shapes

Every new shape \mathbf{x} can be approximated as

$$\mathbf{x} \approx \bar{\mathbf{x}} + \Phi \mathbf{b} \quad (4.40)$$

where $\bar{\mathbf{x}}$ is the mean of the training data, $\Phi = (\phi_1, \dots, \phi_t)$ are the eigenvectors, and \mathbf{b} is t -dimensional column vector of the shape model parameters:

$$\mathbf{b} = \Phi^T (\mathbf{x} - \bar{\mathbf{x}}) \quad (4.41)$$

By applying limits of $\pm 3\sqrt{\lambda_i}$ to each component b_i of \mathbf{b} we ensure that the generated shape is similar to those in the original training set. λ_i is the i^{th} eigenvalue corresponding to the i^{th} eigenvector established by PCA.

Fitting a Model to New Points

The model is fully described by the shape parameters \mathbf{b} and transformation $T_{X_t, Y_t, s, \theta}$ that transforms the shape from the model co-ordinate frame to the image co-ordinate frame. These transformations involve a translation by vector (X_t, Y_t) scaling by s and rotation by θ . The position of the shape points \mathbf{Y} in the image is given by:

$$\mathbf{Y} = T_{X_t, Y_t, s, \theta}(\mathbf{x}) = T_{X_t, Y_t, s, \theta}(\bar{\mathbf{x}} + \Phi \mathbf{b}) \quad (4.42)$$

where

$$T_{X_t, Y_t, s, \theta} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} X_t \\ Y_t \end{pmatrix} + \begin{pmatrix} s \cos \theta & s \sin \theta \\ -s \sin \theta & s \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad (4.43)$$

In order to find the shape and transformation parameters that best fit a model \mathbf{x} to the new image points \mathbf{Y} , we have to specify an error between the desired shape and the current shape estimation given by parameters \mathbf{b} and $T_{X_t, Y_t, s, \theta}$:

$$E(\mathbf{b}, T_{X_t, Y_t, s, \theta}) = |Y - T_{X_t, Y_t, s, \theta}(\bar{\mathbf{x}} + \Phi \mathbf{b})|^2 \quad (4.44)$$



Figure 4.12: The example of matching the ASM model to a facial image. From left to right: the initial rough estimation, after 2 iterations, after 18 iterations [16].

The algorithm that is repeated until no significant changes in pose and shape parameters are made is below:

1. Initiate shape parameters \mathbf{b} to zero, such that $b_i = 0$ for $i = 0, \dots, t$
2. Generate model instance $\mathbf{x} = \bar{\mathbf{x}} + \Phi \mathbf{b}$
3. Find transformation parameters that best map shape \mathbf{x} to shape \mathbf{Y}
4. Invert the transformation parameters and project \mathbf{Y} to the model co-ordinate frame:
 $\mathbf{y} = T_{X_t, Y_t, s, \theta}^{-1}(\mathbf{Y}) = T_{-X_t, -Y_t, \frac{1}{s}, -\theta}(\mathbf{Y})$
5. Project \mathbf{y} into the tangent plane to $\bar{\mathbf{x}}$ by scaling by $1/(\mathbf{y}^T \bar{\mathbf{x}})$
6. Update the shape parameters $\mathbf{b} = \Phi^T(\mathbf{y} - \bar{\mathbf{x}})$
7. Apply constraints on \mathbf{b} in order to generate a plausible shape.

Understanding the image structure

The main task involved in the ASM is to find from the initial rough approximation the actual coordinates of the shape points within the given image. The example of this process is illustrated in Figure 4.12.

The following steps are made in every iteration until convergence:

1. Inspect a normal to shape boundary along each point $X_i = (x_i, y_i)$ of the shape and replace it with X_i' from the boundary that matches best the learned data.
2. Update parameters $T_{X_t, Y_t, s, \theta}$ and \mathbf{b} that best fit to the new shape X' .

A Mahalanobis distance is employed in order to estimate a match between the candidate location of the shape point X_i and the trained data. During the model training, m pixel intensities on either side are sampled at every shape point X_i . If we have s training images, this process provides s samples $\mathbf{g}_{i_1}, \dots, \mathbf{g}_{i_s}$. After that, their mean $\bar{\mathbf{g}}_i$ and covariance matrix \mathbf{S}_i is calculated. The fit quality of a new candidate sample \mathbf{g}_i' is defined as:

$$fit(\mathbf{g}_i') = (\mathbf{g}_i' - \bar{\mathbf{g}}_i)^T \mathbf{S}_i^{-1} (\mathbf{g}_i' - \bar{\mathbf{g}}_i) \quad (4.45)$$

When we are inspecting and sampling a normal to shape boundary along each point X_i , we sample k different sample candidates $\mathbf{g}_{i_1}, \dots, \mathbf{g}_{i_k}$. The sample that fits the best according to the Equation 4.45 to the learned data is chosen as a new point estimation X_i . See Figure 4.13.

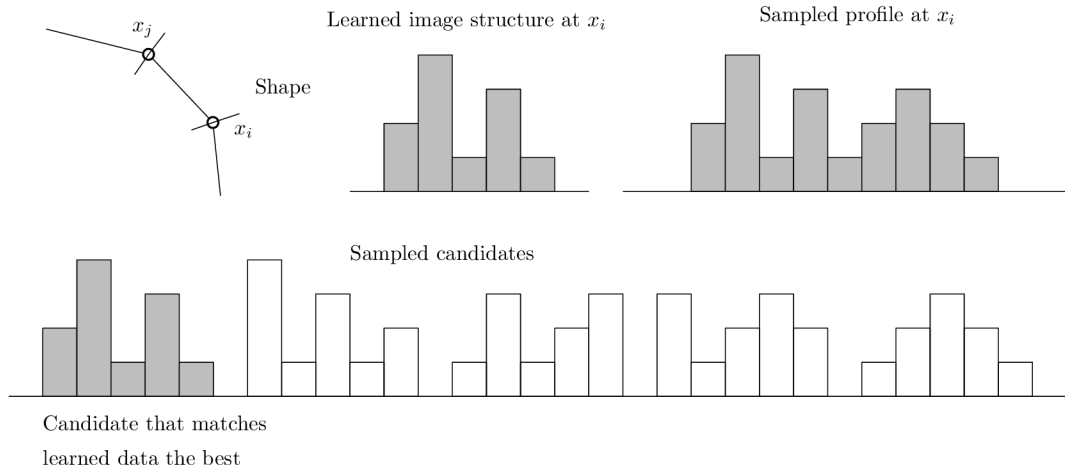


Figure 4.13: The example of sampling the pixel intensities along shape normal. The candidate that matches the learned data the best is chosen as a new point of the shape.

4.6.2 Active Appearance Models

While the ASM only use limited neighbourhood around each vertex of the mesh, AAM benefits from the entire texture within the mesh [16]. The terminology and notation presented in this chapter is from the paper by Iain Matthews and Simon Baker [46].

The shape portion of the AAM is similar to ASM. The positions of mesh vertices are controlled by the linear model. This means that the shape can be expressed as a linear combination of principal components. The shape \mathbf{s} is a vector $\mathbf{s} = (x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_n)^T$:

$$\mathbf{s} = \bar{\mathbf{s}} + \sum_{i=1}^{2n} p_i \mathbf{s}_i \quad (4.46)$$

where $\bar{\mathbf{s}}$ is the mean shape, \mathbf{s}_i is an i^{th} principal component of the shape linear model, p_i is its corresponding parameter, and n is the number of used principal components (eigenvectors).

The appearance portion of the ASM is also defined by the linear model of the texture defined within the base mesh \mathbf{s}_0 . Let \mathbf{s}_0 also denotes the set of pixels $\mathbf{x} = (x, y)^T$ that are contained within the mean mesh \mathbf{s}_0 . The appearance is then an image $A(\mathbf{x})$ defined over the pixels $\mathbf{x} \in \mathbf{s}_0$:

$$A(\mathbf{x}) = A_0 + \sum_{i=1}^m \lambda_i A_i(\mathbf{x}) \quad \forall \mathbf{x} \in \mathbf{s}_0 \quad (4.47)$$

where λ_i are m coefficients associated with the m principal components of the appearance portion of the AAM model.

The instance of the AAM is then defined by the set of parameters $\mathbf{p} = (p_1, p_2, \dots, p_n)^T$ and $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_m)^T$. The appearance portion of the model is defined within the frame of base mesh \mathbf{s}_0 . The resulting shape \mathbf{s} together with the mean shape \mathbf{s}_0 define a piecewise affine warp from \mathbf{s}_0 to \mathbf{s} which is denoted $\mathbf{W}(\mathbf{x}, \mathbf{p})$. The final AAM instance, denoted $M(\mathbf{W}(\mathbf{x}, \mathbf{p}))$, is computed by warping the appearance A from \mathbf{s}_0 to \mathbf{s} using $\mathbf{W}(\mathbf{x}, \mathbf{p})$. For the illustration see Figure 4.14.

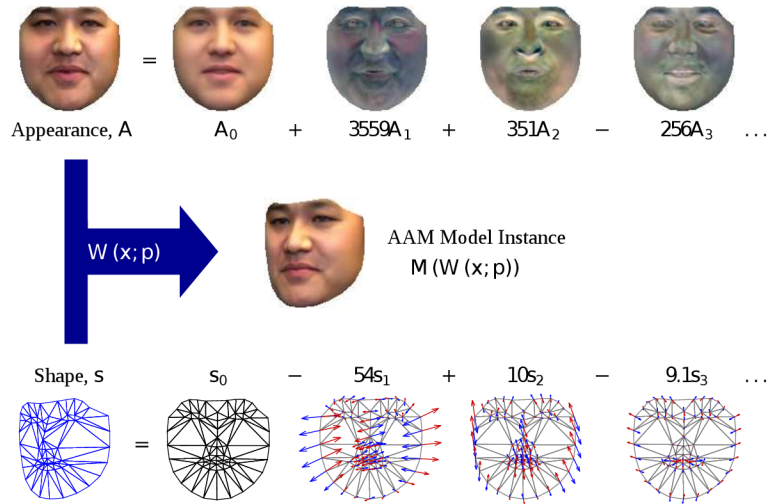


Figure 4.14: The instantiation of the AAM model. The mesh s together with mean mesh s_0 define a warp $\mathbf{W}(\mathbf{x}, \mathbf{p})$ [46].

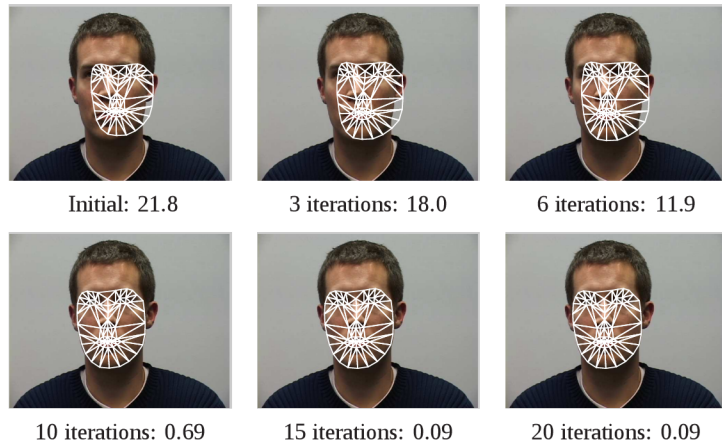


Figure 4.15: The example of matching an AAM model to a given image [46]. The number after the iteration counter represents the current matching error (see Eq. 4.48).

There are two types of active appearance models, those with an independent shape and appearance (referred to as independent AAM) and those with a single set of linear parameters that control both the shape and appearance. In general, latter type of AAM can be produced by applying the third PCA on the concatenated \mathbf{p} and $\boldsymbol{\lambda}$ parameters.

AAM Fitting algorithms

The essential part of the AAM is the ability to fit the shape and appearance of the model to the new image. The example of the iterative AAM fitting process is in Figure 4.15.

Suppose that we have a new input image $I(\mathbf{x})$ and we are trying to fit the AAM. This means we are trying to find optimal shape and appearance parameters \mathbf{p} and $\boldsymbol{\lambda}$. The model instance $M(\mathbf{W}(\mathbf{x}, \mathbf{p})) = A(\mathbf{x})$ and the input image $I(\mathbf{x})$ have to be the same. This leads to the following error function:

$$E(\mathbf{x}) = (A_0(\mathbf{x}) + \sum_{i=1}^m \lambda_i A_i(\mathbf{x})) - I(\mathbf{W}^{-1}(\mathbf{x}; \mathbf{p})) \quad (4.48)$$

The first part of the error formula is the appearance of the model defined within the frame of the base mesh \mathbf{s}_0 . The second part is the input image backwards warped onto the base mesh. These two parts are subtracted and form the overall fit error function. The objective function of the model parameters \mathbf{p} and $\boldsymbol{\lambda}$ that we are trying to minimize is defined for each pixel \mathbf{x} within the base mesh \mathbf{s}_0 :

$$\sum_{\mathbf{x} \in \mathbf{s}_0} [(A_0(\mathbf{x}) + \sum_{i=1}^m \lambda_i A_i(\mathbf{x})) - I(\mathbf{W}^{-1}(\mathbf{x}; \mathbf{p}))]^2 \quad (4.49)$$

4.6.3 Active Models and Face Recognition

The model parameters \mathbf{p} and $\boldsymbol{\lambda}$ may be directly used as the input to a classifier. However, the parameters capture the class specific information as well as the information about facial expression or head rotation. One approach to deal with this problem is the usage of the class-specific Mahalanobis distances. The Mahalanobis distance d_i of the model parameters $\mathbf{c} = \begin{pmatrix} \mathbf{p} \\ \boldsymbol{\lambda} \end{pmatrix}$ from class i is given by:

$$d_i = (\mathbf{c} - \bar{\mathbf{c}}_i)^T \mathbf{C}_i^{-1} (\mathbf{c} - \bar{\mathbf{c}}_i) \quad (4.50)$$

where $\bar{\mathbf{c}}_i$ is the mean of class i and \mathbf{C}_i is its covariance matrix. However, this is very restrictive because this approach needs a sufficient number of examples for each subject enrolled in the biometric system. Another approach is to assume a common within-class covariance matrix \mathbf{C} . With this assumption, a classification using LDA may be used (see Section 4.4.2).

4.7 3D Face Recognition

In this section, various approaches of the three-dimensional face recognition are described. The three-dimensional adaptation of eigenface and fisherface methods will be 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, are described.

The three-dimensional face recognition brings several advantages compared to the two-dimensional recognition. Mainly because it provides new facilities of discrimination as depth data are added.

Another advantage of the three-dimensional face recognition is related to the data capture technique. Many three-dimensional cameras operate in the infrared part of the spectrum [48] which is independent to lighting conditions, such that the direction of light and shadows on the face does not negatively affect the face recognition.

4.7.1 Representations of the 3D Face Model

A three-dimensional face model could be represented in various forms. Most often point-clouds [66], meshes [43] and range images [25] are used. Although all of these representations

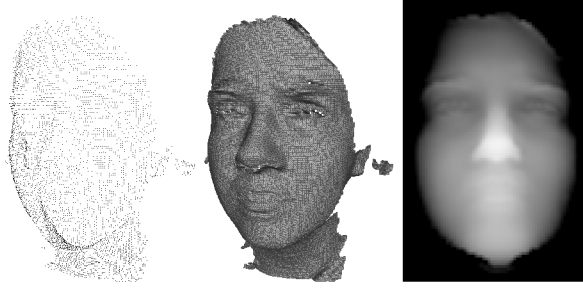


Figure 4.16: Pointcloud, mesh, and range image representations of the same face.

could be mutually converted to other representations, face recognition technique depends on the form of a three-dimensional model.

The common used representation of the 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 the camera, therefore this array could be stored as greyscale bitmap image. Range images are not universal for all types of three-dimensional models. Mainly because they cannot store information about points that are hidden by other points. Using the range image representation on three-dimensional faces is not affected by this limitation because the frontal view of a face does not contain many points that are hidden by other parts of the face.

All three representations of the same face (from GavabDB face database [49]) are in Figure 4.16.

4.7.2 Curvature Analysis

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

Assume a parametric curve $\gamma(S)$, where S is a parameter which determines tangent vector $T(S)$ and a 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 the surface they measure, how the surface bends by different amounts in different directions at that point. An illustration is in Figure 4.17a. 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 surface, several options are available.

Gaussian curvature K [76] of a point on a surface is defined as the product of principal curvatures k_1 and k_2 .

$$K = k_1 k_2 \tag{4.51}$$

Mean curvature H [76] is the average of principal curvatures k_1 and k_2 .

$$H = \frac{1}{2}(k_1 + k_2) \tag{4.52}$$

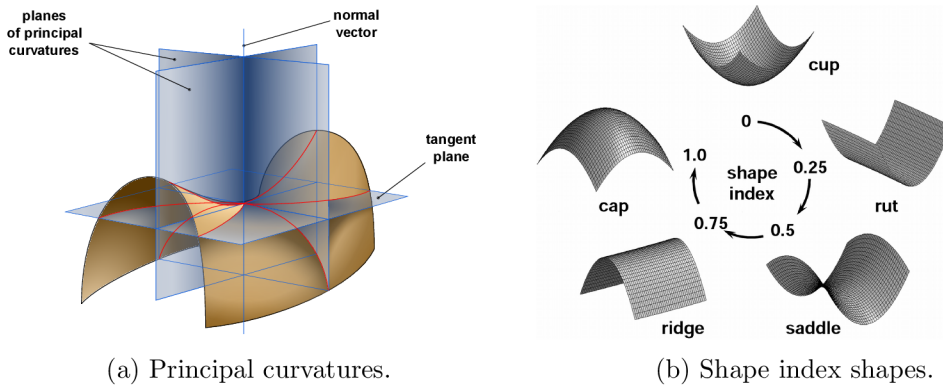


Figure 4.17: Principal curvatures (a) and shape index shapes (b) [45].

Table 4.1: Shape classification based on the sign of mean and Gaussian curvature.

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

Shape index S [43] is for the classification of the surface into categories. See Figure 4.17b.

$$S = \frac{1}{2} - \frac{1}{\pi} \tan^{-1} \frac{k_1 + k_2}{k_1 - k_2} \quad (4.53)$$

where the principal component k_1 is greater than k_2 .

In Figure 4.18, the original range image, Gaussian curvature, mean curvature, shape index, and marked pits and peaks 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 [76], see Table 4.1. The pit and peak points of the face are also shown in Figure 4.18.

4.7.3 Facial Landmarks Detection

Detecting the facial landmarks from the three-dimensional data cannot be performed using the same algorithms as on the 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.

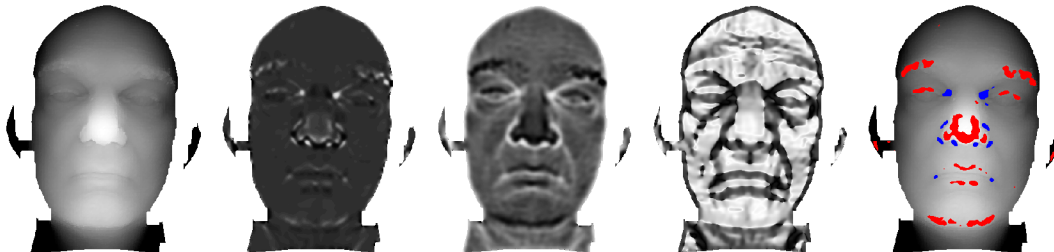


Figure 4.18: From left to right: the original range image, corresponding Gaussian curvature, mean curvature, shape index, and pit (blue) and peak points (red) on the face surface.

Locating the Nose Tip

Location of the nose tip is in many three-dimensional facial recognition methods the fundamental part of preprocessing [25, 43, 45, 61]. Various techniques of localization of this point are used.

Heseltine [25], during the face preprocessing in his recognition approach, claimed that the nose is the most protruding point on the surface. 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. [76] proposed the algorithm for the nose tip localization that consists of two stages. First, the y -coordinate is found and then an 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 is created. Maximum of this curve along y -axis is the y -coordinate of the nose. See Figure 4.19.

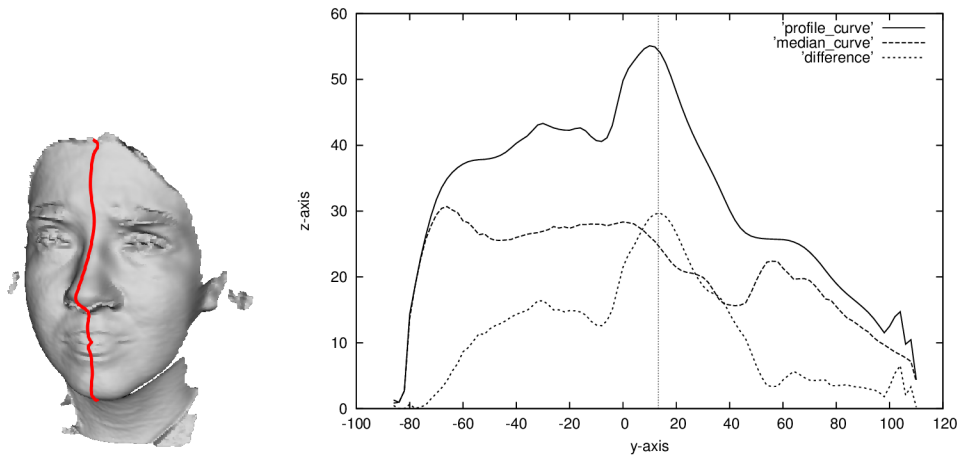


Figure 4.19: Localizing the y -coordinate of the nose.

The x -coordinate of the nose is found by projecting the x -projection of the curvature image. This is done by calculating the percentage of peak curvature points of neighbour 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 4.20.

Nose Corners

A nose corners localization method is similar to the the nose tip localization in principle. Segundo et al. [76] recommend 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 gradient information of this curve and look for one peak on each side of the nose tip.

A more accurate nose corner localization could be achieved by the use of x -projection data obtained from the curvature analysis. Either searching for minimum values in the mean

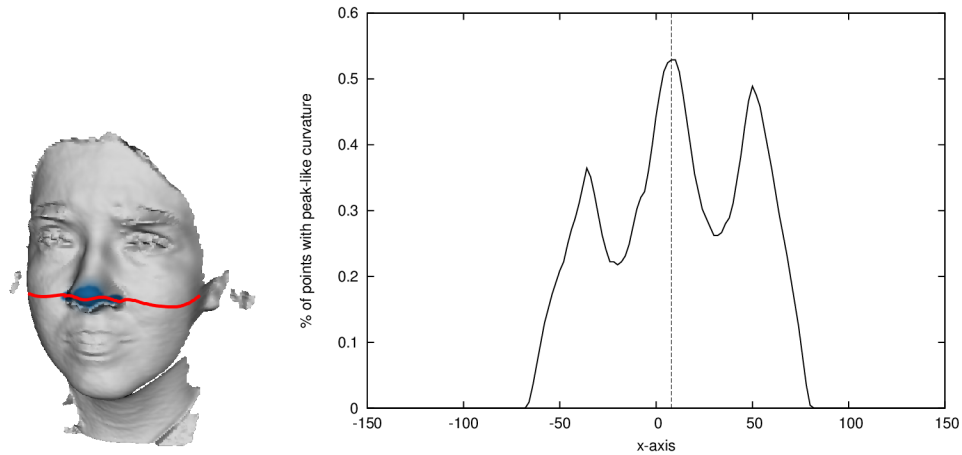


Figure 4.20: Localizing x -coordinate of the nose.

curvature projection or searching for the minimum values in the shape index projection is possible. See Figure 4.21.

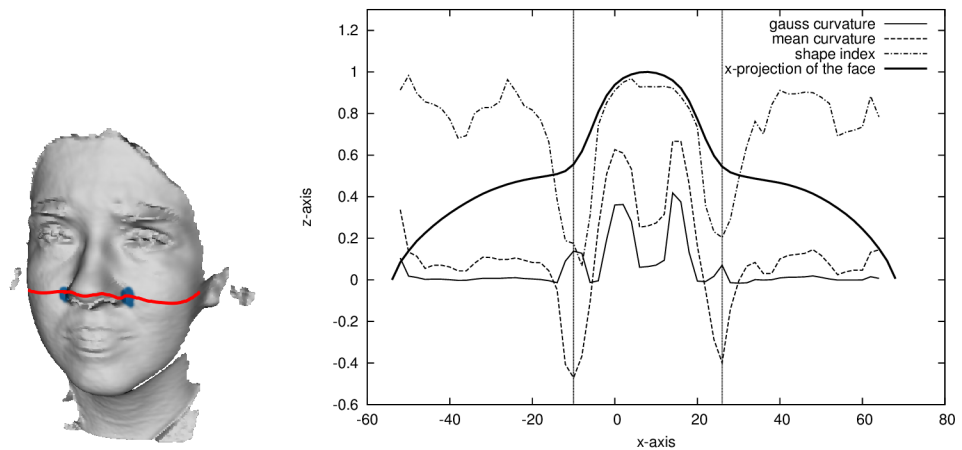


Figure 4.21: Localizing nose corners.

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 4.22). As the nose coordinates are known, each peak can be assigned to its respective facial feature. The x -coordinates of the eye corners are computed from another x -projection curve, where is the ratio of the pit curvature points for each x -coordinate calculated. The eyes match to the two arches 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.

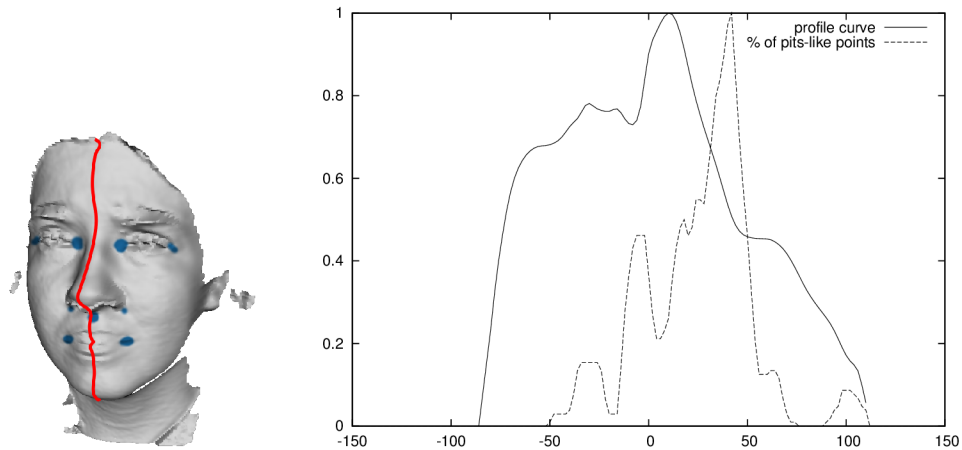


Figure 4.22: Localizing y -coordinate of the eye corners.

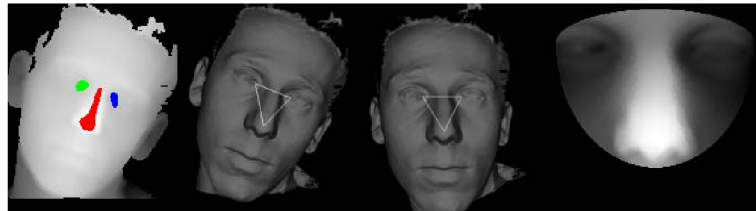


Figure 4.23: Localization of eye corners and nose, projection to generic position and conversion to the range image [14].

4.7.4 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 [25] in his recognition algorithm normalizes the face orientation during the landmark detection. First, the nose tip is detected and the face is translated so that the nose tip is located at the coordinate origin. Then, the roll correction is performed by locating the nose bridge and rotating the whole face about the z -axis to make the 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 rotation about the y -axis to correct the head pan. During this step, the symmetry of the face is used.

Colombo et al. [14] locate 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 a mask. The result of this process is the input for eigenface-based face recognition. The illustration of this process is in Figure 4.23.

4.7.5 Eigenfaces in Three-Dimensional Face Recognition

The eigenfaces method applied on three-dimensional face recognition is very similar to its two-dimensional variant described in Section 4.4.1. The mean face and afterwards the

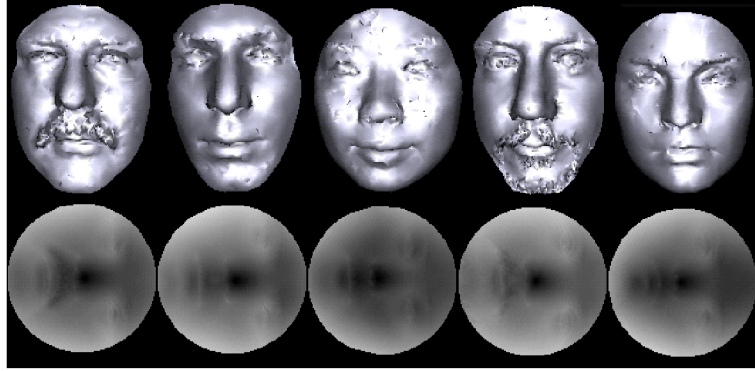


Figure 4.24: Examples of mapped range images [61].

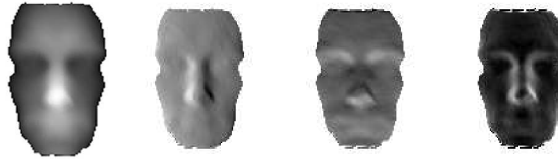


Figure 4.25: Original range image, Sobel x, Sobel y, and Sobel magnitude [25].

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.

Range Image Processing

The face recognition method proposed by Pan et al. [61] 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 measures the distortion between the original surface and the plane is used. The transformation to the planar circle is performed so that E is minimal. Some examples of mapped range images are in Figure 4.24.

Heseltine [25] 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 which increases the recognition performance. In Figure 4.25, the original range image and some applied filters are shown.

4.7.6 Model Based 3D Face Recognition

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

Lu et. al [43] proposed a method that compares a face scan to a 3D model stored in a 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 [85] is performed in the second step.

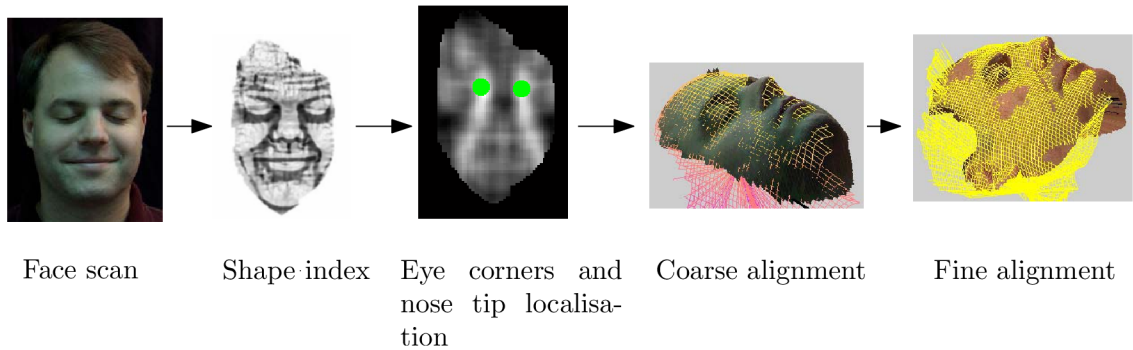


Figure 4.26: Process of the model based recognition [43].

The fine registration process, the final step, uses the Iterative Closest Point algorithm [8]. 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 4.26.

4.7.7 Face Recognition Using Histogram Based Features

The face recognition algorithm introduced by Zhou et al. [89] 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 that 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:

$$bin_{n,i} = [Z_{n,k-1}, Z_{n,k}] \quad (4.54)$$

$$Z_{n,0} = Z_{n,min}, Z_{n,1}, \dots, Z_{n,K} = Z_{n,max} \quad (4.55)$$

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

$$v_{k,n} = \frac{|\{p_i(x_i, y_i, z_i) \mid p_i \in S_n, Z_{k-1} < z_i < Z_k\}|}{|S_n|} \quad (4.56)$$

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

The input range image and the corresponding feature vector is shown in Figure 4.27. The comparison between two faces is performed by a distance calculation between the two corresponding feature vectors.

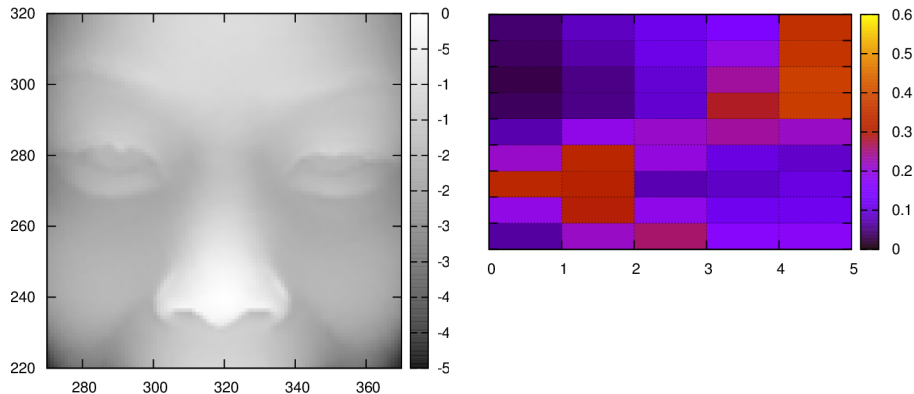


Figure 4.27: Original range image and corresponding feature vector calculated as a histogram of z -coordinates in N stripes.

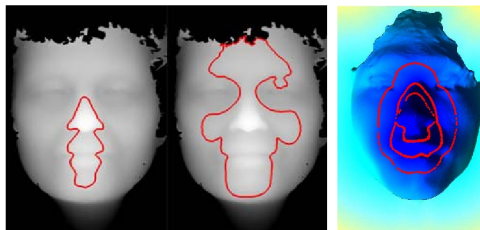


Figure 4.28: Iso-depth curves (left, middle) and iso-geodesic curves (right).

4.7.8 Recognition Based on Facial Curves

In recent years, a family of the 3D face recognition methods, which is based on the comparison of facial curves, has emerged [30, 31, 7]. In these methods, the nose tip is located first. After that, a set of closed curves around the nose is created, and the features are extracted.

In [31], recognition based on iso-depth and iso-geodesic curves is proposed. The iso-depth curve is extracted from the intersection between the face surface and the parallel plane, perpendicular to the z -axis. The iso-geodesic curve is a set of all points on the surface that have the same geodesic distance from a given point (see Figure 4.28). The geodesic distance between two points on the surface is a generalization of the term distance on a curved surface.

Contrary to the iso-depth curves, from a given point, iso-geodesic curves are invariant to translation and rotation. This means that no pose normalization of the face is needed in order to deploy a face recognition algorithm strictly based on iso-geodesic curves. However, the precise localization of the nose-tip is still a crucial part of the recognition pipeline.

There are several shape descriptors used for feature extraction in [31]. A set of 5 simple shape descriptors (convexity, ratio of principal axes, compactness, circular variance, and elliptical variance) is provided. Moreover, the Euclidian distance between the curve center and points on the curve is sampled for 120 points on the surface and projected using LDA in order to reduce dimensionality of the feature vector. Three curves are extracted for each face.

The 3D face recognition algorithm proposed in [7] uses iso-geodesic stripes and the surface data are encoded in the form of a graph. The nodes of the graph are the extracted stripes and the directed edges are labeled with *3D Weighted Walkthroughs*. The walk-

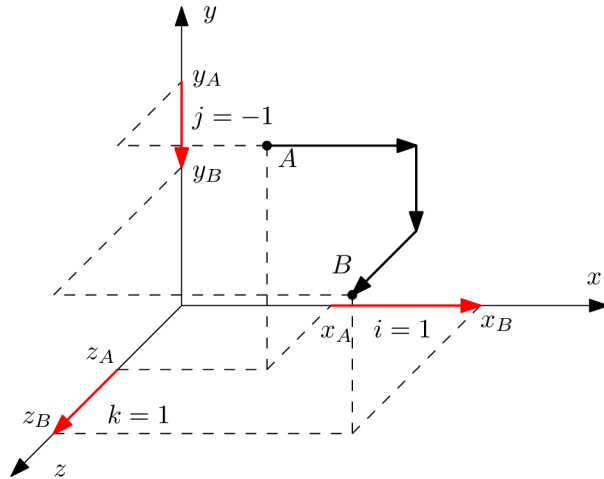


Figure 4.29: Example of 3D walkthrough between points A and B : $\langle i, j, k \rangle = \langle 1, -1, 1 \rangle$.

through from point $A = (x_A, y_A, z_A)$ to $B = (x_B, y_B, z_B)$ is illustrated in Figure 4.29. It is a pair $\langle i, j, k \rangle$ that describes the sign of mutual positions projected on all three axes. For example, if $x_A < x_B \wedge y_A > y_B \wedge z_A < z_B$ holds, then $\langle i, j, k \rangle = \langle 1, -1, 1 \rangle$. For more information about the generalization of walkthroughs from points to a set of points, see [7].

Chapter 5

Proposal of the Recognition Algorithm

This chapter describes the generalized recognition pipeline that takes the input face mesh and normalizes it, such that the rotation is compensated for. After that, some image representations of the surface, texture, and curvature are generated from the normalized mesh. The pipeline continues with the application of specific image filters. Finally, subspace projections are used in order to extract features.

The main idea of the proposed method is the score-level fusion of involved individual recognition units. By the application of some filter bank, e.g. Gabor filter bank, on the input image we obtain m new images. However, the number of filters within the bank is quite high, therefore some optimization selection method is needed in order to improve speed as well as remove redundancy. We employ hill-climbing selection and the optimization criterion is fusion EER.

The similar approach has been proposed by Yang et al. [86]. Yang et al. use AdaBoost to select a small set of Gabor features (weak classifiers) in order to form a strong classifier. Moreover, he proposed intra-face and extra-face difference space to transform a multi-class classification to a binary decision. The task is to assign the input two images to intra-personal or extra-personal space.

Su et al. [78] came with an algorithm exploiting both local and global features. The global features are extracted from the whole face images by keeping the low-frequency coefficients of Fourier transform, which we believe encodes the holistic facial information, such as the facial contour. For local feature extraction, Gabor wavelets are applied on the face image patches. The resulting classifier is based on the hierarchical feature-level fusion utilizing *Linear Discriminant Analysis*. However, both methods from Yang as well as from Su are designated for 2D face recognition. Our proposed method is able to profit from both 2D texture and 3D shape data.

5.1 Generalized Recognition Pipeline for Face Recognition

Biometric recognition pipeline usually consists of data acquisition, preprocessing, feature extraction, and comparison that yields to the final decision whether the user is accepted or not [84].

The recognition pipeline suitable for the face recognition presented in this thesis is depicted in Figure 5.1. Individual components are described in the following sections. First,

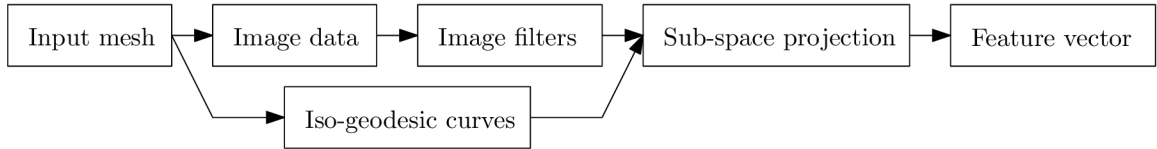


Figure 5.1: Recognition pipeline for 3D face recognition.

the input mesh is aligned. One can use an automatic landmark detection (see Section 4.7.3) followed by translation and rotation of the detected points to some predefined position. Another approach might be the ICP align of the entire input face mesh to a reference template (see Section 4.7.6).

Image data are extracted from the aligned face mesh. The mesh is transformed to the range image and texture representations. The range image is further processed in order to gain 4 new curvature representations – mean curvature, Gaussian curvature, eigencurvature and shape index image.

The alternative to image filters approach are the iso-geodesic curves. A set of curves centering at a given point is retrieved from the mesh and converted to a set of points.

The next step, common to both iso-geodesic curves as well as image filters, is some subspace projection of the input data. Image matrix is transformed to the column vector representation and projected to the low-dimensional space after that. A set of 3D points is transformed to the simple column vector in the same manner. PCA (Section 4.4.1), ICA (Section 4.4.3), as well as LDA (Section 4.4.2) are suitable for a subsequent subspace projection of the input column vector.

5.2 Face Alignment

The proper face alignment is a crucial part of the input face pre-processing. Here, we use an alignment based on the reference face template. Input face mesh is aligned to the template such that the sum of square differences between the input face mesh and corresponding points on the template is minimal. In the following subsections, the creation of the reference face template and the alignment itself will be described.

5.2.1 Reference Template Creation

The reference face template was created from 100 face scans taken from the FRGC database. Nine points were manually annotated on each scan – 2 outer eye corners, 2 inner eye corners, nasal bridge, nose tip, outer nose corners, and lower nose corner. *Procrustes analysis* [30, 16] was used in order to align annotated points as well as corresponding scans. The algorithm operates in the following steps:

1. Translate each scan so that its center of gravity (CoG) is at the origin.
2. Arbitrarily choose one example as an initial estimate of the mean.
3. Align all scans (using translation and rotation) with the mean.
4. Re-estimate the mean from aligned scans.
5. If not converged, return to 3.



Figure 5.2: Mean reference face template used for proper registration of input faces.

The key point during the Procrustes analysis is the alignment of all scans to the reference mean face template. The translation is quite simple – scan is moved in such direction that its CoG merges with the CoG of the reference mean scan.

The rotation is performed using *Singular Value Decomposition* (SVD). Let the $X = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_9)$ denotes a set of individual annotated landmarks $\mathbf{x}_i = (x_{xi}, x_{yi}, x_{zi})^T$. The similar 3×9 matrix Y denotes landmarks from the mean face scan. First, the covariance matrix S is calculated:

$$S = XY^T \quad (5.1)$$

The next step is singular value decomposition of S . It seeks for real matrices U and V and for diagonal matrix Σ , such that it holds the equation:

$$S = U\Sigma V^T \quad (5.2)$$

The optimal rotation matrix R from X to Y is finally computed:

$$R = V \begin{pmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ & & & \det(VU^T) \end{pmatrix} U^T. \quad (5.3)$$

Individual aligned face scans were converted into a range image representation (see Section 5.3) and the mean range image was computed. The resulting mean range image was cropped such that it contains only the eyes, nose, upper part of mouth, and cheeks. The range image was sub-sampled to the resolution 17×21 pixels and converted to the mesh representation again. The resulting face alignment reference template is shown in Figure 5.2.

5.2.2 Iterative Closest Point Alignment

The main task of the *Iterative Closest Point* (ICP) algorithm is to align (using translation and rotation) a scanned face to the reference face mesh [8]. Although the scans in our testing databases were taken when the subjects were front-facing the capturing device, variations in rotation and even more in position are present.

The algorithm steps are:

1. Associate all points in the reference template with the corresponding points from the input face using the nearest neighboring criteria.
2. Compute an optimal translation and rotation using least-squares as the optimization criteria.

3. Translate and rotate the input face mesh.
4. Iterate until convergence.

The first problem is that, contrary to the Procrustes analysis align, there is no explicit mapping between the points of the input face and the reference template. Moreover, the scans in FRGC database contain up to 80,000 vertices. The exhaustive linear search for the nearest matching neighbor is thus ineffective. *Fast Linear Approximation of Nearest Neighbor* (FLANN) is therefore used [57].

The ICP algorithm may fail if the initial position of the input face requires significant translation. The corresponding points are then wrongly estimated and the convergence is not assured. In order to avoid this, some rough estimation of the initial translation has to be supplied. We use a simple template matching. The input face mesh is gradually moved over the reference template and a sum of square differences is calculated between template points and appropriate points on the input mesh. The rough initial position estimation is at the point where the sum of square differences is minimal.

The k-means index is used for the nearest neighbor search during the computation of the optimal translation and rotation. The problem is that the input face mesh points index has to be re-calculated after each ICP iteration. However, this issue can be solved with a simple trick: In each ICP iteration the inverse translation and rotation is added to the temporary stack. When there is a need to gain a point p on the reference face mesh that is the nearest to some arbitrary point on the input face, the inverse transformation is applied on the point p first.

5.3 Source Image Data

Although many 3D face recognition algorithms operate directly in three-dimensional space (see [30, 45]) we propose an approach that converts the input 3D face mesh into a 2D matrix on which the subsequent recognition (image filters and feature extraction) operates. Since we have texture as well as the 3D model, several representations describing the texture, depth, and curvature may be deduced.

The range image (depthmap) is created from the input face scan in several steps: First, the point-cloud representation is transformed to the triangular mesh using Delaunay triangulation [41]. After that, the mesh vertices are projected to the x - y plane and the z -coordinate is transformed to a pixel brightness. The brightness of the remaining points within the triangles is linearly interpolated. The Pineda algorithm [67] is used for fast triangle rasterization. The resulting range image is slightly smoothed with Gaussian kernel in order to soften the edges between the triangles.

The remaining images of the surface representation depend on the calculation of the principal curvatures. Curvature k at each point B on the range image is calculated from the z -coordinate b_z of the point B as well as from its surrounding points A and C and their z -coordinates a_z and c_z respectively (see Figure 5.3). The curvature is approximated as the signed angle $\alpha = \pi - |\angle ABC|$. Its sign is deduced from the comparison of b_z and $d_z = \frac{a_z + c_z}{2}$. If the $b_z < d_z$ then the sign is negative. The principal curvatures k_1 and k_2 are estimated in x axis as well as in y axis direction and swapped eventually such that $k_1 > k_2$.

Despite the mean curvature, Gaussian curvature, and shape index (see Section 4.7.2) *eigencurvature* [74] is used. It is computed directly from the image point $\mathbf{P} = (p_x, p_y, p_z)$ and its 8 surroundings $(\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_8)$. It is based on the PCA of the matrix M :

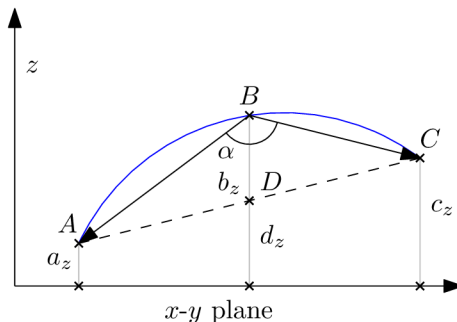


Figure 5.3: Principal curvatures estimation.



Figure 5.4: Image representations of the face surface. From left to right: texture, range image, mean curvature, Gaussian curvature, shape index image, eigencurvature.

$$M = (\mathbf{P} \ \mathbf{P}_1 \ \cdots \ \mathbf{P}_8) = \begin{pmatrix} p_x & p_{1_x} & \cdots & p_{8_x} \\ p_y & p_{1_y} & \cdots & p_{8_y} \\ p_z & p_{1_z} & \cdots & p_{8_z} \end{pmatrix} \quad (5.4)$$

The PCA reveals 3 eigenvectors and their corresponding eigenvalues l_0 , l_1 , and l_2 ($l_0 > l_1 > l_2$). The *eigencurvature* $E_{\mathbf{P}}$ is then:

$$E_{\mathbf{P}} = \frac{l_2}{l_0 + l_1 + l_2} \quad (5.5)$$

The examples of curvature representation images could be found in Figure 5.4.

5.4 Filter Banks

The image filter banks are widely used technique in the area of texture analysis, segmentation, and classification. Individual filters in the bank are used in order to remove unwanted components or features. The two-dimensional filter (kernel k with size $k_w \times k_h$) is convoluted with the input image i and the response d using the following equation is thus calculated:

$$d(x, y) = \sum_{0 \leq x' \leq k_w} \sum_{0 \leq y' \leq k_h} = k(x', y') \cdot i(x + x' - a_x, y + y' - a_y) \quad (5.6)$$

The $\mathbf{a} = (a_x, a_y)$ is the kernel anchor – center of the kernel and is usually set to $\mathbf{a} = (k_w/2, k_h/2)$. In fact, the Equation 5.6 does not compute the real convolution since the kernel is not mirrored around the anchor point. The image filter banks are set of m 2D kernels that are convoluted with the input image (see Figure 5.5).

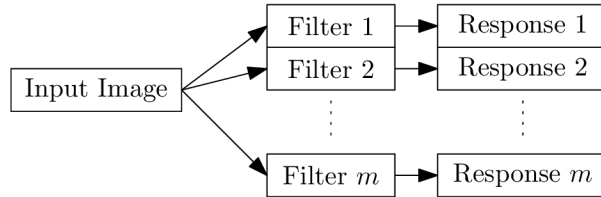


Figure 5.5: General filter bank.

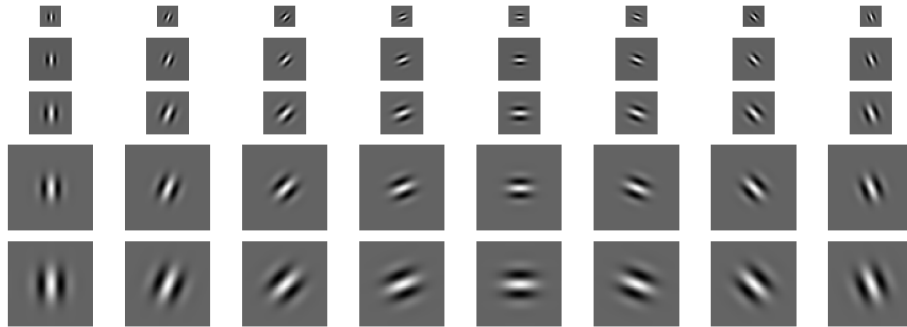


Figure 5.6: Gabor filter bank.

5.4.1 Gabor Filter Bank

The complex Gabor filter is defined as the product of a Gaussian kernel and a complex sinusoid:

$$g(x, y, \omega, \theta, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x'^2 + y'^2}{2\sigma^2}} \left(e^{i\omega x'} - e^{-\frac{\omega^2 \sigma^2}{2}} \right) \quad (5.7)$$

where x and y are coordinates within the Gabor kernel, $x' = x \cos \theta + y \sin \theta$ and $y' = -x \sin \theta + y \cos \theta$. It is often used as an image edge detector with respect to specific frequencies and orientations. The Gabor space is very useful in image processing applications such as optical character recognition [23], iris recognition [17] and fingerprint recognition [26]. The complex sinusoid is known as the *carrier* and the Gaussian-shaped function is known as the *envelope*. The rotation as well as the frequency of the carrier is controlled through the parameters θ and ω , respectively. The parameter σ controls the envelope size.

The Gabor filter is usually controlled with just two discrete-value parameters – orientation $o \in (0, 1, \dots, 7)$ and scale $s \in (1, 2, \dots, 7)$. The parameters ω , θ , and σ are set to: $\omega \leftarrow \frac{\pi}{2} \sqrt{2}^{-s}$, $\sigma \leftarrow \frac{\pi}{\omega}$, and $\theta \leftarrow \frac{o\pi}{8}$. The example of Gabor filter bank is in Figure 5.6.

The Figure 5.7 shows the application of the complex Gabor filter on the input shape index image. The Figure 5.8 shows the superposition ability of Gabor filter bank.

5.4.2 Gauss-Laguerre Filter Bank

The Gauss-Laguerre wavelets are polar-separable functions with harmonic angular shape. They are steerable in any desired direction by simple multiplication with a complex steering factor and as such they are referred to as self-steerable wavelets [2]. Our Gauss-Laguerre filter bank consists of 35 filters that were created with parameters $n \in (1, 2, 3, 4, 5)$, $k = 0$, $j = 0$ with sizes 16×16 , 24×24 , 32×32 , 48×48 , 64×64 , 72×72 , and 96×96 pixels.

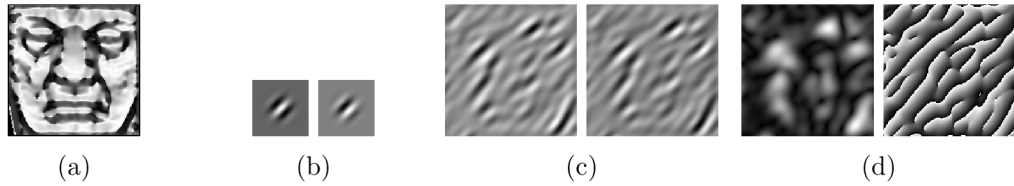


Figure 5.7: Application of the complex Gabor filter on the input shape index representation of the face surface. From left to right: input image (a), real kernel and imaginary kernels (b), real response and imaginary responses (c), and absolute response (magnitude) with angle response (d).

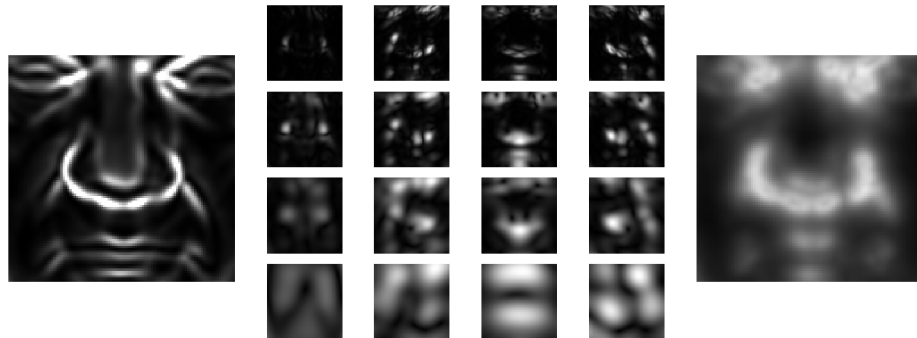


Figure 5.8: Superposition of individual absolute responses of complex Gabor filter. 16 different Gabor filters were applied on the input eigencurvature image (left). The responses are shown in the center grid. Four different frequencies (one for each line) and four different orientations (one for each column) were applied. The resulting superposition is shown on the right side of the figure.

The θ has been set to $\theta \leftarrow \text{atan2}(x, y)$ and $r \leftarrow \sqrt{x^2 + y^2}$. See Figure 5.9 for an example of Gauss-Laguerre filter bank.

5.4.3 Other Filters

Histogram Equalization

The histogram equalization [73] may improve face recognition based on the classic 2D photographs or on thermal imaging. It improves the contrast in an image in order to stretch out the intensity range. Equalization implies mapping one distribution (the given histogram) to another distribution (a wider and more uniform distribution of intensity values) so the intensity values are spread over the whole range. Figure 5.10 shows the impact of histogram equalization on the set of images that belong to the same subject but the lighting conditions vary.

Gaussian blur and difference of Gaussians

Gaussian blur filter may improve recognition robustness against noise and wrong face alignment. The *Difference of Gaussians* (DoG) [80] works as a bandwidth filter. Shading induced by surface structure is a potentially useful visual cue but it is predominantly low spatial frequency information that is hard to separate from effects caused by illumination gradients. Suppressing the highest spatial frequencies potentially reduces both aliasing and noise

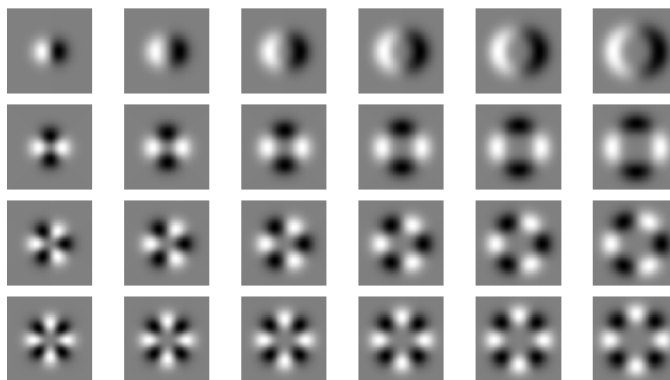


Figure 5.9: Gauss-Laguerre filter bank.



Figure 5.10: The example of application of histogram equalization filter. The upper row contains photographs of the same subject in different lighting conditions. The lower row contains the same scans with histogram equalization applied.

without destroying too much of the underlying recognition signal.

Local Binary patterns

The *Local Binary Pattern* (LBP) [81, 80, 24] operator labels the pixels within image by thresholding the 3×3 neighborhood with the center value and considering the result as a binary number. At a given pixel, LBP is defined as an ordered set of binary comparisons of pixel properties between the center pixel and its eight surrounding pixels. The decimal form of the resulting 8-bit word (LBP code) is used to represent the detail property of the center pixel. The example of application of LBP filter is in Figure 5.11.

Local binary patterns are often used with spatial histograms – the image is divided into grid and histograms are calculated within each cell. Concatenated histograms may form the feature vector directly or they can be further processed with some subspace projection.

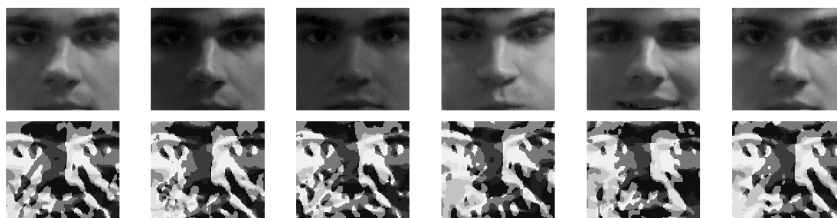


Figure 5.11: The example of application of the LBP filter. The input images are in the first row while the results after filter application are in the second row.



Figure 5.12: Iso-geodesic curves.

5.5 Iso-geodesic curves

3D Face recognition utilizing iso-geodesic curves and iso-geodesic stripes has been described in Section 4.7.8. The extracted curves are sampled to a set of points in 3D space and directly processed by some subspace projection technique, i.e., PCA, LDA, and ICA.

The center of all facial iso-geodesic curves is the nose-tip N that has been previously located during the ICP align. n curves c_1, c_2, \dots, c_n with corresponding geodesic distances d_1, d_2, \dots, d_n are thus extracted. Each curve c_i consists of m points: $c_i = (p_{i_1}, p_{i_2}, \dots, p_{i_m})$, such that $d_{geo}(N, p_{i_j}) = d_i$, where $d_{geo}(\cdot, \cdot)$ denotes geodesic distance.

The individual points p_{i_j} of curve c_i are gained in the following manner: The neighborhood of the center is equally divided into m sectors such that the angle between individual sector beams is $\frac{2\pi}{m}$. On each sector beam, a point with a specific geodesic distance from the center is denoted. Example of 5 iso-geodesic curves with geodesic distance 1, 2, 3, 4, and 5 cm consisting of 100 points is shown in Figure 5.12.

5.6 Feature Extraction and Metric Selection

Although specific Gabor filter may reveal features important for the subject classification, the dimensionality of the face image space remains the same. Moreover, if we apply 10 Gabor filters on the image with size 50×50 pixels, the resulting dimensionality is $50 \cdot 50 \cdot 10 = 25000$. Therefore, the image is projected to some low-dimensional space using techniques described in Section 4.4.

In plain PCA, the components of the projected vector are proportional to the variability that is expressed as the corresponding eigenvalue. This unbalance of individual feature vector components may lead to a neglect of those feature vector components that may have positive impact on the recognition performance, however, their associated eigenvalue is too small. In order to avoid that, individual feature vector components can be normalized after the subspace projection using z-score normalization. That is, an arbitrary feature vector $X = (x_1, x_2, \dots, x_m)$ is modified such that $x_i \leftarrow \frac{x_i - \bar{x}_i}{\sigma_i}$, where \bar{x}_i is the mean value of the component i and σ_i is corresponding standard deviation – see Figure 5.13.

Usually, the basic Euclidean distance is used in order to compare two feature vectors. We have tried other metric functions as well – namely a sum of absolute differences (city-

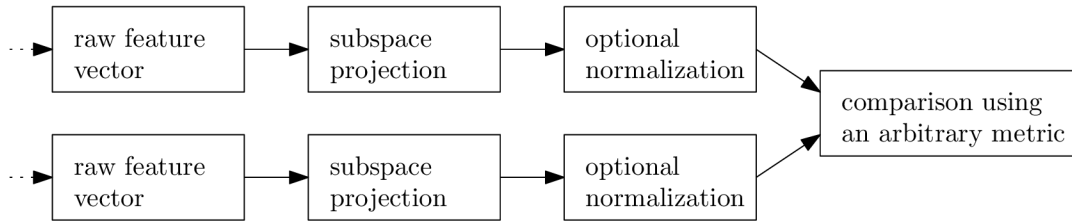


Figure 5.13: End of the recognition pipeline – raw feature vectors (iso-geodesic curves, processed images) are projected to low-dimensional space and optionally normalized using z-score normalization. The comparison is conducted using an arbitrary distance function.

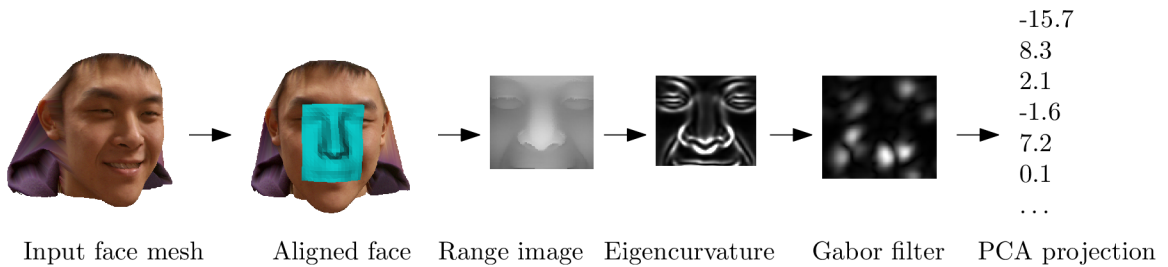


Figure 5.14: The example of one possible recognition unit. The input face mesh is aligned first. After that, the range image and subsequently eigencurvature is calculated. On the curvature image, Gabor filter is applied. The subspace projection using PCA is made as the last step.

block, Manhattan metric), cosine metric, and correlation metric.

The alternative to the PCA projection approach is the utilization of spatial histograms that are closely related to the LBP-based recognition [80]. After the application of the LBP filter, the image is divided into the grid. The grid cell size depends on the specific application as well as on the size of the input image. The image is converted to the grayscale representation and the histogram of the intensity values is calculated. Individual histogram values from all grid cells are concatenated and the resulting feature vector is thus created. The size d of the feature vector depends on the cell count: $d = r \cdot c \cdot 255$, where r and c are the numbers of grid rows and columns respectively. The resulting feature vectors are directly used for comparison in [3]. However, they can be also further processed with PCA and subsequent z-score normalization.

5.7 Multi-algorithmic Score-level Fusion

In the previous sections, the face alignment, image data extraction, image filters, iso-geodesic curves and subspace projections were described. While the face alignment is common for all recognition pipelines, image data, filters, and subspace projections are variables. One example of a possible recognition algorithm (*unit* in further text) is in Figure 5.14.

The basic principles of multi-algorithmic and score-level fusion were already described in Section 2.3. In this section, specific problems and issues regarding the practical implementation will be presented.

5.7.1 Score normalization

One of the most important concerns, when the score-level fusion is involved, is the score normalization [68, 70]. Comparison scores of individual units have to be normalized to some common domain prior to fusion itself.

There are several techniques of score normalization. Although the metric should be defined so that it satisfies the non-negativity axiom ($d(x, y) \geq 0$), a biometric comparison score s as well as the normalized value s' may be lower than zero. Let S is the set of all comparison scores from some evaluation run. S_{gen} and S_{imp} are sets of all genuine and impostor scores respectively ($S = S_{gen} \cup S_{imp}, S_{gen} \cap S_{imp} = \emptyset$).

Probably the simplest normalization is min-max:

$$s' = \frac{s - \min(S)}{\max(S) - \min(S)} \quad (5.8)$$

Min-max normalization is highly sensitive to outliers. Let $\overline{s_{imp}}$ and $\overline{s_{gen}}$ denote the mean impostor and genuine scores respectively. The normalized score s' from input score s is computed using the following formula:

$$s' = \frac{s - \overline{s_{gen}}}{\overline{s_{imp}} - \overline{s_{gen}}} \quad (5.9)$$

The normalization from Equation 5.9 transforms the input score, such that the mean impostor comparison value is 1 and mean genuine comparison value is 0. Further robustness against outliers may be achieved when the median instead of mean is used:

$$s' = \frac{s - \text{median}(S_{gen})}{\text{median}(S_{imp}) - \text{median}(S_{gen})} \quad (5.10)$$

Another frequently used normalization technique is z-score. Let σ and \bar{s} denote standard deviation and mean value of S respectively:

$$s' = \frac{s - \bar{s}}{\sigma} \quad (5.11)$$

The similar normalization technique is based on the *Median of Absolute Deviation* (MAD) [72]. MAD of set S is defined as:

$$\text{MAD}(S) = \text{median}\{|s_1 - \text{median}(S)|, |s_2 - \text{median}(S)|, \dots, |s_n - \text{median}(S)|\} \quad (5.12)$$

The MAD-based normalization uses the computed median and MAD of S :

$$s' = \frac{s - \text{median}(S)}{\text{MAD}(S)} \quad (5.13)$$

The normalization based on the hyperbolic tangent is defined as:

$$s' = \frac{1}{2} \left(\tanh \left(\frac{s - \bar{s}}{100\sigma} \right) + 1 \right) \quad (5.14)$$

5.7.2 Classifier-based fusion

Suppose that we have n recognition units. Each unit employs its own image processing, feature extraction using some subspace projection and comparison metrics. The resulting comparison scores provided by individual units are normalized using Equation 5.9. The task is to combine normalized scores to a single value that can be thresholded in order to decide whether an input scan is accepted or not.

From the machine-learning perspective, the task is to create a classifier C that is capable to assign a class label $c \in \{gen, imp\}$ (genuine or impostor) for a given vector of normalized scores $\mathbf{s} = (s_1, s_2, \dots, s_n)$:

$$C : \mathbf{s} \mapsto c \quad (5.15)$$

In order to have both the biometric security and user convenience configurable, the vector of scores is mapped to genuine likelihood or signed distance from the genuine/impostor decision hyperplane rather than class label.

Logistic Regression

The learning of fusion based on logistic regression requires preprocessing of the training data. First, the design matrix Φ is created:

$$\Phi = \begin{pmatrix} 1 & \mathbf{s}_1 \\ \vdots & \vdots \\ 1 & \mathbf{s}_n \end{pmatrix} \quad (5.16)$$

Each row i in Φ contains 1 in the first column followed by individual normalized comparison scores from i^{th} comparison.

The target column vector \mathbf{t} contains labels corresponding to individual comparisons: $\mathbf{t} = (c_1, c_2, \dots, c_n)^T$. c_i is set to 1 if the same (genuine) users were compared. In case of different (impostor) comparison holds $c_i = 0$.

The projection matrix \mathbf{W} is computed after that:

$$\mathbf{W} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{t} \quad (5.17)$$

The genuine/impostor classification of input normalized scores \mathbf{s} is based on the fusion score gained from the following equation:

$$s = \frac{1}{1 + \exp(-\mathbf{W}^T \psi)} \quad (5.18)$$

where $\psi = (1 \ \mathbf{s})$.

Support Vector Machine (SVM)

In fact, Support Vector Machine is an optimization problem. SVM attempts to find a hyperplane that divides the two classes with the largest margin. The support vectors are the points which fall within this margin. If the classes are not linearly separable, soft margin SVM is introduced. Parameter C controls the number of points that may stray over the line into the margin.

In this work, the implementation libSVM [13] is used.

Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis has been described in Section 4.4.2. However, it is used as the binary classifier in this case. The n -dimensional space, where n is the number of involved units, is projected to one-dimensional line that separates genuine and impostor scores clusters the best.

5.7.3 Hill-climbing Unit Selection

Beside the score normalization, another issue that has to be taken into account is to measure the score correlation and performance bias [69, 39]. If the score correlation between the involved units is high, the resulting recognition performance of the multi-algorithmic system may not be significantly better than the individual units. Moreover, huge performance bias between units may reduce the recognition performance.

The task is to select from the given set of units $U = (u_1, u_2, \dots, u_n)$ a non-empty subset S that achieves possibly the best recognition performance. The exhaustive search of the relatively small set U is simple. However, when the number of units exceeds certain thresholds, the exhaustive search is impossible.

We employ a wrapper selection [37]. This approach has been originally developed for feature selection, however, it can be also used for the unit selection in the multi-algorithmic fusion.

The optimization criterion is achieved EER (EER_i) of a particular unit i . The algorithm is as follows:

1. Select the unit b that achieves the best EER_b , remove it from the set U and add it to the set of selected units S .
2. For each remaining unit j in U measure fusion EER of the set $S \cup j$.
3. Select the best unit and add it to S . If there was no improvement of fusion EER, exit. Otherwise return to 2.

There is a potential drawback of the hill-climbing selection – selected units might be too specific for the training set and cannot generalize. Validation on the separate set is therefore recommended. However, our further experiments did not reveal any significant performance drop between the training and testing data. It has emerged that the hill-climbing is a good choice for recognition unit selection.

Chapter 6

Evaluation

The presented algorithm will be evaluated in this chapter. In order to compare the achieved results with other available state-of-the-art algorithms, evaluation is performed on the Face Recognition Grand Challenge version 2.0. Detailed methodology, tests as well as achieved results are described in the following sections.

The face recognition algorithm was also tested on the databases obtained with low-cost 3D sensors – such are Microsoft Kinect¹ and SoftKinetic DepthSense DS325². The expansion of personal depth sensors related with the new ways of the human-computer interaction in recent years markedly lowered the price of 3D acquiring devices for personal use. However, the biggest challenge of the face recognition based on the low-cost depth sensors is the quality of the acquired scans. While, for example, the Minolta Vivid or Artec 3D M scanners provide a highly precise geometry with an outstanding resolution and level of detail, the scans retrieved from the Kinect or DepthSense DS325 sensors are noisy, have low resolution and sometimes contain holes. The last two sections of this chapter are dedicated to the performance evaluation on those low-cost depth sensors.

6.1 Database Description and Evaluation Methodology

The proposed face recognition algorithm was trained and tested on the Face Recognition Grand Challenge Database v 2.0 - a standard evaluation database for facial biometrics (see Section 3.3.1). This database contains scans captured in spring 2003, fall 2003, and spring 2004. The Spring 2004 portion (2,114 scans) was divided into five parts, such that each part contains the same count of subjects. No subject is present in more than one part. The face alignment algorithm presented in the next section failed 36 times and thus these scans were removed. See Table 6.1 for the detailed information about dividing the Spring 2004 portion of the FRGC v2.0 database for evaluation purposes.

The first part of Spring 2004 was used for the training of the face alignment and the training of individual parameters of subspace projections. The second part was used for the validation of selected parameters and for the training of final fusion. The last three parts were used for evaluation purposes.

¹<http://www.xbox.com/kinect/>

²<http://www.softkinetic.com/products/depthsensecameras.aspx>

Table 6.1: FRGC2 Spring 2004 statistics.

| Part | Subjects | Scans | Purpose |
|------|----------|-------|----------------------|
| 1 | 69 | 416 | training |
| 2 | 69 | 451 | training, validating |
| 3 | 69 | 414 | evaluation |
| 4 | 69 | 417 | evaluation |
| 5 | 69 | 380 | evaluation |

Table 6.2: Comparison of two face alignment approaches.

| Align approach | EER |
|----------------|-------|
| Landmark-based | 0.060 |
| ICP | 0.043 |

6.2 Face Alignment

The alignment of the input face mesh is one of the most important tasks in the pre-processing part of the recognition pipeline. There are two main evaluation characteristics - precision and speed. The first mentioned can be implicitly evaluated by the evaluation of the overall biometric performance. The latter is important for practical purposes.

The alignment of the face can be performed in several ways. It can rely on the landmark detection. The inner corner of the eyes, nose tip, and nose corners are detected and the face is aligned subsequently such that the sum of absolute differences between landmarks on the input face and landmarks on the reference face template is minimal. Although the alignment of the predefined points is simple and fast to compute, the notable downside of this approach is the requirement of precise landmark detection. Just a slight inaccuracy can lead to a wrong alignment, and the recognition performance is thus negatively affected.

The second option of the face alignment is the involvement of *Iterative Closest Point* (ICP) algorithm. There is no initial landmark estimation. The entire face mesh model is aligned to the reference mean face template. However, this approach consumes much more CPU power.

Both alignment approaches were evaluated on the FRGC database. The first part of the Spring 2004 set was used for training and the second part for validation. The recognition was performed on the range images of size 120×120 pixels. Features were extracted using PCA and individual feature vector components subsequently normalized using z-score. The correlation metric was employed for comparing the resulting feature vectors. The evaluation results of both approaches are in Table 6.2.

The alignment utilizing the ICP algorithm outperformed landmark-based approach. Therefore it will be used in the following test. On the other hand, the drawback of the ICP is much higher CPU usage. The problem comes from the fact that there is no explicit mapping between the input face mesh and the mean reference template. In each ICP iteration, the points between the input and the reference have to be associated. This can be speeded-up when the index (e.g. k-means) of the points from the input face is created. However, the points in each iteration are changing their position and the index has to be recalculated. This issue might be solved with the inverse transformation trick described previously in Section 5.2.2. The comparison of mean duration of ICP alignment for FRGC scans is in Figure 6.1.

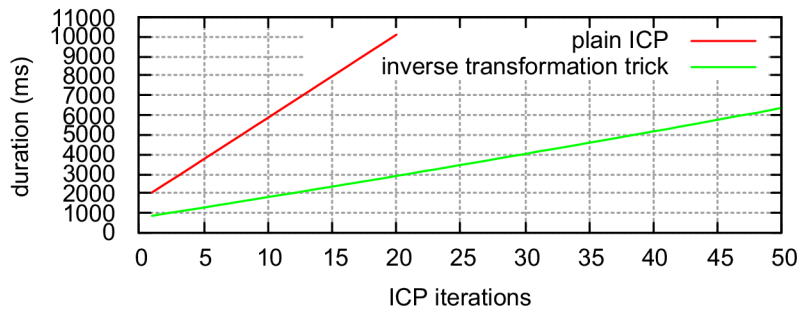


Figure 6.1: Comparison of ICP alignment approaches for FRGC scans. The duration (in milliseconds) is drawn for plain ICP as well as for ICP alignment with the inverse transformation trick.

Table 6.3: Selection of eigenvectors for the PCA projection (partial results only).

| Selection threshold (t) | Number of eigenvectors (T) | Achieved EER |
|-----------------------------|--------------------------------|---------------|
| 0.9 | 35 | 0.0751 |
| 0.95 | 75 | 0.0526 |
| 0.99 | 232 | 0.0425 |
| 0.995 | 296 | 0.0416 |
| 1 | 416 | 0.0430 |

6.3 Particular Parameters of the Recognition Algorithm

There are several important parameters common to all recognition units based on some filter applied to the image representation of the surface – region of interest, type of subspace projection and its specific parameters, and metric selection. The last two are also related with the recognition employing the iso-geodesic curves. All these parameters will be evaluated in this section in order to find the most suitable solution.

6.3.1 PCA parameters





First, let us focus on the PCA-related parameters, namely the number of eigenvectors that are used for the projection to the PCA subspace. This test was performed on the range images with size 120×120 pixels. The PCA was trained on the first part of the FRGC Spring 2004 set and evaluated on the second part of the same set. The correlation metric was employed.

The selection threshold t that controls the number of selected eigenvectors has been consecutively raising from 0.9 to 1 with step 0.001. Suppose that the PCA revealed N eigenvalues $(\lambda_1, \lambda_2, \dots, \lambda_N)$. The selection of T eigenvalues and corresponding eigenvectors was achieved by the fulfillment of the following equation:

$$t = \frac{\sum_{i=1}^T \lambda_i}{\sum_{j=1}^N \lambda_j} \quad (6.1)$$

The results are summarized in Table 6.3. The best result was achieved when the selection threshold $t = 0.995$ was used.

Table 6.4: ROI selection evaluation.

| ROI | width | height | EER |
|-----------------------------------------------------------------------------------|-------|--------|--------------|
|  | 120 | 120 | 0.040 |
|  | 100 | 90 | 0.038 |
|  | 100 | 120 | 0.039 |
|  | 120 | 90 | 0.039 |

Later on, the same test was repeated on the shape index images, texture images, and iso-geodesic curves. The optimal selection threshold $t = 0.995$ has been achieved again.

6.3.2 Region of Interest

In order to select only the rigid parts of the face that are not affected by the facial expressions, various *Regions of Interest* (ROI) were evaluated. The partial results of the evaluations are summarized in Table 6.4. The showed results were achieved on the plain range images using PCA projection, z-score normalization and correlation metric.

The best results are achieved when the area of mouth is limited. This is due to the fact that mouth is the area of the face that is affected by facial expressions the most. On the other hand, nose and eyes are the most stable areas from the intra-class variance perspective.

There is a trade-off between a too large area and a small area. The first can provide much more discriminative abilities but, on the other hand, it suffers from a high intra-class variance. The latter has a limited abilities for seeking for individualistic features.

The next question related to the ROI selection is the optimal size of the input image – if we resize the input image to just 50×45 pixels, will it still contain enough information in order to distinguish persons? The test suggests that neither the half size nor the double size significantly affects the recognition performance. Moreover, feature vectors extracted from the half-size image have much lower dimensionality an thus the evaluation is also faster. On the other hand, the double size range image does not contain any additional meaningful information.

6.3.3 Projection and Metric Selection

The crucial part of the recognition pipeline is also a feature extraction projection and a subsequent comparison utilizing some metric. Various metrics on the projected range images based on the parameters obtained in previous evaluations were tested – see Table 6.5.

The first notable conclusion is that the ICA and PCA with z-score normalization (zPCA in further text) employing either cosine or correlation metrics outperform any other combination. The more interesting fact is that they provide almost the same results. See Figure 6.2 for example of genuine score correlation between these recognition units.

Table 6.5: Metric selection. Achieved EERs for each combination of projection and comparison metrics are given.

| Projection | Metric | | | |
|--------------|--------------|-----------|------------|--------------|
| | Correlation | Euclidean | City-block | Cosine |
| PCA | 0.103 | 0.135 | 0.132 | 0.103 |
| PCA, z-score | 0.038 | 0.186 | 0.185 | 0.038 |
| Fisherface | 0.110 | 0.132 | 0.137 | 0.110 |
| ICA | 0.038 | 0.185 | 0.184 | 0.038 |

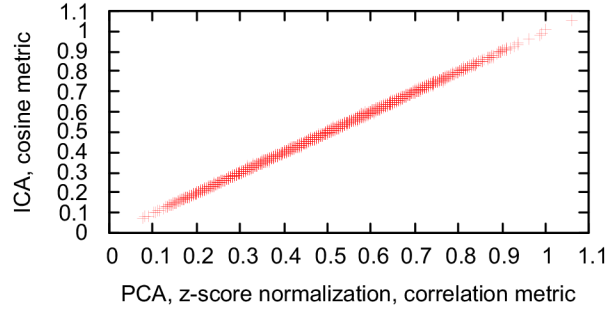


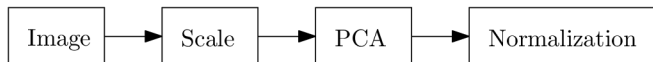
Figure 6.2: Correlation of genuine scores between PCA with z-score normalization employing correlation metric and ICA utilizing cosine distance metric.

If we look deeper, the correlation between zPCA and ICA is less surprising. It has been shown earlier that the most important of ICA calculation is the whitening process [87]. Moreover, since the individual PCA eigenvectors are decorrelated, the whitening process is similar to z-score normalization. The only difference between the cosine and correlation metrics is the calculation of mean values of the input vectors when the correlation metric is involved. However, for a given input, those mean values are constant and thus have no impact on the relative comparison of the two measured comparison scores (distances).

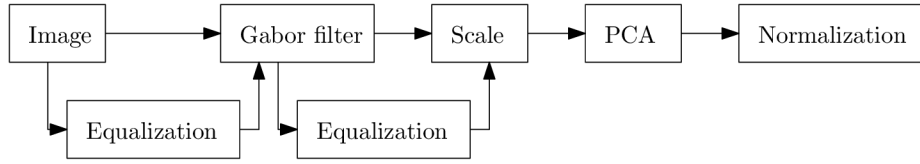
6.4 Evaluation of Individual Recognition Units

The individual recognition units were evaluated. Each unit is represented by input image data (e.g. texture, depth or curvature representation) on which some filters are applied. The resulting filter response is further processed with some feature extraction technique (e.g. PCA). The following types of units were tested:

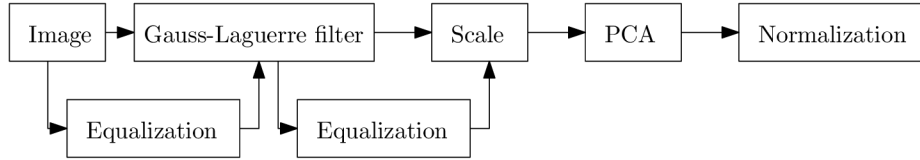
Plain image units – the input image (texture, depth, curvature) is scaled to half of its size, processed with PCA subspace projection and z-score normalized. Individual feature vectors are compared with correlation metric.



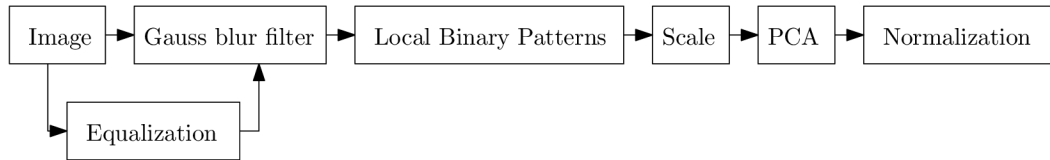
Gabor-based image units – the input image is processed by Gabor filter with a specific size and orientation, scaled to half size, projected using PCA and normalized. Correlation metric is used. Image is optionally equalized before or after the Gabor filter application.



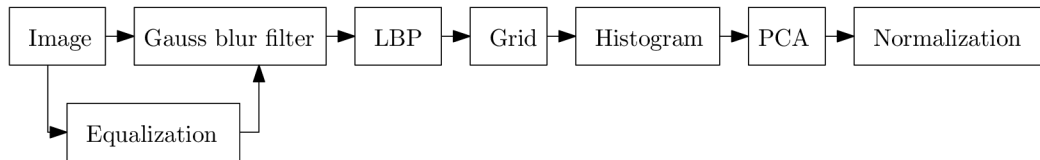
Gauss-Laguerre-based image units are similar to Gabor-based units, except that the Gauss-Laguerre filter is used.



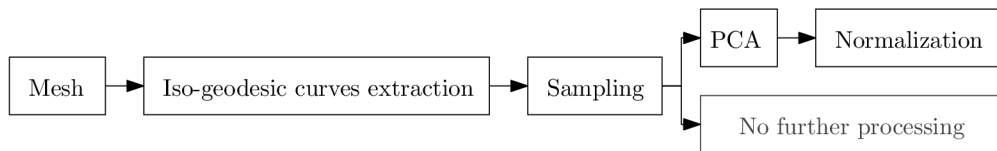
LBP units – the input image is equalized optionally. After that, it is blurred slightly with Gauss filter. Finally, the LBP filter is applied, the image is scaled to half of its size and projected using PCA. Z-score normalized feature vectors are compared with the correlation metric.



LBP units with histogram are similar to the plain LBP units but the resulting image after the LBP application is divided into a grid and the histogram of intensity values is calculated in each cell of the grid. The resulting histograms are concatenated and further processed with PCA.



Iso-geodesic curves – 5 iso-geodesic curves are extracted at a specific point with geodesic distance 1, 2, 3, 4, and 5 cm. Curves are sampled to 100 points per curve. After that, coordinates of individual points are concatenated to one column vector. It is compared with other feature vector using the city-block metric or processed with PCA and z-score normalized. The correlation metric is used in the second case.



There were evaluated 1,720 different units. The histogram of achieved equal error rates for each of them is in Figure 6.3. Partial results are in Table 6.6

The best unit consists of the following pipeline. The Gaussian curvature representation of the surface is blurred with the Gaussian kernel of size 11 pixels. This image is further processed with a local binary patterns filter. The resulting image is divided into 10 horizontal and 9 vertical cells. A histogram of values within each cell of size 10×10 pixels

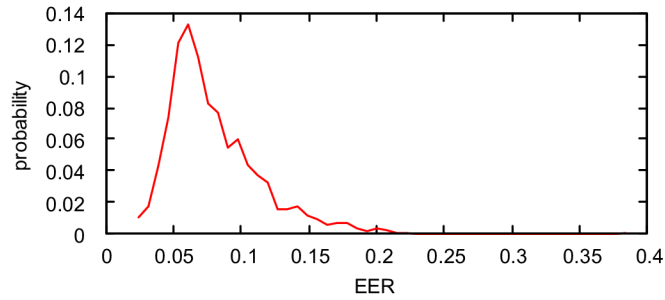


Figure 6.3: Histogram of EERs for all tested recognition units.

Table 6.6: Partial results of the units evaluation - the best representatives of each unit type.

| Rank | Type | Input data | Applied filters | EER |
|------|---------------------|--------------------|-------------------------------------|--------|
| 1 | LBP histogram | Gaussian curvature | gaussBlur(7); LBP; histogram(10,9) | 0.0247 |
| 7 | Gauss-Laguerre | Range image | gaussLag(48,1,0); scale(0.5) | 0.0267 |
| 12 | Gabor | Eigencurvature | gaborAbs(1,2); equalize; scale(0.5) | 0.0295 |
| 34 | LBP | Mean curvature | gaussBlur(11); LBP; scale(0.5) | 0.0361 |
| 73 | Plain | Range image | scale(0.5) | 0.0416 |
| 759 | Iso-geodesic curves | | 5 curves centered at the nosetip | 0.0769 |

is calculated. A set of histograms is further processed with PCA in order to reduce the correlation of values as well as the number of feature vector components. The entire process is depicted in Figure 6.4.

Another example is 7th best unit. A range image (depthmap) is convolved with both real and imaginary Gauss-Laguerre kernels of size 48 pixels and parameters set to $n = 1$ and $k = 0$. The absolute response is calculated from real and imaginary responses. The resulting image is scaled with factor 0.5 to size 50×45 pixels and finally processed with PCA.

The 12th best unit applies Gabor filter with scale 1 and orientation 2 on the eigencurvature surface representation. The histogram of absolute response is equalized and scaled with factor 0.5. As in the previous cases, PCA is applied on the image to reduce the feature vector size.

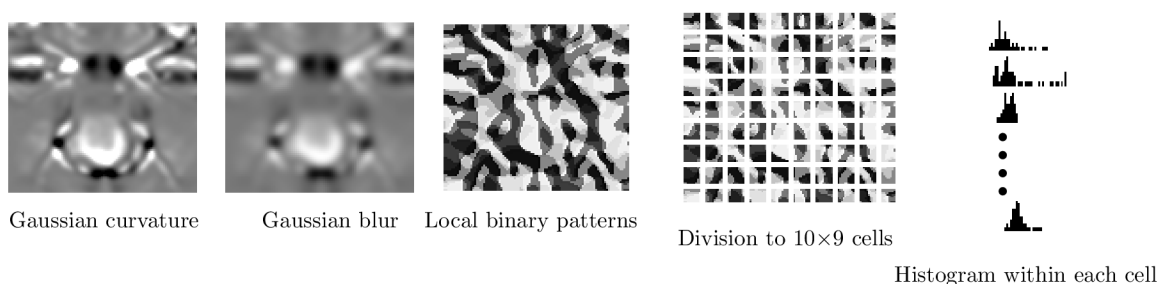


Figure 6.4: LBP-based recognition unit. The input Gaussian curvature image is blurred using Gaussian kernel. After that, local binary patterns are calculated. The image is divided into a grid and a histogram of intensity values is calculated within each grid cell. Individual histogram values are concatenated and further projected using PCA. The feature vector is thus created.

Table 6.7: Hill-climbing selection of individual units for SVM-based score-level fusion classifier.

| Iteration | Input data | Unit | | Unit EER | Fusion EER |
|-----------|--------------------|------------------------------------|---------|----------|------------|
| | | | Filters | | |
| 1 | shape index | GaussBlur(7); LBP; histogram(10,9) | | 0.0247 | 0.0247 |
| 2 | range image | GaussLag(48,1) | | 0.0266 | 0.0114 |
| 3 | texture | Equalize(); GaussLag(64, 4) | | 0.0963 | 0.0075 |
| 4 | shape index | Gabor(3, 2); Equalize() | | 0.0585 | 0.0063 |
| 5 | mean curvature | Gabor(2, 4); Equalize() | | 0.0657 | 0.0050 |
| 6 | texture | Gabor(3, 2) | | 0.1391 | 0.0043 |
| 7 | eigencurvature | Gabor(4, 2); Equalize() | | 0.0522 | 0.0035 |
| 8 | shape index | Equalize(); GaussLag(48, 5) | | 0.0881 | 0.0030 |
| 9 | | iso-geodesic curves | | 0.1163 | 0.0028 |
| 10 | range image | Equalize(); Gabor(7, 7) | | 0.1079 | 0.0022 |
| 11 | Gaussian curvature | Equalize(); GaussLag(64, 1) | | 0.0564 | 0.00175 |
| 12 | mean curvature | Gabor(7, 0); Equalize(); | | 0.1014 | 0.00174 |
| 13 | eigencurvature | Gabor(1, 0) | | 0.0642 | 0.00173 |

6.5 Multi-algorithmic Fusion

6.5.1 Score Normalization Techniques

Score-normalization is an important task preceding the fusion itself. Individual comparison scores have to be transformed into common domain in order to combine them meaningfully.

Figure 6.5 shows the score normalization result of three different recognition units. Each unit employs specific comparison metric and thus the score ranges vary. Every graph shows an impostor-genuine distribution of the comparison scores. The red curve belongs to the unit employing the correlation metric, the green curve corresponds to the unit with the Euclidean comparison and the blue curve corresponds to the unit with the city-block comparison metric. The graph in the first row shows the original pre-normalized values. Each subsequent graph shows normalized values where the normalization was achieved using one of the techniques previously described in Section 5.7.1.

The good normalization technique is able to align the curves of the impostor-genuine distribution that comes from different recognition units having feature vectors of different dimensionality and employing various metrics. It can be also measured implicitly by evaluation of the overall biometric performance. It has emerged that simple mean min-max normalization (see Equation 5.9) is the best choice for our purposes.

6.5.2 Greedy Hill-climbing Unit Selection

The greedy hill-climbing unit selection for final fusion was described previously in Chapter 5.7.3. All units from Section 6.4 were used as an input to the hill-climbing selector. The score-level fusion was provided by binary SVM classifier. Another classifiers, density-based and combination techniques were used as well in further experiments.

The hill-climbing selector chose 13 units – see Table 6.7. In the first iteration, the unit employing the application of LBP histogram on the shape index image was selected. The subsequent iteration chose a specific Gauss-Laguerre filter applied on the range (depth) image. The equalized texture followed by the application of Gauss-Laguerre filter was selected in the third iteration.

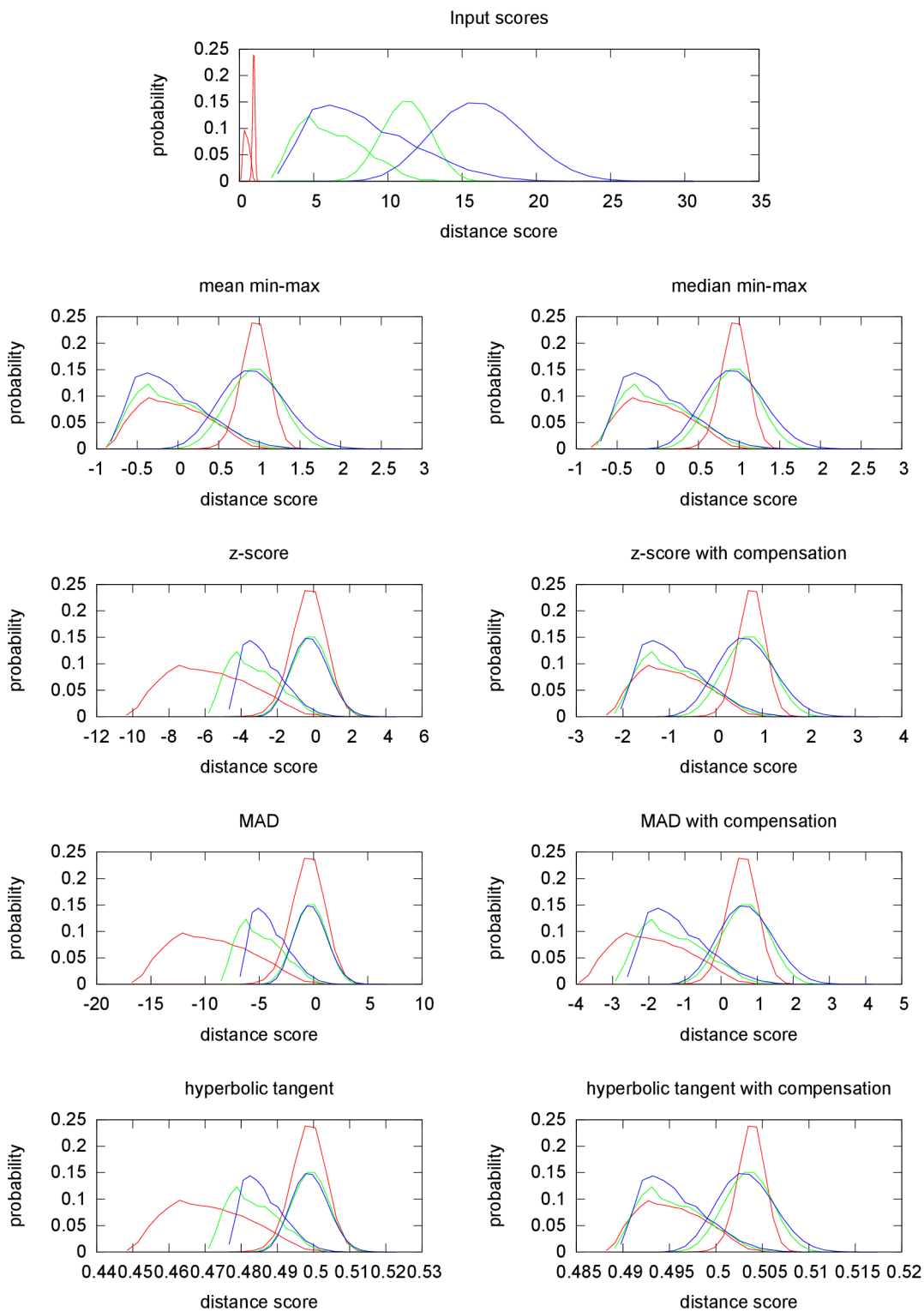


Figure 6.5: Score normalization techniques - genuine/impostor score distributions for unis employing correlation metric (red), city-block metric (green) and Euclidean distance (blue).

Table 6.8: Comparison of individual score-level fusion techniques (evaluated on the 3rd part of Spring 2004 portion).

| Fusion technique | EER on training set | EER on testing set | FNMR at given FMR | | |
|------------------|---------------------|--------------------|-------------------|--------|---------|
| | | | 0.001 | 0.0001 | 0.00001 |
| sum | 0.0096 | 0.0106 | 0.0459 | 0.0968 | 0.1838 |
| weighted sum | 0.0091 | 0.0106 | 0.0424 | 0.0954 | 0.1824 |
| GMM | 0.0050 | 0.0099 | 0.0353 | 0.0600 | 0.1497 |
| LogR | 0.0069 | 0.0112 | 0.0332 | 0.0912 | 0.1478 |
| LDA | 0.0083 | 0.0105 | 0.0424 | 0.0968 | 0.1669 |
| SVM | 0.0017 | 0.0096 | 0.0388 | 0.0721 | 0.1258 |

6.5.3 Comparison of Fusion Techniques

In the subsequent experiment, individual fusion techniques were compared. The selection of 13 units gained by the hill-climbing was used as the input to the training of multi-algorithmic fusion. Moreover, new scans were evaluated in order to test robustness of the fusion techniques. The results are shown in Table 6.8.

The transformation-based fusion is represented by a simple sum rule and the weighted sum. The weights of individual units are proportional to the achieved EER on the training set. For example, if the EER of the unit i is e_i , the corresponding weight is set to $w = 0.5 - e_i$.

The only representative of the density-based fusion is *Gaussian Mixture Model* (GMM). The probability density distributions of impostor as well as the genuine scores were modeled using 5 Gaussians with diagonal covariance matrices.

The classifier-based fusion is represented by logistic regression, LDA, and SVM. A plain linear kernel was used. The experiments suggest that there is no significant difference in recognition performance between the individual fusion techniques.

6.6 Comparison with the State-of-the-art

The results of Face Recognition Vendor Test 2006 (FRVT) was based on the evaluation of almost the entire FRGC dataset [65]. The performance of a biometric system vary with different sets of biometric samples. It is important to measure both the overall performance of a biometric system and the scale of the variability to measure statistical uncertainty. In the FRVT, the performance variability is measured by partitioning the test images into a set of smaller test sets. The performance is then computed on each of the partitions. According to the FRVT report, 3,589 out of the 5,000 scans in FRGC were used for the evaluation. These scans were divided into 13 partitions with the total count of 330 subjects. Unfortunately, the selection of these 3,598 scans is not clear from the report. Therefore, we bring the comparison of best algorithms involved in FRVT with our achieved results reduced only to Spring 2004 part in this section.

We have utilized the Spring 2004 FRGC subset such that its evaluation part contains 1211 scans from 207 individuals designated for the evaluation. This evaluation subset was divided into 3 partitions, as it was mentioned earlier in Section 6.1. Table 6.9 shows achieved results with a classifier utilizing SVM-based fusion. The comparison with other FRVT competitors is in Table 6.10. While the Viisage algorithm outperforms all others, our algorithm achieves the second best results.

However, the presented comparison is for illustration purposes only, since we did not

Table 6.9: Achieved results on the Spring 2004 evaluation subset.

| Evaluation partition | FNMR at given FMR | | |
|----------------------|-------------------|--------|--------|
| | 0.01 | 0.001 | 0.0001 |
| 1 | 0.0091 | 0.0388 | 0.0721 |
| 2 | 0.0098 | 0.0329 | 0.0596 |
| 3 | 0.0167 | 0.0589 | 0.1091 |
| Median | 0.0098 | 0.0388 | 0.0721 |

Table 6.10: Comparison of our method with FRVT competitors. The exact numbers were taken from the graphs in appendix section of FRVT report [65].

| Algorithm Name / Organization | Abbrev. | Median FNMR at given FMR | | |
|-------------------------------|---------|--------------------------|-------|--------|
| | | 0.01 | 0.001 | 0.0001 |
| Cognitec | Cog1 | 0.050 | 0.070 | 0.160 |
| Geometrix | Geo | 0.035 | 0.085 | 0.155 |
| Univ. of Houston | Ho1 | 0.030 | 0.050 | 0.100 |
| | Ho3 | 0.025 | 0.050 | 0.095 |
| Tsinghua Univ. | Ts1 | 0.035 | 0.145 | - |
| | V | 0.005 | 0.020 | 0.070 |
| Viisage | Va | 0.010 | 0.055 | 0.170 |
| | | | | |
| Our method | | 0.010 | 0.039 | 0.072 |

follow the same evaluation methodology as in the FRVT. One of the goals of this thesis is the utilization of low-cost depth sensors for the 3D face recognition. The comprehensive evaluation on databases obtained with SoftKinetic DS325 and Microsoft Kinect 360 is presented in the next two sections.

6.7 Evaluation on SoftKinetic Database

We have created the SoftKinetic database during spring 2014. It contains 398 scans from 52 individuals. During the capturing, the emphasis was put on following points:

- Various lighting conditions.
- Various (but limited) facial expressions – we allowed the subjects to have a slight smile, lifted eyebrows or frowned face.
- Scanning of some subject was splitted into several sessions in different days.
- An effort was made to have a diversity in gender, race, and age of scanned subjects.

The example of some scans in the SoftKinetic database is in Figure 6.6. Contrary to the scans acquired with the Minolta Vivid scanner (FRGC database), the data captured with SoftKinetic DepthSense DS325 sensor suffer from high noise among the z axis [42, 18], therefore some sort of denoising has to be applied on the 3D models in the preprocessing portion of the recognition pipeline.

Although one can use a stronger Gaussian smooth filter, our experiments show that much better, in terms of recognition performance, is the application of the feature-preserving mesh denoising algorithm [79]. An example of application of such filter is in Figure 6.7.



Figure 6.6: Example of scans in the SoftKinetic database (processed, aligned, and cropped).

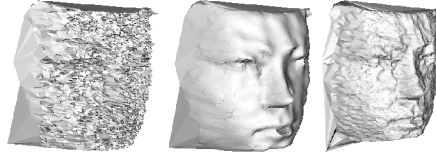


Figure 6.7: Application of feature preserving mesh denoising – before (left) and after (middle). Basic Gaussian smoothing is on the right side of the figure.

The field of view of DS325 sensor is very wide ($74^\circ \times 58^\circ \times 87^\circ$)³. Therefore, the quality and resolution of face scans rapidly decreases when the subject moves away from the sensor. Two scans from the same subject acquired from the distance of 35cm and 70cm are depicted in Figure 6.8. While the near-scan contains 6,951 verticies, the far scan contains only 1,560 verticies, which is more than 4 times fewer.

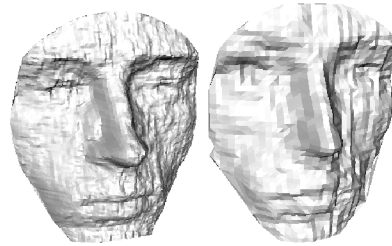


Figure 6.8: Two scans acquired with the SoftKinetic DS325 sensor captured from 35cm (left) and 70cm (right).

6.7.1 Finding Suitable Smoothing and Denoising Algorithm

The initial test evaluated recognition performance on the range images and shape index images. The scans from the training part of the SoftKinetic dataset were subsequently smoothed and aligned using ICP algorithm. After that, range images and shape index images were calculated. PCA trained on FRGC depth and shape index images with zScore normalization was used in order to extract features. Feature vectors were compared using correlation metric.

Both Gaussian smooth filter (Gauss) and feature-preserving mesh denoising (M-Denoise) algorithms with various parameters were evaluated. The results are in Table 6.11. The lowest EER was achieved when the M-Denoise filter was applied.

³<http://www.softkinetic.com/en-us/products/depthsensecameras.aspx>

Table 6.11: Evaluation of mesh smoothing and denoising algorithms on the SoftKinetic dataset.

| Smoothing method | Smooth iterations | Smoothing parameter | EER on range images | EER on shape index images |
|------------------|-------------------|---------------------|---------------------|---------------------------|
| None | - | - | 0.077 | 0.135 |
| M-Denoise | 5 | 0.01 | 0.053 | 0.116 |
| M-Denoise | 5 | 0.02 | 0.052 | 0.114 |
| M-Denoise | 5 | 0.04 | 0.052 | 0.108 |
| M-Denoise | 10 | 0.01 | 0.044 | 0.111 |
| M-Denoise | 10 | 0.02 | 0.048 | 0.103 |
| M-Denoise | 10 | 0.04 | 0.046 | 0.105 |
| M-Denoise | 20 | 0.01 | 0.041 | 0.097 |
| M-Denoise | 20 | 0.02 | 0.041 | 0.095 |
| M-Denoise | 20 | 0.04 | 0.040 | 0.096 |
| Z-Smooth | 5 | 0.2 | 0.068 | 0.131 |
| Z-Smooth | 5 | 0.5 | 0.061 | 0.120 |
| Z-Smooth | 5 | 1.0 | 0.057 | 0.114 |
| Z-Smooth | 10 | 0.2 | 0.067 | 0.119 |
| Z-Smooth | 10 | 0.5 | 0.061 | 0.119 |
| Z-Smooth | 10 | 1.0 | 0.052 | 0.110 |
| Z-Smooth | 20 | 0.2 | 0.061 | 0.121 |
| Z-Smooth | 20 | 0.5 | 0.053 | 0.119 |
| Z-Smooth | 20 | 1.0 | 0.042 | 0.111 |

6.7.2 Multi-Algorithmic Fusion

The multi-algorithmic fusion similar to the fusion used for FRGC (see Section 6.5) was trained and evaluated. The hill-climbing optimization selected 10 units. Contrary to the FRGC fusion, no iso-geodesic or LBP-based recognition unit is present. The lower quality of SoftKinetic scans causes the absence of the iso-geodesic curves unit. On the other hand, missing LBP-based unit is not so obvious. The selected units are in Table 6.12.

The DET curves from the evaluation of the training as well as test parts of the SoftKinetic dataset are in Figure 6.9. The impostor-genuine score distributions are in Figure 6.10 and the particular achieved FNMRs at given FMRs are in Table 6.13.

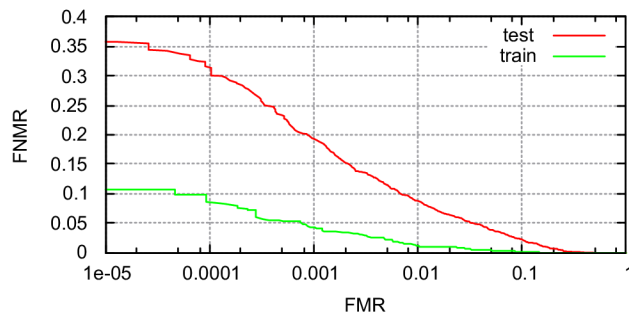


Figure 6.9: DET curves from the SoftKinetic evaluation.

Table 6.12: Selected recognition units gained from the training part of the SoftKinetic dataset.

| Unit | Input data | Filters |
|------|----------------|--------------------------|
| 1 | range image | GaussLag(72, 2) |
| 2 | texture | Equalize(); Gabor(5, 6) |
| 3 | range image | Gabor(3, 1) |
| 4 | texture | Equalize(); Gabor(6, 0); |
| 5 | eigencurvature | GaussLag(24, 5) |
| 6 | texture | Gabor(4, 2) |
| 7 | range image | GaussLag(48, 0); |
| 8 | shape index | GaussLag(96, 1) |
| 9 | shape index | Gabor(2, 5) |
| 10 | texture | Equalize(); Gabor(2, 7) |

Table 6.13: Evaluation on SoftKinetic database results.

| Data | EER | FNMR at given FMR | | |
|-----------|-------|-------------------|-------|--------|
| | | 0.01 | 0.001 | 0.0001 |
| Train set | 0.011 | 0.012 | 0.043 | 0.098 |
| Test set | 0.043 | 0.087 | 0.192 | 0.312 |

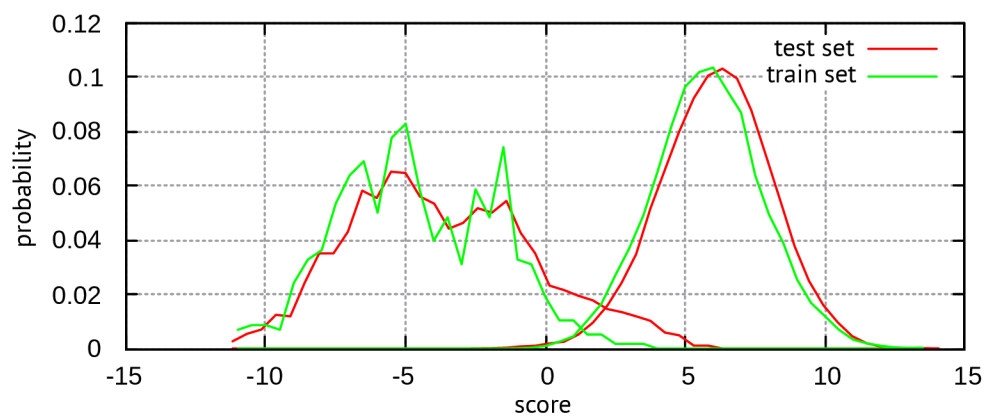


Figure 6.10: Impostor-genuine score distributions from the SoftKinetic evaluation.



Figure 6.11: Scans of one subject from all sessions in SoftKinetic dataset.

6.7.3 Real-World Scenarios

The real-world implementation of biometric system has to deal with the problems that are not always considered when it is being evaluated in the laboratory conditions. Usually, the users are not experienced enough in order to position their face such that the biometric system achieves the best results. The lighting conditions vary and the mood of users affecting their 3D face appearance also changes. All these factors can decrease the biometric performance rapidly.

The most convenient scenario for the users of an access-control biometric system is the identification. Users do not need to claim their identity before the scanning process begins. Based on their identity, the system decides if they have an authorization to proceed.

In this subsection, the template aging and identification with respect to the real-world application will be evaluated. The enrollment of a new user to the biometric system usually consists of acquiring several (four) scans for the creation of a reference template. The problem is that these scans are acquired in very short time period (just few seconds) and the lighting conditions are still the same. Moreover, the users use the biometric system for the first time, they are fully concentrated on the capture process and thus they have no facial expressions. When they use the biometric system later, the environment condition might change as well as the user's mood might affect mimics of the face.

Subjects with more than or equal of 15 scans were selected from the SoftKinetic dataset. When the dataset was captured, 5 scans were acquired in each session. This means that if the subject participated in 4 sessions, 20 scans of this particular subject are stored in the dataset. The sessions took place in different days in different places. The example of all scans of one subject from all sessions is shown in Figure 6.11.

The first four scans of each subject were used for creating of the gallery templates. The remaining scans were used for the evaluation. When the input probe is compared with the gallery templates, the arithmetic mean of individual comparisons between the probe and the templates is returned. The time evolution of the comparison scores is in Figure 6.12. If the decision threshold is set to 0 or lower, there will be just one false reject – last scan of subject subj4.

Once we have more than one reference template (4 in this case), we can have several decision strategies to compare the input feature vector with the stored templates. The simplest approach is a $1 : N$ comparison. The resulting identity is based on the reference template with the smallest distance between the input feature vector i and template t_i , where $i \in \{1, 2, \dots, N\}$ (see Figure 6.13 left). The more robust method that can handle outlying reference template is in the same figure on the right. The classification is based on the average distance to the k nearest reference templates for class C_j . Let the $\min_k S$ denote the k minimal elements in set S . The distance between input vector X and class

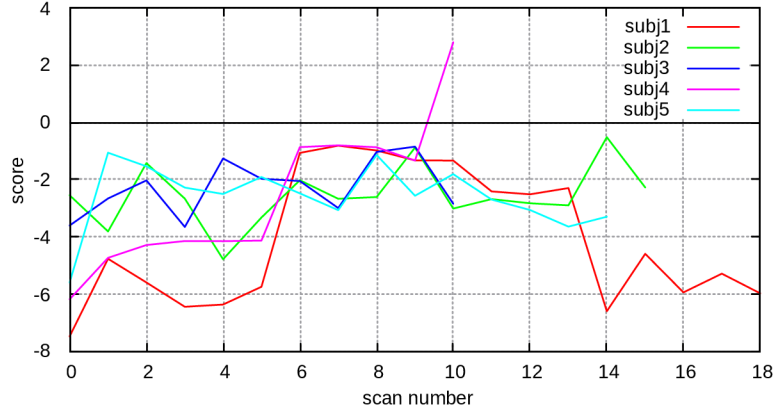


Figure 6.12: Time evolution of the comparison scores from SoftKinetic dataset subjects with more than or equal of 15 scans. Except for the last scan of the subject 4, all other scans were successfully verified with a comparison score lower than zero.

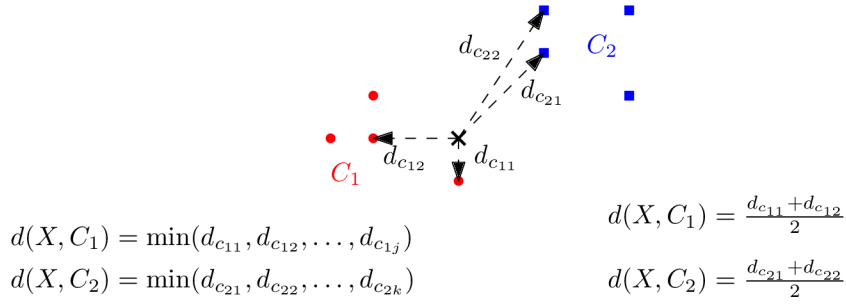


Figure 6.13: Classification strategies for identification in biometric systems. Basic 1 : N comparison is on the left, more robust approach is on the right hand side.

$C_j = \{c_{j_1}, c_{j_2}, \dots, c_{j_m}\}$ is:

$$d(X, C_j) = \frac{\sum_k \min_k \{d(X, c_{j_1}), d(X, c_{j_2}), \dots, d(X, c_{j_m})\}}{k} \quad (6.2)$$

The question is, if the system employing the modified distance metric from Equation 6.2 outperforms basic 1 : N comparison. The results are in Table 6.14. There are two approaches for the creation of the reference templates. The first (denoted *without randomizing*) was described previously – the first 4 scans from each subject were used for template, remaining scans were used for the evaluation. The latter approach (denoted *with randomizing*) randomly selects 4 scans for the template creation among all subject scans. The results are a bit surprising. The modified distance metric does not outperform the simple 1 : N comparison. Moreover, it is worse, when the randomized templates are used. On the other hand, the randomized templates outperform the non-randomized significantly. This was, however, expected, as the randomized templates capture more intra-class variability.

6.8 Kinect Evaluation

We would like to show that the proposed algorithm is robust enough in order to be easily adapted to any depth sensor. Kinect cannot be power supplied by a USB cable and re-

Table 6.14: Identification evaluation on SoftKinetic dataset.

| Reference template count (k) | without randomizing | | with randomizing | |
|-------------------------------------|---------------------|-------------------|------------------|-------------------|
| | EER | FNMR @ FMR = 0.01 | EER | FNMR @ FMR = 0.01 |
| 1 | 0.147 | 0.278 | 0.028 | 0.052 |
| 2 | 0.152 | 0.310 | 0.052 | 0.084 |
| 3 | 0.142 | 0.321 | 0.063 | 0.121 |
| 4 | 0.142 | 0.310 | 0.073 | 0.136 |

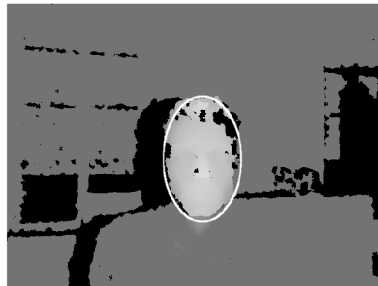


Figure 6.14: Range image of the scanned face. Face within the image is marked with the white ellipse.

quires an external power adapter. This restricts its usage in the embedded face recognition systems. On the other hand, since it is quite often used in households, the utilization of a 3D face recognition is shifted from security applications to home entertainment.

Contrary to the SoftKinetic DepthSense DS325, Kinect sensor is designed for scanning of the entire room and full-body capturing. The minimal distance where the depth is captured is 80cm, but the practical limit for capturing is rather 120cm. Moreover, the fields of view of both the depth sensor and the RGB sensor are very wide such that almost entire room is captured. When a subject is scanned with Kinect, just 10% of sensor is used. On the other hand, Kinect scans require less denoising treatment than those captured with SoftKinetic DS325. Therefore, the biggest challenge of the Kinect dataset is the small resolution of the input meshes rather than the noise. For an example of Kinect range visualization with a marked face region see Figure 6.14. An average Kinect face mesh contains about 5,000 vertices.

Our Kinect database consists of 108 scans divided into two parts - training and testing. Each part contains 9 different subjects that provided 6 scans. Facial expressions as well as varying lighting conditions are present in some scans. The example of Kinect database is in Figure 6.15.

The DET curve of our recognition algorithm evaluated on the Kinect database is in Figure 6.16. SVM classifier has been used for the final fusion of the individual recognition



Figure 6.15: Some examples from the Kinect database.

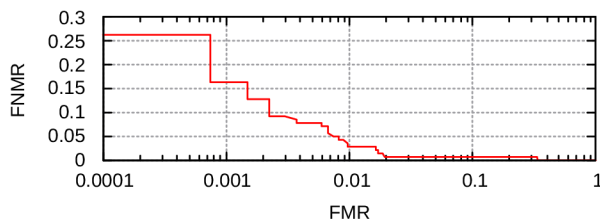


Figure 6.16: Evaluation of SVM fusion on the Kinect database – DET curve.

Table 6.15: Hill-climbing selection of individual units for SVM-based score-level fusion classifier on the Kinect database.

| Iteration | Input data | Unit Filters | Unit EER | Fusion EER |
|-----------|----------------|------------------------------------------------|----------|------------|
| 1 | texture | DoG(5, 3); GaussBlur(11); LBP; histogram(10,9) | 0.0639 | 0.0639 |
| 2 | range image | GaussLag(64, 2) | 0.0983 | 0.0337 |
| 3 | eigencurvature | Gabor(5, 5) | 0.3191 | 0.0210 |
| 4 | mean curvature | Gabor(2, 6) | 0.3601 | 0.0152 |
| 5 | mean curvature | Gabor(6, 5) | 0.3555 | 0.0137 |
| 6 | shape index | GaussLag(64, 2) | 0.2778 | 0.0134 |

units. The process of selecting the individual units is in Table 6.15. Iso-geodesic curves were not selected for the final fusion. This is probably due to the fact that the Kinect scans are quite rough and thus the curves do not contain much discriminative ability. On the other hand, a unit utilizing texture images processed with the *Difference of Gaussians* (DoG) filter was selected in the first place.

6.9 Limitations of Face Biometrics

6.9.1 Analysis of Facial Mimics

Our application for acquiring the facial scans is able to deal with slight rotations of the face. On the other hand, cooperation is still required from users. Perhaps the biggest challenge for secure and convenient face biometric system is dealing with facial mimics. The intra-class variability for some facial expressions may outweigh the inter-class variability of the biometric modality.

Intra-class (same-class, s) variability of a specific feature vector component can be expressed mathematically as the mean of standard deviations within each subject/class. On the contrary, the inter-class (between-class, b) variability can be expressed as the standard deviation of the class means. Since it is good to have intra-class variability as low as possible and the inter-class variability as high as possible, the *Discriminative Ability* (DA) of the specific feature vector component f could be expressed as:

$$DA_f = b_f - s_f \quad (6.3)$$

DA tells us how much we can rely, in terms of biometric recognition performance, on a specific feature. We can evaluate the discriminative ability of the input face image, not just the feature vector. Figure 6.17 shows the graphical representations of DA of texture, depth, and curvature images from both FRGC and SoftKinetic datasets. From the depth DA image, it is clear that the highest inter-class variability and the lowest intra-class

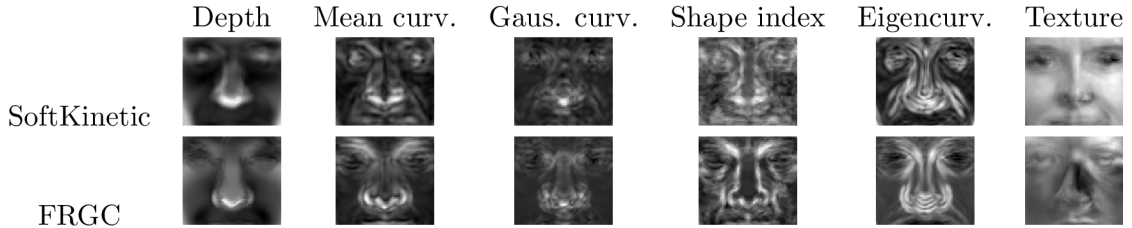


Figure 6.17: Discriminative Ability (DA) of face representation images. Brighter pixels denote areas with high DA, darker pixels correspond to low DA.



Figure 6.18: Application of DA weights on the input texture image.

variability is the nose and nosetip. High DA area is also around eyes. The lowest DA is at the mouth area – this in accordance with the face ROI selection experiments that were described previously in Section 6.3.2.

The next question is, whether the DA can be used to improve the recognition performance. There are two possible approaches of the involvement of DA. Pixels of the input image can be weighted by individual DA components and further processed by PCA and the feature vector is thus created (see Figure 6.18). The second possibility is the application of DA weights after the PCA projection. We have evaluated both methods and compared them with plain PCA-based feature extraction. The correlation metric was used for the feature vector comparison. The results are in Table 6.16. It has emerged that the utilization of DA does not improve recognition performance significantly. Moreover, when the range image or mean curvature was processed with DA, the recognition performance slightly decreased.

There are two major challenges in the face recognition – varying lighting conditions and facial expressions. Another challenge comes when the 3D model is involved – glasses. Especially low-cost depth sensors have a big problem with glass lenses. The structured light pattern is blurred or shifted as it comes through the lens and the reconstructed surface is thus imprecise. In the next experiment, we have evaluated all three mentioned challenges and evaluated their impact on the recognition performance. 5 scans from the same subject

Table 6.16: Evaluation of the application of Discriminative Ability on FRGC database.

| Image type | EER | | |
|--------------------|--------------|---------------|--------------|
| | Plain | DA before PCA | DA after PCA |
| Range image | 0.038 | 0.044 | 0.043 |
| Shape index | 0.056 | 0.043 | 0.068 |
| Mean curvature | 0.057 | 0.062 | 0.066 |
| Gaussian curvature | 0.066 | 0.064 | 0.077 |
| Eigencurvature | 0.085 | 0.066 | 0.087 |
| Texture | 0.080 | 0.090 | 0.079 |

were acquired using SoftKinetic DS325 sensor. The first scan was designated for the creation of the reference template. Remaining 4 scans were acquired in order to fulfill these scenarios:

- Glasses – subject was wearing dioptric glasses
- Smile – with widely opened mouth
- Neutral face
- Different lighting conditions – while the reference scan was captured in the room with a natural light source from the outdoors, this scan was captured with a closed curtain and artificial light from the lamp.

All 4 probe scans were consecutively compared to the reference scans using a previously trained multi-algorithmic classifier. This classifier is slightly different than the one presented in Section 6.7.2. Although it does not achieve such recognition accuracy, it contains a wider selection of the involved unit types and thus it is more suitable for a survey whether the specific unit type is more robust against the face recognition challenges.

The outcome from this experiment is in Table 6.17. The output score from the SVM fusion is in the first line. The SVM classifier is trained such that the output is zero for $FMR = 0.0001$. If the output score is greater than zero, probe scan is rejected. If it is lower than zero, the probe scan is accepted as a genuine user. The next lines in the table describe the output of the individual involved units (see Section 6.4 for more details). The reported values are already normalized using the min-max normalization. If the value is lower than zero it is very likely that it belongs to a genuine user (and it is denoted with green color in the table). On the contrary, if the normalized score for a particular unit is greater than 0.5, it might belong to the impostor and thus it is denoted with red color. It can be seen from the table that although some units may reject the user falsely when he is smiling, the overall SVM fusion score is still below zero and the user is still recognized. Different lighting conditions pose no difficulties either. On the other hand, the presence of glasses causes false reject. This is due to the fact that the geometry of the face is highly deformed under lenses – see Figure 6.19. There is an evident drop in depth at the frames and lenses area. One possible solution is mask out this area for recognition. However, this solution neglects a lot of otherwise important information because it was shown earlier that the area around nose and eyes have a very high Discriminative Ability.

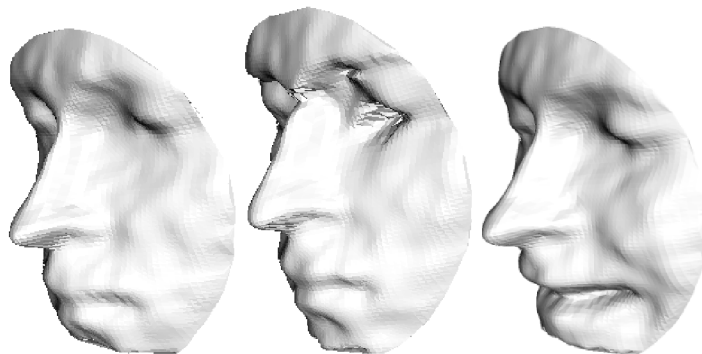


Figure 6.19: Three face meshes from the same subject acquired with DS325 sensor – without glasses (left), with glasses (center), and smiling (right).

Table 6.17: Comparison of output scores of the same subject in different scenarios.

| Unit | Glasses | Smile | Neutral | Different lighting conditions |
|-------------------------------|---------|--------|---------|-------------------------------|
| SVM Fusion | 0.028 | -2.569 | -7.991 | -7.368 |
| Range image | 0.351 | 0.440 | -0.118 | -0.123 |
| Texture; LBP histogram | -0.287 | -0.135 | -0.456 | -0.195 |
| Iso-geodesic curves | 1.058 | 0.170 | -0.120 | -0.118 |
| Eigencurvature; LBP histogram | 0.439 | 0.046 | -0.281 | -0.477 |
| Texture; Gabor | 0.276 | 0.525 | -0.542 | -0.359 |
| Mean curvature; Gabor | 0.308 | 0.361 | -0.953 | -0.493 |
| Texture; Gabor | -0.125 | 0.185 | -0.487 | -0.242 |
| Texture | 1.005 | 0.604 | -1.076 | -0.244 |
| Shape index; Gabor | -0.118 | 0.009 | 0.021 | 0.117 |
| Texture; Gabor | 0.117 | 0.519 | -0.409 | -0.181 |
| Gaussian curvature; Gabor | -0.255 | -0.516 | -1.004 | -0.908 |
| Texture; Gauss-Lagguere | 0.802 | 0.131 | -0.480 | 0.089 |

6.9.2 Tampering a Face Recognition System

There are many areas where a potential attacker may direct in order to fraud the biometric system:

1. Attack the biometric sensor – present a fake biometric sample to the sensor.
2. Attack the communication from the sensor – if the sensor and rest of the system is separated, attacker may intercept the data sent by the sensor.
3. Manipulating feature extraction and/or the template creation process – the attacker may inject his own feature vector instead of the vector calculated from the input data.
4. Attack the communication between the feature extraction unit and the comparison module.
5. Attack the comparison unit, e.g. modify the decision threshold for a success verification.
6. Attack the biometric database – attacker may inject his own malicious template directly to the database and thus bypass the proper enrollment.
7. Attack the transmission between the database and the comparison module – data may be corrupted, intercepted or modified.
8. Attack the final decision.

Despite its popularity and widespread, the face biometric systems, and especially classic 2D approaches, are vulnerable to presenting a fake input data. Since the other types of attack is common to all biometric systems, this subsection will be focused on presenting a fake samples to both 2D and 3D face biometric systems.

Compared to 2D and 2D/3D approaches, the classical 2D approach is easiest to fraud. If the system does not involve any liveness detection module, face recognition systems can be spoofed by facial pictures such as a portrait photographs [12]. There are several options how to recognize fake face sample:

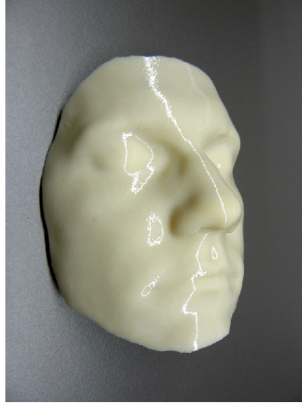


Figure 6.20: Fake 3D face model printed with the 3D printer. The data was obtained with Artec 3D M Sensor.

Movement of facial features [34, 62] approach detects eyes in sequential input images. Temporal variations of both eye regions are calculated subsequently. The assumption is that because of eyelid blinking and uncontrolled movements of the pupils, there should be shape variations.

Variable focusing [36] approach utilizes the variation of pixel values by focusing between two images sequentially taken in different focuses. Two sequential pictures focusing the camera on facial components are taken. One is focused on a nose and the other is focused on ears. A differences of *Sums of Modified Laplacians* (SML) between these two images are used for determination if the presented sample is real or fake. However, good camera with small depth of field is needed in order to decide correctly.

Optical flow estimation [5]. A liveness detection method utilizing differences in optical flow fields generated by movements of two-dimensional planes and three-dimensional objects. Under the assumption that the test region is a two-dimensional plane, a reference field from the actual optical flow field data is obtained. Then the degree of differences between the two fields can be used to distinguish between a three-dimensional face and a two-dimensional photograph.

The facial features movement analysis may be bypassed with video that is presented to the sensor instead of still photograph. The other methods assume that the presented fake sample is flat and does not contain any depth information. However, in recent years a market with 3D printers emerged and the fabrication of 3D face model is much easier than ever before. This is directly related to the 3D face recognition where the liveness detection is assumed implicitly. In order to investigate whether our 3D face recognition system is resistant to 3D fake samples we had manufactured realistic 3D face models – see Figure 6.20. The pure 3D face recognition system may be tampered with this model. However, the model has to be painted with similar optical properties as human skin. Otherwise it cannot be reconstructed with the 3D sensor properly.

Our experiments show that it is very hard to achieve the exact optical properties of the human skin. Thus it is also very hard to spoof a fake sample to the face biometric system utilizing both the shape as well as the texture information.

Chapter 7

Conclusion

This thesis presented a 3D face recognition approach based on the multi-algorithmic fusion of individual units utilizing iso-geodesic curves and specific image filters. The hill-climbing selection was used in order to combine only those units that have a positive impact on the recognition performance.

The idea of the multi-algorithmic approach has been published in [50, 55, 51]. These two papers and the book present the combination of anatomical soft-biometrics and holistic algorithms. It shows that the combination of multiple algorithms improves the recognition performance. We have utilized biometrics fusion in [56]. This paper describes the thermal face recognition pipeline where multiple subspace projection techniques are combined. The further extension of this approach has been presented in [83]. We have added the image filters prior to the subspace projections. The overview of the thermal face as well as 3D face recognition techniques was described in chapter “3D and Thermo-face Recognition” of the book “New Trends and Developments in Biometrics” [52].

The utilization of the hill-climbing unit selection was presented in [53]. This paper describes the basic idea of the iterative selection of those recognition units into a resulting multi-algorithmic system. A more robust selection has subsequently been presented in [54]. The main focus of this paper was targeting the 3D face recognition to low-cost depth sensors, such as Microsoft Kinect or SoftKinetic DepthSense DS325.

The presented recognition method requires user collaboration – the scanned subject has to be in a specific range from the sensor, look towards the camera and have a neutral face expression. All the mentioned factors (distance, dramatic facial expressions, and head rotation) can decrease the recognition performance although their impact can be reduced to some extent. The head rotation and distance from the sensor is easily compensable by the ICP registration. Facial expressions are solved implicitly - by selecting only the rigid parts of the face and selecting only recognition units robust to deformations caused by facial expressions.

The recognition algorithm was trained and evaluated on publicly available FRGC database. Moreover, we have conducted tests on our own databases acquired with Kinect and DepthSense DS325 sensors. Our results suggest that even the low-cost depth sensors that provide poor depth accuracy and noisy output can be used for successful identification in a relatively small database (up to 100 users). Our final experiments show that the main face recognition challenges - head orientation, facial mimics and varying lighting conditions may be solved. On the other hand, dioptric glasses pose difficulties. Recognition of persons wearing glasses and twins may be a promising direction for further research.

List of Abbreviations

DET – *Detection Error Trade-off* – graphical plot of error rates plotting *False Match Rates* (FMR) against *False Non-Match Rates* (FNMR). See Section 2.2.

DoG – *Difference of Gaussians* – feature enhancement filter that subtracts one blurred version of an original image from another, less blurred version of the original. See Section 5.4.3.

EER – *Equal Error Rate* is a value where the false rejection rate and false acceptance rate for a given decision threshold are equal. It is often used as criteria for evaluating performance of the biometric systems. See Section 2.2.

FRGC – *Face Recognition Grand Challenge* is a large dataset of three-dimensional face scans as well as high and low resolution photographs captured in controlled and uncontrolled lighting conditions. See Section 3.3.1

GMM – *Gaussian Mixture Models* are formed by combining multivariate normal density components. Gaussian mixture models are often used for data clustering.

ICP – *Iterative Closest Point* algorithm minimizes the difference between two clouds of points by transforming one to the other one. See Section 5.2.2.

LBP – *Local Binary Pattern* is a type of feature used for classification in the computer vision. See Sections 4.5 and 5.4.3.

LDA – *Linear Discriminant Analysis* is a subspace projection technique that seeks for vectors that provide the best discrimination between classes after the projection. See Section 4.4.2.

PCA – *Principal Component Analysis* is a subspace projection where the dimensionality reduction is based on the data distribution. See Section 4.4.1.

ICA – *Independent Component Analysis* is a subspace projection looking for the transformation of the input data that maximizes non-gaussianity. See Section 4.4.3.

SVM – *Support Vector Machine* is a binary classifier attempting to find a hyperplane that divides the two classes with the largest margin. See Section 5.7.2.

Bibliography

- [1] Y. Adini, Y. Moses, and S. Ullman. “Face Recognition: the Problem of Compensating for Changes in Illumination Direction”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19.7 (1997), pp. 721–732. ISSN: 0162-8828.
- [2] H. Ahmadi and A. Pousaberi. “An Efficient Iris Coding Based on Gauss-Laguerre Wavelets”. In: *Advances in Biometrics*. 2007, pp. 917–926. ISBN: 978-3-540-74548-8.
- [3] T. Ahonen, A. Hadid, and M. Pietikäinen. “Face Recognition With Local Binary Patterns”. In: *8th European Conference on Computer Vision*. 2004, pp. 469–481. ISBN: 978-3-540-21984-2.
- [4] *AT&T Database of Faces*. 1994. URL: <http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html> (visited on 02/05/2014).
- [5] W. Bao et al. “A liveness detection method for face recognition based on optical flow field”. English. In: *International Conference on Image Analysis and Signal Processing*. IEEE, 2009, pp. 233–236. ISBN: 978-1-4244-3987-4.
- [6] P. Belhumeur, J. Hespanha, and D. Kriegman. “Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19.7 (1997), pp. 711–720. ISSN: 0162-8828.
- [7] S. Berretti, A. Del Bimbo, and P. Pala. “3D Face Recognition Using iso-Geodesic Stripes”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.12 (Dec. 2010), pp. 2162–2177. ISSN: 1939-3539.
- [8] P. J. Besl and H. D. McKay. “A Method for Registration of 3-D Shapes”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14.2 (1992), pp. 239–256. ISSN: 0162-8828.
- [9] C. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2007. ISBN: 978-0-387-31073-2.
- [10] V. Blanz. “Face Recognition Based on a 3D Morphable Model”. In: *Proceedings of the 7th Int. Conference of Automatic Face and Gesture Recognition*. IEEE, 2006, pp. 617–624. ISBN: 0-7695-2503-2.
- [11] J. F. Cardoso. “Blind signal separation: statistical principles”. In: *Proceedings of the IEEE*. Vol. 86. 10. 1998, pp. 2009–2025.
- [12] S. Chakraborty and D. Das. “An Overview of Face Liveness Detection”. In: *International Journal on Information Theory (IJIT)* 3.2 (2014), pp. 11–25.
- [13] C. Chang and C. Lin. “LIBSVM: A Library for Support Vector Machines”. In: *ACM Transactions on Intelligent Systems and Technology* 2.3 (2011), p. 27. ISSN: 2157-6904.

- [14] A. Colombo, C. Cusano, and R. Schettini. “3D Face Detection Using Curvature Analysis”. In: *Pattern Recognition* 39.3 (Mar. 2006), pp. 444–455. ISSN: 0031-3203.
- [15] P. Comon. “Independent Component Analysis”. In: *Signal Processing - Special Issue on Higher Order Statistics* 36.3 (1994), pp. 287–314. ISSN: 0165-1684.
- [16] T. F. Cootes and C. J. Taylor. *Statistical Models of Appearance for Computer Vision*. Tech. rep. 2004, p. 125.
- [17] J. Daugman. “How Iris Recognition Works”. In: *IEEE Transactions on Circuits and Systems for Video Technology* 14.1 (Jan. 2004), pp. 21–30. ISSN: 1051-8215.
- [18] D. Falie and V. Buzuloiu. “Noise Characteristics of 3D Time-of-Flight Cameras”. In: *International Symposium on Signals, Circuits and Systems*. IEEE, July 2007, p. 4. ISBN: 1-4244-0968-3.
- [19] P. Fechteler, P. Eisert, and J. Rurainsky. “Fast and High Resolution 3D Face Scanning”. In: *IEEE International Conference on Image Processing (ICIP 2007)*. San Antonio, Texas, USA, 2007, pp. 81–84. ISBN: 978-1-4244-1437-6.
- [20] A. R. Fisher. “The Use of Multiple Measurements in Taxonomic Problems”. In: *Annals of Eugenics* 7.2 (1936), pp. 179–188.
- [21] S. B. Gokturk, H. Yalcin, and C. Bamji. “A Time-Of-Flight Depth Sensor - System Description, Issues and Solutions”. In: *Conference on Computer Vision and Pattern Recognition Workshop*. IEEE, 2004, pp. 35–44.
- [22] M. Haag and J. Romberg. *Eigenvectors and Eigenvalues*. Tech. rep. 2009.
- [23] Y. Hamamoto et al. “A Gabor Filter-Based Method for Recognizing Handwritten Numerals”. In: *Pattern Recognition* 31.4 (1998), pp. 395–400. ISSN: 0031-3203.
- [24] G. Hermosilla et al. “A Comparative Study of Thermal Face Recognition Methods in Unconstrained Environments”. In: *Pattern Recognition* 45.7 (July 2012), pp. 2445–2459. ISSN: 0031-3203.
- [25] T. Heseltine. “Face Recognition: Two-Dimensional and Three-Dimensional Techniques”. PhD thesis. The University of York, 2005.
- [26] L. Hong, Y. Wan, and A. Jain. “Fingerprint Image Enhancement: Algorithm and Performance Evaluation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20.8 (1998), pp. 777–789. ISSN: 0162-8828.
- [27] A. Hyvärinen. “Fast and Robust Fixed-Point Algorithms for Independent Component Analysis”. In: *IEEE Transactions on Neural Networks* 10.3 (1999), pp. 626–34. ISSN: 1045-9227.
- [28] A. Hyvärinen and E. Oja. “Independent Component Analysis: Algorithms and Applications”. In: *Neural Networks* 13 (2000), pp. 411–430. ISSN: 0893-6080.
- [29] *ISO/IEC JTC 1/SC 37, Harmonized Biometric Vocabulary*. Tech. rep. ISO/IEC, 2014.
- [30] S. Jahanbin, R. Jahanbin, and A. C. Bovik. “Passive Three Dimensional Face Recognition Using Iso-Geodesic Contours and Procrustes Analysis”. In: *International Journal of Computer Vision* 105.1 (June 2013), pp. 87–108. ISSN: 0920-5691.
- [31] S. Jahanbin et al. “Three Dimensional Face Recognition Using Iso-Geodesic and Iso-Depth Curves”. In: *2nd IEEE International Conference on Biometrics: Theory, Applications and Systems*. Ieee, 2008. ISBN: 978-1-4244-2729-1.

- [32] A. K. Jain, P. Flynn, and A. Ross. *Handbook of Biometrics*. Springer-Verlag New York, Inc., 2008, p. 556. ISBN: 978-1441943750.
- [33] A. K. Jain, A. Ross, and S. Prabhakar. “An Introduction to Biometric Recognition”. In: *IEEE Transactions on Circuits and Systems for Video Technology* 14.1 (2004), pp. 4–20. ISSN: 1051-8215.
- [34] H. K. Jee, S. U. Jung, and J. H. Yoo. “Liveness Detection for Embedded Face Recognition System”. In: *International Journal of Biomedical Sciences* 1.4 (2006), pp. 235–238. ISSN: 1555-2810.
- [35] D. G. Kendall. “A Survey of the Statistical Theory of Shape”. In: *Statistical Science* 4.2 (1989), pp. 87–99.
- [36] S. Kim et al. “Face Liveness Detection Using Variable Focusing”. English. In: *International Conference on Biometrics (ICB)*. IEEE, June 2013, pp. 1–6. ISBN: 978-1-4799-0310-8.
- [37] R Kohavi. “Wrappers for Feature Subset Selection”. In: *Artificial intelligence* 97.1 (1997), pp. 273–324. ISSN: 0004-3702.
- [38] G. A. Korn and T. M. Korn. *Mathematical Handbook for Scientists and Engineers: Definitions, Theorems, and Formulas for Reference and Review*. 2nd revise. Courier Corporation, 2000, p. 1130. ISBN: 0-486-41147-8.
- [39] L. I. Kuncheva et al. “Is Independence Good for Combining Classifiers?” In: *Proceedings of 15th International Conference on Pattern Recognition (ICPR-2000)*. Vol. 2. IEEE, 2000, pp. 168–171. ISBN: 0-7695-0750-6.
- [40] A. C. Lamont, S. Stewart-Williams, and J. Podd. “Face Recognition and Aging: Effects of Target Age and Memory Load”. In: *Memory & Cognition* 33.6 (2005), pp. 1017–1024. ISSN: 0090-502X.
- [41] D. T. Lee and B. J. Schachter. “Two Algorithms for Constructing a Delaunay Triangulation”. In: *International Journal of Computer & Information Sciences* 9.3 (1980), pp. 219–242. ISSN: 00917036.
- [42] F. Lenzen, H. Schäfer, and C. Garbe. “Denoising time-of-flight data with adaptive total variation”. In: *Lecture Notes in Computer Science*. Vol. 6938 LNCS. Springer Berlin Heidelberg, 2011, pp. 337–346. ISBN: 978-3-642-24027-0.
- [43] 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 Pattern Recognition*. 2004, pp. 362–366. ISBN: 0-7695-2128-2.
- [44] *Machine Readable Travel Documents, ICAO doc 9303*. Tech. rep. ICAO - International Civil Aviation Organization, 2006, p. 54.
- [45] M. H. Mahoor and M. Abdel-Mottaleb. “Face Recognition Based on 3D Ridge Images Obtained from Range Data”. In: *Pattern Recognition* 42.3 (2009), pp. 445–451. ISSN: 0031-3203.
- [46] I. Matthews and S. Baker. “Active Appearance Models Revisited”. In: *International Journal of Computer Vision* 60.2 (Nov. 2004), pp. 135–164. ISSN: 0920-5691.
- [47] A. Mian. “Illumination Invariant Recognition and 3D Reconstruction of Faces using Desktop Optics.” In: *Optics express* 19.8 (2011), pp. 7491–7506. ISSN: 1094-4087.

- [48] D. Modrow et al. “3D Face Scanning Systems Based on Invisible Infrared Coded Light”. In: *Advances in Visual Computing*. Springer Berlin Heidelberg, 2007, pp. 521–530. ISBN: 978-3-540-76857-9.
- [49] A. B. Moreno and A. Sanchez. “GavabDB: A 3D Face Database”. In: *Proceedings of 2nd COST Workshop on Biometrics on the Internet: Fundamentals, Advances and Applications*. 2004, pp. 77–82.
- [57] M. Muja and D. G. Lowe. “Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration”. In: *VISAPP International Conference on Computer Vision Theory and Applications*. 2009, pp. 331–340.
- [58] *Multi-Modal and Other Multi-Biometric Fusion*. Tech. rep. ISO/IEC, 2007.
- [59] K. Nandakumar et al. “Likelihood Ratio-Based Biometric Score Fusion.” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.2 (Feb. 2008), pp. 342–347. ISSN: 0162-8828.
- [60] T. Ojala, M. Pietikäinen, and T. Mäenpää. “Multiresolution Gray-scale and Rotation Invariant Texture Classification with Local Binary Patterns”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.7 (2002), pp. 971–987. ISSN: 0162-8828.
- [61] G. Pan et al. “3D Face Recognition Using Mapped Depth Images”. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05) - Workshops*. Vol. 3. IEEE, 2005, p. 175. ISBN: 0-7695-2372-2.
- [62] G. Pan et al. “Eyeblink-based Anti-Spoofing in Face Recognition from a Generic Webcamera”. English. In: *IEEE 11th International Conference on Computer Vision*. IEEE, 2007, pp. 1–8. ISBN: 978-1-4244-1630-1.
- [63] K. Pearson. “LIII. On lines and planes of closest fit to systems of points in space”. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11 (1901), pp. 559–572. ISSN: 1941-5982.
- [64] T. Peng. “Algorithms and Models for 3-D Shape Measurement Using Digital Fringe Projections”. PhD thesis. University of Maryland, 2006, p. 257.
- [65] P. J. Phillips et al. “FRVT 2006 and ICE 2006 Large-Scale Experimental Results”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.5 (May 2010), pp. 831–46. ISSN: 1939-3539.
- [66] P. J. Phillips et al. “Overview of the Face Recognition Grand Challenge”. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*. IEEE, 2005, pp. 947–954. ISBN: 0-7695-2372-2.
- [67] J. Pineda. “A parallel algorithm for polygon rasterization”. In: *ACM SIGGRAPH Computer Graphics* 22.4 (1988), pp. 17–20. ISSN: 00978930.
- [68] N. Poh. “User-specific Score Normalization and Fusion for Biometric Person Recognition”. In: *Advanced Topics in Biometrics, World Scientific Publisher* (2011), pp. 401–418.
- [69] N. Poh and S. Bengio. “How Do Correlation and Variance of Base-Experts Affect Fusion in Biometric Authentication Tasks?” In: *IEEE Transactions on Signal Processing* 53.11 (2005), pp. 4384–4396. ISSN: 1053-587X.

- [70] L. Puente et al. “Biometrical Fusion – Input Statistical Distribution”. In: *Advanced Biometric Technologies*. Ed. by G. Chetty and J. Yang. InTech, 2011, pp. 87–110. ISBN: 978-953-307-487-0.
- [71] A. Ross. “An Introduction to Multibiometrics”. In: *Handbook of Biometrics*. Ed. by A. K. Jain, P. Flynn, and A. A. Ross. Springer US, 2008, pp. 271–292. ISBN: 978-0-387-71040-2.
- [72] D. Ruppert. “Modeling Univariate Distributions - Robust Estimation”. In: *Statistics and Data Analysis for Financial Engineering*. 978-1-4419-7786-1, 2010, pp. 117–119. ISBN: 978-1-4419-7786-1.
- [73] J. C. Russ. “Image Enhancement in Spatial Domain”. In: *The Image Processing Handbook*. CRC Press, 2011, pp. 269–297. ISBN: 0-8493-7254-2.
- [74] R. B. Rusu. “Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments”. PhD thesis. Aug. 2009, p. 260.
- [75] J. Salvi, J. Pagès, and J. Batlle. “Pattern Codification Strategies in Structured Light Systems”. In: *Pattern Recognition 37.4* (2004), pp. 827–849. ISSN: 0031-3203.
- [76] M. Segundo et al. “Automatic 3D Facial Segmentation and Landmark Detection”. In: *Proceedings of the 14th International Conference on Image Analysis and Processing (ICIAP '07)*. 2007, pp. 431–436. ISBN: 0-7695-2877-5.
- [77] C. Shan, S. Gong, and P. W. McOwan. “Robust Facial Expression Recognition Using Local Binary Patterns”. In: *IEEE International Conference on Image Processing (ICIP 2005)*. 2005, pp. 370–373. ISBN: 0-7803-9134-9.
- [78] Y. Su, S. Shan, and X. Chen. “Hierarchical Ensemble of Global and Local Classifiers for Face Recognition”. In: *Image Processing, IEEE* 18.8 (2009), pp. 1885–1896. ISSN: 1057-7149.
- [79] X. Sun et al. “Fast and Effective Feature-Preserving Mesh Denoising”. In: *IEEE Transactions on Visualization and Computer Graphics* 13.5 (2007), pp. 925–938. ISSN: 1077-2626.
- [80] X. Tan and B. Triggs. “Enhanced Local Texture Feature Sets for Face Recognition Under Difficult Lighting Conditions”. In: *IEEE Transactions on Image Processing* 19.6 (2010), pp. 1635–1650. ISSN: 1941-0042.
- [81] H. Tang et al. “3D Face Recognition Using Local Binary Patterns”. In: *Signal Processing* 93.8 (2013), pp. 2190–2198. ISSN: 0165-1684.
- [82] M. A. Turk and A. P. Pentland. “Face recognition using eigenfaces”. In: *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 591.1 (1991), pp. 586–591. ISSN: 1063-6919.
- [84] C. Vielhauer, J. Dittmann, and S. Katzenbeisser. “Security and Privacy in Biometrics”. In: *Security and Privacy in Biometrics*. Ed. by P. Campisi. London: Springer London, 2013, pp. 25–43. ISBN: 978-1-4471-5229-3.
- [85] D. M. Weinstein. *The Analytic 3-D Transform for the Least-Squared Fit of Three Pairs of Corresponding Points*. Tech. rep. University of Utah, 1998, p. 10.
- [86] P. Yang et al. “Face Recognition Using Ada-Boosted Gabor Features”. In: *Proceeding of 6th IEEE International Conference on Automatic Face and Gesture Recognition*. 2004, pp. 356–361. ISBN: 0-7695-2122-3.

- [87] D. Zhang, J. Yang, and J. Y. Yang. “Is ICA Significantly Better than PCA for Face Recognition?” In: *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV’05)*. 2005, pp. 198–203. ISBN: 0-7695-2334-X.
- [88] W. Zhao, R. Chellappa, and P. J. Phillips. “Face Recognition: A Literature Survey”. In: *ACM Computing Surveys (CSUR)* 35.4 (2003), pp. 399–458. ISSN: 0360-0300.
- [89] X. Zhou, H. Seibert, and C. Busch. “A 3D Face Recognition Algorithm Using Histogram-Based Features”. In: *Eurographics 2008, Workshop on 3D Object Retrieval*. 2008, pp. 65–71. ISBN: 978-3-905674-05-7.

My publications

- [50] Š. Mráček. “3D Face Recognition”. In: *Proceedings of the 16th Conference and Competition STUDENT EEICT 2010*. Brno, CZ, 2010, pp. 124–126. ISBN: 978-80-214-4078-4.
- [51] Š. Mráček. *3D Face Recognition*. Saarbrücken: Lambert Academic Publishing, 2011. ISBN: 978-3-8465-4450-1.
- [52] Š. Mráček et al. “3D and Thermo-face Recognition”. In: *New Trends and Developments in Biometrics*. 2012, pp. 31–59. ISBN: 978-9-53510-859-7.
- [53] Š. Mráček et al. “3D Face Recognition Based on the Hierarchical Score-Level Fusion Classifiers”. In: *Biometric and Surveillance Technology for Human and Activity Identification XI*. 2014. ISBN: 978-1-62841-012-9.
- [54] Š. Mráček et al. “3D Face Recognition on Low-Cost Depth Sensors”. In: *Proceedings of the International Conference of Biometrics Special Interest Group (BIOSIG 2014)*. 2014. ISBN: 978-3-88579-624-4.
- [55] Š. Mráček et al. “Inspired by Bertillon - Recognition Based on Anatomical Features from 3D Face Scans”. In: *Proceedings of the 3rd International Workshop on Security and Communication Networks*. 2011, pp. 53–58. ISBN: 978-82-91313-67-2.
- [56] Š. Mráček et al. “Thermal Face Recognition - Fusion of Common Used Methods”. In: *Proceedings of the Emerging Security Technologies (EST 2012)*. 2012. ISBN: 978-0-7695-4791-6.
- [83] J. Váňa et al. “Applying Fusion in Thermal Face Recognition”. In: *International Conference of the Biometrics Special Interest Group (BIOSIG)*. 2012. ISBN: 978-3-88579-290-1.

Appendix A

Implementation

The **FaceLib** framework was developed as one of the inseparable parts of this thesis. The source code is available at <https://github.com/stepanmracek/face>. It contains the core library **libFaceCommon** as well as other support libraries and executables. It is developed in C++ language with the following dependencies:

- OpenCV – open-source computer vision library – <http://opencv.org/>
- POCO – open-source C++ class libraries and frameworks – <http://pocoproject.org/>
- CMake – cross-platform open-source software designed to build, test and package software. – <http://www.cmake.org/>
- OpenMP (optional dependency) – multi-platform shared memory multiprocessing programming in C and C++ <http://openmp.org/wp/>
- Qt (optional dependency) – cross-platform application framework. The application has been tested with Qt 4.8.6 and 5.3. – <https://qt-project.org/>
- Qwt (optional dependency) – Qt Widgets for Technical Applications – <http://qwt.sourceforge.net/>

The dependency graph of **FaceLib** framework components is illustrated in Figure A.1. The only mandatory components are **libFaceCommon** and **appAutoTrainer**. The remaining libraries and application require optional 3rd party libraries or sensor drivers.

libFaceCommon is the core component of the framework. It contains classes and methods covering linear algebra, machine learning, biometric data processing and 3D model processing. It contains the following namespaces:

Helpers namespace consists of support various support classes used across the entire project.

LinAlg contains classes and methods for linear algebra and machine learning.

FaceData is intended for processing of 3D models.

Biometrics namespace contains classes for modality-independent feature extraction, feature vector comparison, evaluation of biometric systems and multibiometrics.

ObjectDetection is focused on object detection in video and still images

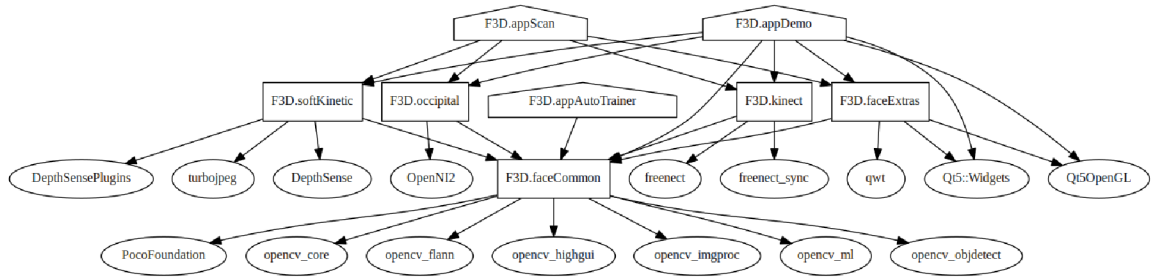


Figure A.1: Dependencies within the FaceLib framework.

libFaceExtras is optional component that uses Qt framework and Qwt library for displaying 3D face models and visualizing the performance of biometric systems.

libFaceSensors contains interface common to all sensors that may be used within the FaceLib framework.

Kinect is the implementation of **libFaceSensors** interface for Microsoft Kinect for XBox 360 sensor. It relies on **freenect** driver¹.

SoftKinetic is the implementation **libFaceSensors** interface for SoftKinetic DepthSense DS325 sensor².

Occipital contains partial support for Occipital Structure.IO sensor as well as other OpenNI2 compliant sensors³.

appAutoTrainer application is used for training of multi-algorithmic face recognition system involving score-level fusion. It can also be used reporting of biometric performance.

appScan is intended for creating of training 3D face database. See screenshot in Figure A.2.

appDemo is the demo application that demonstrates the identification as well as verification process. See screenshot in Figure A.3.

¹<http://openkinect.org/>

²<http://www.softkinetic.com/Support/Download>

³<https://github.com/OpenNI/OpenNI>

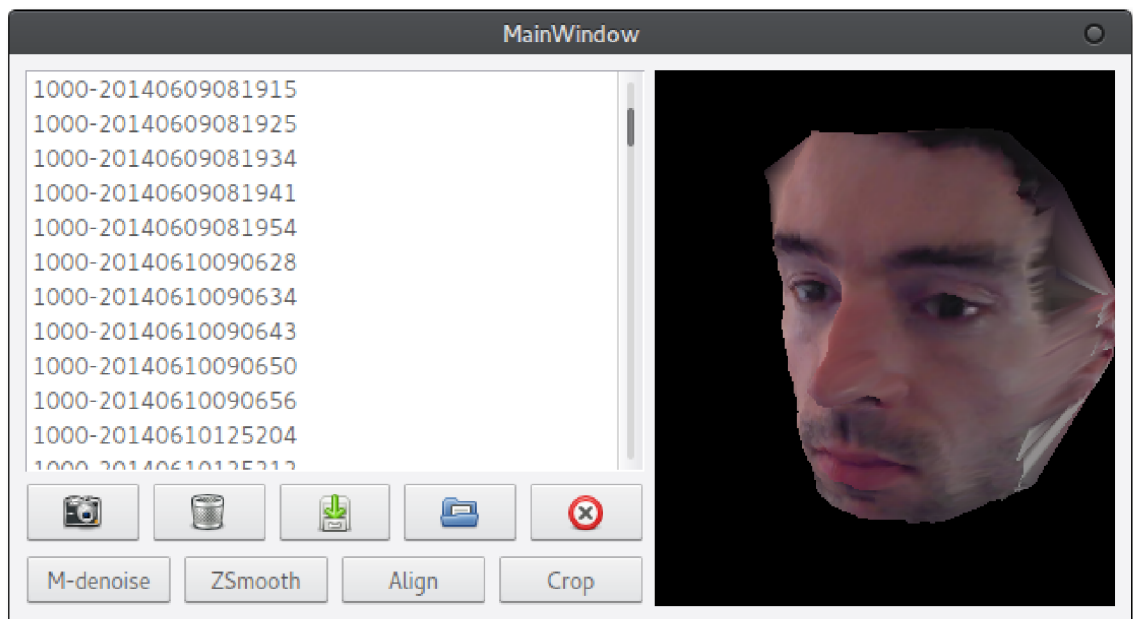


Figure A.2: Screenshot of appScan application.

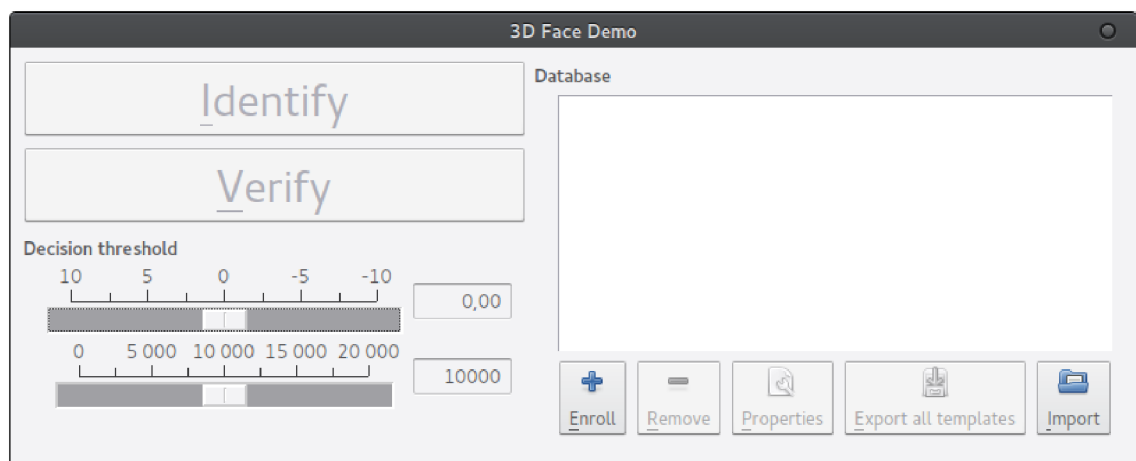


Figure A.3: Screenshot of appDemo application.