PhD Thesis

FINGERPRINT RIDGE ORIENTATION MODELING

Surinder Ram

Graz University of Technology
Institute for Computer Graphics and Vision

In cooperation with the
Biometrics Center of Siemens Austria
Siemens IT Solutions and Services

Supervisor
Univ.-Prof. Dipl.-Ing. Dr.techn. Horst Bischof

Second Advisor
Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Robert Sablatnig

TU Graz
Graz University of Technology
Senat

Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008
Genehmigung des Senates am 1.12.2008

Deutsche Fassung:

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Graz, am 17.6.2009 ……………………………………………..
(Unterschrift)

Englische Fassung:

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

17.6.2009
.... ……………………………………………..
date (signature)

Contents

Abstract

In the last decade, automatic fingerprint based personal authentication has matured to the point where it can be successfully applied in a myriad of applications. These applications are ranging from the low cost door opener for private use up to the critical government application ensuring national security.

One common ground of all this fingerprint based personal authentication systems is the necessity for estimation of fingerprint ridge orientation. The importance of ridge orientation can be deflected from the fact that it is inevitably used for detecting, describing and matching fingerprint features such as minutiae and singular points. Using the ridge orientation, not only the error rates can be improved, but also more efficient image compression and a speed up in database queries can be achieved. This is the main motivation of this thesis and of many publications available in literature for modelling fingerprint ridge orientation.

In this thesis we analyse current available techniques and propose a novel method for fingerprint ridge orientation modelling. One of the main problems it addresses is smoothing orientation data while preserving details in high curvature areas, especially around singular points. We show that singular points, which result in a discontinuous orientation field, can be modelled by the zero-poles of orthogonal polynomials. The models parameters are obtained in a fast two staged optimization procedure.

Another contribution of this thesis is the application of a priori knowledge in fingerprint orientation models. Starting from the view point of flexible templates models, we develop a method which constraints the fingerprint orientation to vary only in ways as they occur in nature.

Extensive experiments using a commercial state-of-the-art fingerprint matcher, have been carried out. We can report statistically significant improvements in both, singular point detection and matching rates.

Zusammenfassung

Im letzten Jahrzehnt konnte die Leistungsfähigkeit von automatischen Fingerabdruck-Erkennungssystemen erheblich verbessert werden. Mittlerweile kommen diese biometrischen Erkennungssysteme immer häufiger im öffentlichen als auch im privaten Bereich zum Einsatz.

Eine Gemeinsamkeit aller Fingerabdruck-Erkennungssysteme ist die Notwendigkeit zur Extraktion der Richtungen von papillaren Linien eines Fingerabdruckes. Dies resultiert aus der Tatsache das diese Orientierungsinformation für die Detektion, Beschreibung und für das Verifizieren von Fingerabdrücken verwendet wird. Weiters können diese Orientierungfelder für die effiziente Kompression von Fingerabdruck-Bildern als auch zum Beschleunigen von vorhandenen Algorithmen verwendet werden. Fehlerraten eines Fingerabdruck-Erkennungssystemen hängen damit direkt mit der Qualität des Orientierungfeldes zusammen. Dies ist die Hauptmotivation für die Forschung an einem modellbasierten Verfahren zur Bestimmung des Orientierungsfeldes.

In der vorliegenden Arbeit werden vorhandene Methoden der Literatur zur Modellierung von Orientierungsfeldern von Fingerabdrücken analysiert. Es werden neue, verbesserte Methoden erarbeitet, die eine erhebliche Verbesserung im Vergleich zu vorhandenen Methoden darstellen. Ein zentrales Problem bei der Bestimmung von Orientierungsfeldern besteht darin den Einfluss von Störungen möglichst zu reduzieren aber gleichzeitig die singulären Punkte zu erhalten.

Eine weitere Kontribution der vorliegenden Dissertation ist die Anwendung von a priori Wissen für die Extaktion von Orientierungsfeldern. Hier wird ausgehend von sogenannten 'flexible template models' systematisch eine Methode entwickelt die Orientierungsfelder auf biologisch plausible Varianten begrenzt.

Im experimentellen Teil werden die Vorteile des in dieser Arbeit beschriebenen Modells mittels kommerziell erhältliche Fingerprint-Software nachgewiesen. Es konnte einerseits die Fehlerrate verringert und andrerseits die Zuverlässigkeit der Detektion von Singularitäten erhöht werden.

All models are wrong, but some are useful.

George Edward Pelham Box

Acknowledgment

This thesis is the result of more than three years of work during which I have been accompanied and supported by many people. It is my great pleasure to take this opportunity to thank them.

First and foremost I would like to thank my supervisor Professor Horst Bischof for giving me the opportunity to carry out this research. I want to thank him for his constant encouragement, guidance and help. He gave me the freedom to conduct the research in my own way, while arranging all the necessary conditions for doing good research.

I owe my deepest gratitude to Dipl.-Ing. Josef Birchbauer at Siemens for his many thoughtful and thought-provoking comments on this work - regularly putting its applicability and usefulness to scrutiny, and thus ensuring its practical worth.

I would also like to thank the whole Institute for providing a pleasant and inspiring atmosphere. On a personal note, I would like to thank Manfred Klopschitz, Arnold Irschara, Georg Pacher and Stefan Kluckner for all the fun and encouraging discussions.

I wish to thank my family for their understanding and support during these years that I have pursued my doctorate. To my wife and our children, your patience, love and encouragement have upheld me, particularly in those many days in which I spent more time with my computer than with you.

Graz, December 2009
Surinder Ram

Chapter 1

Introduction

Contents

Abstract

The primary goal of the first chapter is to give a general introduction on biometrics with focus on fingerprint verification. It is explained how a fingerprint verification system extracts distinctive features and uses these features for 'matching' two fingerprints. One of the main modules of a fingerprint verification system, namely fingerprint ridge orientation estimation, is described in detail. We will describe the role of this module and discuss the motivations behind fingerprint ridge orientation modelling. These motivations are mainly concerned with lower error rates, higher compression ratios and lower processing times. In the last part of this chapter an outline of the thesis is given.

1.1 Biometric systems

Humans have experience in recognizing a familiar person based on his/her specific characteristics, like voice, face, gait etc. Nowadays, there is an increasing need for reliable personal identification by automatic means. This has resulted in the establishment of a new research and technology area known as biometric recognition or simply 'biometrics' [47]. In this case the term 'biometrics' refers to automatic recognition of an individual based on behavioural and/or anatomical characteristics (e.g.: fingerprints, face, iris, voice, signature, etc.).

Most of the basic knowledge associated with person identification was already established by forensic scientists in the early 20^{th} century. For example Alphonse Bertillon developed the anthropometric identification approach, based on the measure of physical characteristics of a person [37]. Galton [27] identified the characteristics by which fingerprints can be identified. These characteristics (called minutia) are basically still in use today, and are also referred to as Galton's Details.

Today, we are seeing an increasing deployment of biometric systems in many aspects of life, including face, fingerprint, gait and iris recognition systems at airports (examples are given in Figure 1.1 and 1.2) as well as access to highly secure facilities [19]. Biometric systems offer great convenience and several advantages over traditional security systems based on passwords (can be forgotten, shared, copied) or keys (can also be stolen, copied or lost). Without sophisticated means, biometrics are difficult to share, steal or forge and cannot be forgotten or lost. Therefore, biometrics provide a higher security level in identity prove. Additionally, the combination of possession (key) and knowledge (pin) with biometrics makes the identity proof even more secure.

A significant difference between a biometrics based person identification and conventional methods is that the latter involves a complex pattern recognition method and hence not always performs accurately as intended by their system designers. A password based authentication method provides a clear answer if a person is who she/he claims to be, whereas a biometric system usually provides a similarity score.

Biometric systems are not perfectly accurate and basically commit two types of errors. In the first case the system may identify an impostor to be a genuine user. In the second case the system rejects a genuine user as an impostor. Whereas the false reject leads to inconvenience for users, a false acceptance provides the access to a non authorised user.

Other issues are related to biometric systems being a privacy concern. Usually, a part of a personal information is stored in a database. In contrast to passwords, which are usually stored encoded by a hash function, securing biometric templates represents a very difficult task. This is due to the reason that it is difficult to find a hash function being error tolerant enough for biometric data and at the same time being non invertible. Biometrics based security systems are generally classified into verification systems and identification systems. Verification systems usually output a decision about whether a query biometric matches the holding template on a one-to-one basis, whereas identification suggest whether or not the query can find a match in the database, which is instead a one-to-many matching process.

Any human physiological or behavioural characteristic can be used as a biometric identifier to recognize a person as long as it satisfies the following requirements [44]:

- **universality**: each person should have the perspective biometric.
- **distinctiveness**: the biometrics of any two persons should be sufficiently different.
- **permanence**: biometrics should be sufficiently invariant over a period of time.
- **collectability**: the biometrics can be measured quantitatively.

Additionally, there are a number of practical issues which should be considered:

- **performance**: refers to achievable recognition accuracy, speed, robustness, the resource requirements to achieve the desired recognition accuracy and speed, as well as operational or environmental factors that affect the recognition accuracy and speed.
- **acceptability**: which indicates the extent to which people are willing to accept a particular biometric identifier in their daily lives.

- **circumvention**: refers to how easily the system can be fooled by fraudulent methods.

A practical biometric system should have acceptable recognition accuracy and speed with reasonable resource requirements, should be harmless to the user, accepted by the user, accepted by the intended population, and sufficiently robust to various fraudulent methods.

Figure 1.1: An official taking fingerprints of a passenger arriving at Dallas' airport.

1.2 Fingerprint Features: global and local

The attractiveness of fingerprints results from their uniqueness which does not change through the life of individuals [72]. Characteristic fingerprint features can be categorized [44,43] into three levels (visual explanation given in Figure 1.3).

Figure 1.2: Regular guests scanning their fingerprints during a recent trial of a fingerprint identification systems at the Metropolis Fremantle nighclub in Perth, WA, Western Australia.

- **Level 1** features are the macro details of the fingerprint. A fingerprint forms a pattern of ridges and valleys on the surface of a fingertip. Ridges and valleys form an almost smooth pattern. On closer examination singular points (SPs) can be found where the orientation is discontinues. Though not unique, these level 1 features play an important role in automatic fingerprint matching algorithms. Typically, level 1 features are used to classify a fingerprint into one of several classes. Level 1 features have been used in fingerprint analysis for a long time. In 1892 Galton [27] already used them to classify fingerprints into three different classes (loops, whorls and arches). A more modern terminology, usually called the Henry-Galton classification schema (shown in Figure 1.4), uses the following six classes: an arch is a fingerprint featuring no SP at all, loops and tented archs show one core and one delta, whorls and twin loops have two deltas and one or two cores. Classification reduces the amount of data to be searched for matches as the database can be partitioned into subsets. Especially in large scale applications this results in a vital speed up. Furthermore, the matching process can be made more robust by using SPs. Usually this is done by aligning a pair of fingerprints

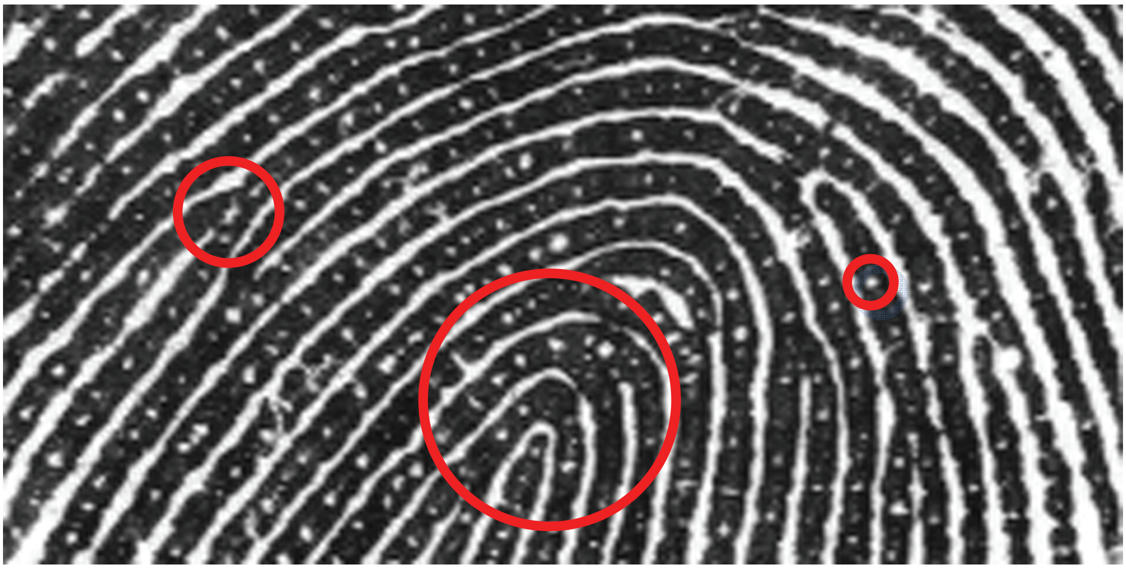
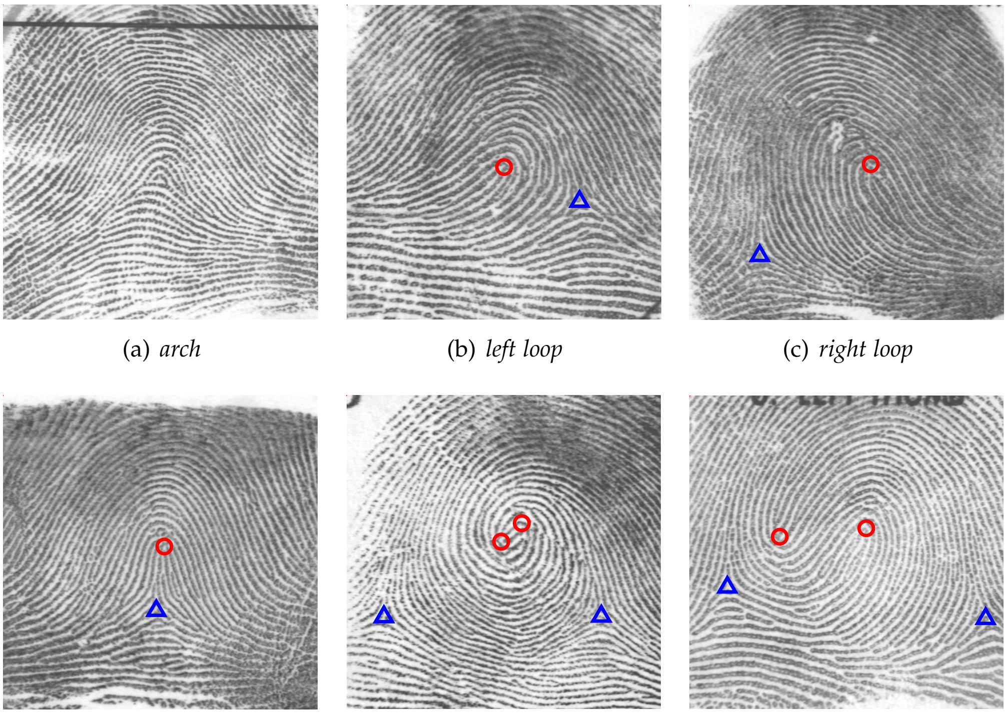


Figure 1.3: Fingerprint features of different levels. From left to right: minutiae, singular point and a pore.

before matching. All SP detection algorithms make use of the ridge orientation, which as described below, can be very difficult to extract in certain regions.

- **Level 2** features represent the back end of automatic fingerprint authentication systems. The so-called minutiae arise at positions where ridges end or bifurcate. An overview of all three features is given in Figure 1.3. Theoretical considerations [72] show that these level 2 features have enough discrimination power to guarantee the individuality of a fingerprint over (at least) billions of samples. Practical analyses [45, 15, 62, 20] show that poor image quality severely affects the recognition performance - especially in large scale applications.

- **Level 3** features are on every ridge of the finger epidermis - in form of many tiny sweat pores and other permanent details (e.g. scares). Pores are considered to be highly distinctive in terms of their number, position, and shape. However, extracting pores is feasible only in high-resolution fingerprint images (for example 1000 DPI) and in very high quality images. Therefore, this kind of representation is only rarely adopted by current systems. A recent example where level 3 features are applied for forensic purpose is given in [43].



(a) *arch*

(b) *left loop*

(c) *right loop*

(d) *tented arch*

(e) *whorl*

(f) *twin loop*

Figure 1.4: **Henry-Galton fingerprint classification scheme:** arch, left loop, right loop (top). tented arch, whorl, twin loop (bottom). Core type singular points are marked with ’○’ and delta type singular points are marked with a ’△’.

1.3 Fingerprint Recognition

As mentioned before, there are different type of biometric identifiers in use in various applications. Among them are ear, face, facial thermogram, hand thermogram, hand vein, hand geometry, fingerprint, iris, retina, signature, footsteps, gait, keystroke dynamics, voice, odor and DNA.

Comparing different biometrics will show that fingerprints are one of the most attractive. In [44] several biometric identifiers are compared (see Table 1.1). This comparison shows that between many biometric modalities fingerprints have a good balance of all the desirable properties mentioned above. Every human possesses fingerprints, with the exception of any hand-related disabilities. Fingerprints are very distinctive, their details are permanent, even if they temporarily change slightly due to cuts and bruises on the skin. Fingerprint scanner can capture high-quality images and tasks like the segmentation into foreground and background are comparably basic problems. The

Biometric Identifier	Universality	Distinctiveness	Permanence	Collectability	Performance	Acceptability	Circumvention
FACE	H	L	M	H	L	H	H
Fingerprint	M	H	H	M	H	M	M
Hand geometry	M	M	M	H	M	M	M
Hand/finger vein	M	M	M	M	M	M	L
Iris	H	H	H	M	H	L	L
Signature	L	L	L	H	L	H	H
Voice	M	L	L	M	L	H	H

Table 1.1: Comparison of common biometric traits. Entries in the table as given in the 2009 Edition of the Handbook of Fingerprint Recognition [44]. High, Medium and Low are denoted by H, M and L, respectively. Note that a reason for the popularity of fingerprint recognition is the very good balance between the given properties.

deployed fingerprint-based biometric systems offer good performance and fingerprint sensors have become very small and affordable. On the downside, fingerprints have the stigma of criminality associated with them because they have a long history of use in forensics. However, this is changing with the high demand of automatic recognition to fight identity fraud in in many applications. Fingerprint recognition has become one of the most mature biometric technologies and is therefore suitable for a large number of applications. See Table 1.1 for a compact comparison of different biometric modalities.

1.3.1 Fingerprint Sensing

Fingerprint images may be classified as off-line or live-scan. Off-line images are typically obtained by smearing ink on the fingertip and creating an inked impression of the fingertip on paper. The inked impression is then digitized by scanning the paper using an optical scanner. In forensic application, there exist so called latent fingerprints found at crime scenes. The latent fingerprints result from the oily nature of the human skin - impressions of a fingerprint are deposited on a surface that is touched by a finger. These impressions can be lifted from the surface by employing certain chemical techniques [44]. Historically, fingerprint databases were acquired by means of the off-line method.

A live-scan image on the other hand, is acquired by sensing the tip of the finger directly, using a sensor that is capable of digitizing the fingerprint on contact. A fingerprint sensor is an electronic device used to capture a digital image of the fingerprint pattern. The most important part of fingerprint scanner is the sensor, which is the component where the fingerprint image is formed. In the following an overview of some of the most commonly used fingerprint sensor technologies is given [44, ?].

- **Optical:** optical fingerprint imaging involves capturing a digital image of the print using visible light. This type of sensor is, in essence, a specialized digital camera (using either a CCD or CMOS sensor). The finger touches the top side of a glass, while the ridges enter in contact with the prism surface, the valleys remain at a certain distance. A lack of reflection allows the ridges to be discriminated from the valleys. A scratched or dirty touch surface can cause a bad image of the fingerprint.

- **Ultrasonic:** ultrasound sensing may be viewed as a kind of echography, based on sending acoustic signals toward the fingertip and capturing the 'echo signal'. This echo signal can be used to compute a fingerprint image. The sound waves are generated using piezoelectric transducers and the reflected energy is also measured using piezoelectric materials. An clear advantage of this method is that it eliminates the need for clean, undamaged epidermal skin and a clean sensing surface. However, scanner of this type are large with many mechanical parts and too expensive for many applications. Moreover, it may take several seconds to acquire an image, although some scanner of the latest generation are as fast as optical scanners.

- **Capacitive:** capacitive sensors utilize the principles associated with capacitance in order to form a fingerprint image. Each sensor array pixel acts as one plate of a small parallel-plate capacitor. The dermal layer of the fingerprint, which is electrically conductive, acts as the other plate while the non-conductive epidermal layer acts as a dielectric.

- **Thermal:** These sensors are made of pyroelectric material that generates current based on temperature differentials. The fingerprint ridges, being in contact with the sensor surface, produce a different temperature differential than the valleys, which are further away from the sensor surface. To increase the temperature difference, thermal sensors are heated up electrically. The temperature difference produces an image when it occurs, but this image soon disappears because the thermal equilibrium is quickly reached. Therefore, a common implementation of this sensor technology is a sweep sensor. A sweep sensor enjoys several advantages, including cost, sensitivity to ESD and sensor size. A general disadvantage of these sensors is the necessity to reliably reconstruct one image out of several slices. Furthermore, novice users have difficulties in performing the sweeping properly.

- **Piezoelectric:** This Sensor consists of pressure sensitive elements that have been designed to produce an electrical signal when a mechanical stress is applied to them (piezoelectric effect). Because ridges and valleys are present at different distances from the sensor surface, they result in different pressure and thus different amounts of generated current. A disadvantage of this type of sensor is the limited sensitivity of the used materials which leads to rather blurry images.

A new generation of touchless live scan sensors that generate a 3D representation of fingerprints is appearing [56]. Such a sensor acquires several images from different views and constructs a 3D representation for a fingerprint. This sensing technology overcomes some of the problems that intrinsically appear in contact-based sensors (e.g. improper finger placement, skin deformation, sensor dirt). On the other hand, some new challenges emerge, for example low ridge-valley contrast and 3D to 2D image mapping.

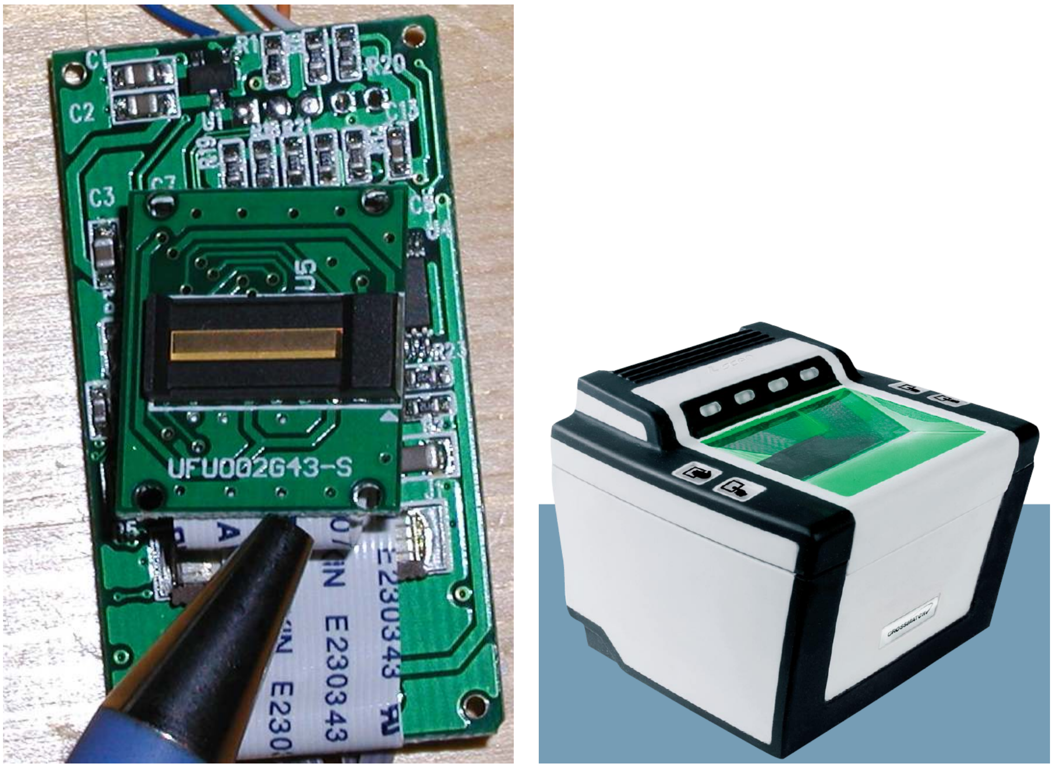
1.3.2 Preprocessing and Feature Extraction

For ’matching’ (comparing) two fingerprints most available methods use distinctive features of the fingerprint. There exist correlation-based methods for fingerprint matching, who directly compare the images (without the need for feature extraction) but usually their performance is poor, mainly because gray scale intensities are known to be unstable. As described in the next section, there exist three different levels of features in fingerprint images: pores, minutiae and flow patterns. Figure 1.6 and 1.7 shows the extraction steps using a sample fingerprint.

To make feature extraction easy and robust, the following pre-processing step are commonly performed:

- **Estimation of local ridge orientation.** The local ridge orientation at a pixel level is defined as the angle that the fingerprint ridges form with the horizontal axis. Usually the local ridge orientation is computed block wise. The simplest approach for local ridge orientation estimation is based on gray-scale gradient. Since this feature extraction step is in the main focus of this thesis, we give thorough details on existing orientation estimation techniques in the next chapter.

- **Computing local ridge frequency.** The local ridge frequency at pixel level can be defined as the number of ridges per a given length along a hypothetical segment centered at this pixel and orthogonal to the local orientation [44]. Similar to



(a) *Finger835K from Sunplus*

(b) *Guardian L SCAN from Crossmatch Technologies*

Figure 1.5: **Commercial fingerprint-scanners.** Figure (a) shows a lowest cost thermal sweep sensor from Sunplus. For an fast and easy assembly this sensor is already equipped with an USB interface. Such sensors can be obtained for only a few Euros. The second sensor (shown in Figure (b)) is a high quality optical ten-print capture scanner as used by the U.S. Bordercontrol. The price of this scanner is in the range of several thousands of Euros.

the local orientation, the local frequency is computed block wise. Many existing methods model the ridge-valley structure as a sinusoidal-shaped wave, where the ridge frequency is estimated as the frequency of this sinusoid. For a more detailed overview see [10] and references therein.

- **Enhancement of the captured fingerprint image.** In fingerprint images, ridges and valleys flow smoothly in a locally constant direction. However, in practice there are many factors that can affect the image quality of a fingerprint.
  - wetness or dryness of the skin
  - noise from the sensor

  - cuts and bruises on the skin

  An example is given in Figure 1.6. This topic is discussed in detail in Chapter 5.

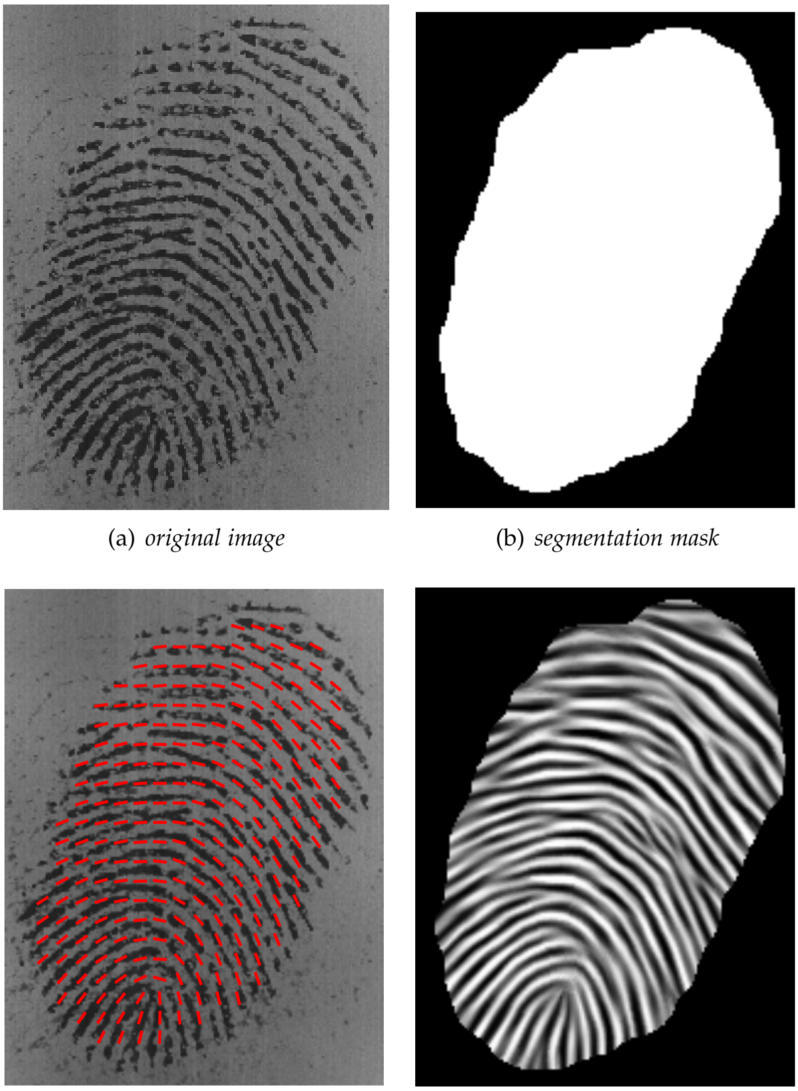
- **Segmentation of fingerprints into foreground and background.** Fingerprint segmentation algorithms are used to separate the fingerprint area (foreground) from the background. This is a useful step to avoid spurious extraction of fingerprint features in the background area. Most of the existing methods exploit the fact that fingerprint images contain oriented patterns while the background area is a non-oriented isotropic pattern. See Figure 1.6 for an visual example.

1.3.3 Matching

A matching step is necessary for the ’comparison’ of two given fingerprints and reporting the degree of similarity (typically reported as a match score). As reported above, reliable fingerprint matching methods are based on minutiae. Existing approaches can be classified into three families [44]:

- **correlation-based matching:** two fingerprint images are superimposed and the correlation between corresponding pixels is computed for different alignments. Due to non-linear distortions, different impressions of the same finger may result in differences of the global structure which finally results in unreliable comparisons. For a complete high resolution fingerprint image such a correlation based method results in slow algorithms. Therefore, this type of algorithm is only used to compare local fingerprint texture (e.g. around minutiae).

- **minutiae-based matching:** this is the most popular and widely used technique and it is based on finding the alignment between two minutiae sets, resulting in the maximum number of minutiae pairings. Minutiae based matching essentially consists of finding the alignment between the template and the input minutiae sets that results in the maximum number of minutiae pairings. This type of algorithm usually starts with the thinning (skeletonization) of a binarized fingerprint image. This thinning step reduces the ridge thickness to one pixel, allowing straightforward minutiae detection. See Figure 1.7 for an visual example. During the thinning step a number of spurious imperfections may appear. Usually, this wrong detections must be handled in a second post processing step. Additionally, binarization and thinning may suffers from several other problems (loss of structural information, computational cost, lack of robustness for lower quality images).



(a) *original image*

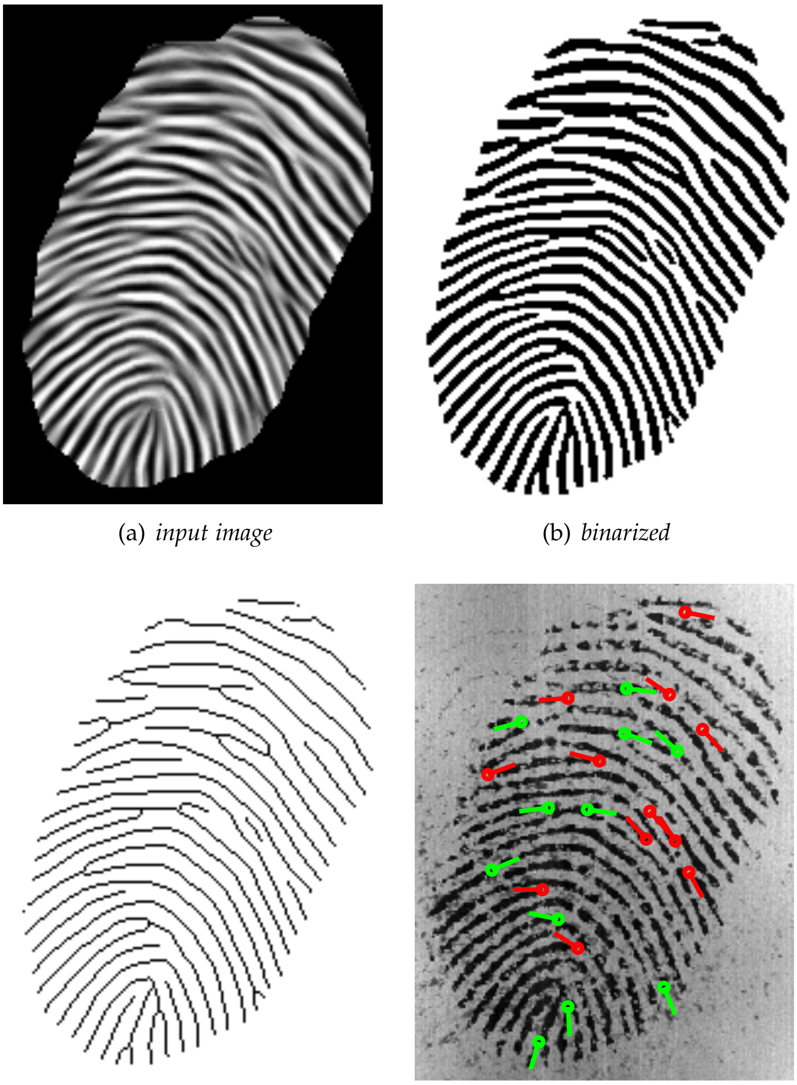
(b) *segmentation mask*

(c) *orientation field (OF)*

(d) *enhanced*

Figure 1.6: This figure shows the segmentation and the extracted OF of a example fingerprint. Subfigure (d) shows the fingerprint after enhancement.

- **ridge feature-based matching:** the approaches belonging to this family compare fingerprints in terms of features extracted from ridge patterns such as local orientation and frequency, ridge shape or texture information. Ridge feature information is less discriminative than minutiae, but more reliable under low quality conditions [21].



(a) *input image*

(b) *binarized*

(c) *skeletonized*

(d) *detected minutiae*

Figure 1.7: **Minutiae detection.** After the binarization of the enhanced (input) image a thinning step is performed. Minutiae are detected directly using the skeletonized image.

- **pore based:** Sweat pores have been used for a long time by forensic experts and have been proven to be very distinctive features. Recently [43, 101], sweat pores have been applied as features for personal identification successfully. However, it is not possible to extract pores from low resolution images - it is necessary to use high resolution scanners with more than 1000 dpi. Furthermore, pore extraction depends even more heavily on clean fingerprint images than the minutiae extraction.

Many available matcher use a combination of the above mentioned approaches. Note that many other techniques described in literature can principally be associated with one of the above mentioned methods.

1.3.4 Classification

In order to identify a person, his/her fingerprint have to be compared with each fingerprint in a database. In case of a large database, the identification typically has an unacceptably long response time. A common strategy to speed up the query is to narrow the search by dividing the fingerprint database into a number of bins (based on some predefined classes). A given fingerprint to be identified is then compared only to the fingerprints in a single bin. The most important and widely used classification schemes are variants of Henry's classification scheme. Most popular is the Henry-Galton classification schema which classifies fingerprints into six classes (see Figure [36]).

Due to its importance, classification has attracted a significant amount of interest. In particular, almost all the methods are based on one or more of the following features: orientation field, singular points and Gabor filter responses. Note that the latter two features can be only computed using the OF. Thus, classification heavily depends on correctly estimated fingerprint orientations. Existent methods can be coarsely assigned to the following categories [44]:

- **rule-based:** A fingerprint is simply classified according to the number and the position of the singularities. This approach is commonly used by human experts for manual classification, therefore several authors proposed to adapt the same technique for automatic classification. In [50] Kawagoe and Tojo proposed to use the Poincáre index to find the SP. Due to it's simplicity and more than adequate performance in most images, this singularity based method enjoys high popularity in fingerprint recognition systems. However, there are principal weaknesses adhered to the method. Many rules and heuristics have been proposed by different authors (e.g. [92]) in order to make the method robust against noise and minor occlusions.

- **syntactic:** Syntactic methods describe patterns by means of terminal symbols and production rules. Therefore, a grammar is defined for each class and parsing process is responsible for classifying each new pattern. Generally, syntactic approaches require very complex grammars whose inference implies complicated and unstable approaches. Therefore, the use of syntactic methods for fingerprint classification has been almost abandoned.

- **structural:** Structural approaches are based on the relational organization of low-level features into higher-level structures. This relational organization is represented by means of symbolical data structures (i.e. trees or graphs) which allow hierarchical organization of the information. Usually, the orientation image is used for structural representation. An example for this type of method is given in [61] which exploits the fact that partioning the orientation image in regions characterizedby homogeneous orientations implicitly reveals the position of singularities. The borderline between two adjacent regions is called a fault-line. By noting that fault lines converges towards loop singularities and diverge from deltas, the authors define a geometrical method for determining the convergence and divergence points. The main advantage of this approach is that it is able to deal with partial fingerprints, where sometimes SPs are not available. A disadvantage of the method is the dependency on robust segmentation of the orientation field into homogeneous regions. This is a difficult problem in noisy images.

- **statistical:** Statistical approaches use a fixed size numerical feature vector derived from each fingerprint and a general purpose statistical classifier. A simple example for such a classifier is the k-nearest neighbour [6]. Many existing approaches directly use the OF as a feature vector, by simply nesting its rows. Since the training of the resulting high dimensional feature vector (minimum 900 dimensions) requires large amounts of memory and time, a principal component analysis is applied first.

1.4 System Error Rates

Due to the variations present on each instance of a fingerprint capture, no recognition system can give an absolute answer about the individual's identity. Instead it provides the individual's identity information with a certain confidence level based on a similarity score. This is different from traditional authentication systems (e.g. as passwords) where the match is exact and an absolute 'yes' or 'no' answer is returned. The validation procedure in such cases is based on whether the user can prove the exclusive possessions (cards, keys, ...) or the secret knowledge (password or PIN number). The biometric signal variations of a persons finger are usually referred to as intraclass variations - whereas variations between different persons are referred to as interclass variations.

A fingerprint matcher takes two fingerprints, F_1 and F_2, and produces a similarity measurement $S(F_1, F_2)$, which is usually normalized in a interval, eg. [0, 1]. If $S(F_1, F_2)$ is close to 1, the matcher has greater confidence that both fingerprints come from the same individual.

In the commonly used terminology, the identity of a queried fingerprint is either a genuine type or an imposter type. Therefore resulting in two statistical distributions of similarity scores - usually called genuine distribution and imposter distribution. Using a matcher, the input fingerprints are classified into 'match' or 'non-match'. This can result into the following four scenarios:

1. a genuine individual is accepted
2. a genuine individual is rejected
3. an imposter individual is accepted
4. an imposter individual is rejected

The ideal fingerprint authentication system should produce only the first and fourth outcomes. Due to low image quality and intraclass variations (e. g. different overlaps) of the fingerprint images, and the limitations of a real fingerprint verification system, a genuine individual could be mistakenly recognized as an imposter and vice versa. The first scenario is referred to as 'false reject' and the corresponding error rate is called the False Reject Rate (FRR). The second scenario, an imposter individual mistakenly recognized as genuine, is referred to as 'false accept' and the corresponding error rate is called the False Accept Rate (FAR). FAR and FRR are used as a measurements in evaluation of fingerprint systems [15, 62, 20]. Commonly, the distributions of the similarity score of genuine attempts and imposter attempts cannot be separated completely by a single carefully chosen threshold. This results in a trade off between FAR and FRR, which must be carefully selected depending on the application. FRR and FAR are functions of a given threshold t and can be computed as following:

$$\mathrm{FRR}(t) = \int_0^t p_i(x)dx \tag{1.1}$$

$$\mathrm{FAR}(t) = \int_t^{\mathrm{inf}} p_g(x)dx \tag{1.2}$$

where $p_i(x)$ and $p_g(x)$ are the imposter and genuine distributions, respectively.

When t decreases, the system would have more tolerance to intraclass variations and noise, however the FAR will increase. Similarly, if t is higher, the system would be more secure and the FRR will increase. Depending on the nature of the application, the biometric system can operate at low FAR configurations (e.g. military use), which requires high security, or on low FRR where easy access is required (i.e. amusement

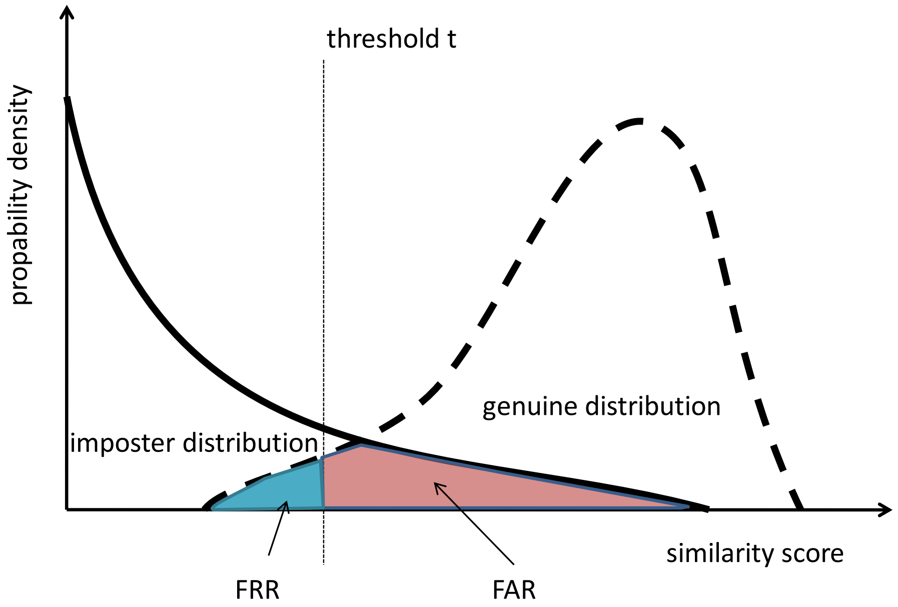


Figure 1.8: Typical distribution of genuine and imposter match scores. Note how FRR and FAR are functions of a given threshold t.

parks). For reporting the quality of fingerprint system, a common technique is to use a Receiver Operating Characteristic (ROC) curve which can be obtained by plotting FAR versus 1-FRR at all thresholds. The threshold t of the related authentication system can be carefully selected to meet the requirement of the given application. An example illustrating the above facts is given in Figure 1.9.

Additionally, the following system error rates are reported:

- Equal Error Rate (EER): the error rate where FAR equals FRR. In practice, the operating point corresponding to EER is rarely adopted in the fingerprint recognition system, and the threshold t is tailored to the security needs of the application.

- Zero FNMR: the lowest FAR at which no false reject occurs.

- Zero FMR: the lowest FRR at which no false accept occurs.

- Failure To Capture Rate: the rate at which the biometric acquisition device fails to automatically capture the biometric features. A high failure to capture rate makes the biometric system hard to use.

1.5 Thesis Motivation

This thesis focuses on fingerprint ridge orientation modelling. The importance of ridge orientation can be deflected from the fact that it is inevitably used for detecting, describing and matching fingerprint features such as minutiae and singular points. Therefore, the orientation field (OF) is a prerequisite for high quality algorithms in fingerprint based personal authentication systems. The research carried out in this thesis is motivated by the below mentioned subsections.

1.5.1 OF prediction for improved feature extraction

Low image quality arises from wet, dirty or greasy fingers and thereby is existent even on impressions made by the highest quality sensors. Figure 1.10 shows two examples of fingerprints with low quality regions. The first fingerprint image (a) shows the effects of wet fingerprints on a sliding sensor, while the second image (b) shows the impression of a greasy finger on an off-line print.

Researchers [70, 38, 82, 11, 1, 95, 8, 26] have found ways to bypass the problem in certain cases by using frequency and orientation selective filters. Usually the successful application of these filters is possible, only if the affected region is correspondingly small. This filtering schema, often referred to as contextual filtering, suppresses unwanted noise and extracts the last bit of information available in the affected region. It increases the detection of correct minutiae and leads to less spurious minutiae. Therefore correct orientation estimation can make a strong difference in the final performance of an fingerprint verification system. The required frequency parameter of these filters can be estimated using any good quality region in the fingerprint image - and if necessary the same parameters can be applied to other regions. However, the computation of the ridge orientation must be done locally, meaning that it needs to be computed for every region in the fingerprint image separately. Unfortunately, determination of ridge orientation becomes more difficult as image quality degrades. Thus even the 'best' orientation estimation algorithm will fail - resulting in a chicken-egg problem: due to the low image quality, one can not extract the orientation; but using the correct orientation, an enhancement of the image could be accomplished.

A visual example showing the perspective effects of image enhancement is given in Figure 1.11. In this experiment we simulated low quality by adding synthetic noise to a given fingerprint image. This simulated scenario, where the 'ground truth' fingerprint (including its OF) is available, shows the benefits of correctly estimated ridge orientation. For illustration, the two images were matched against the original image using a commercial fingerprint matcher. While for the noisy OF the matcher reported 30%

similarity to the original image, it reported more than 65% for the image enhanced using the ground truth OF. This illustration also shows how the large number of spurious minutiae can be drastically reduced by using the ground truth OF.

1.5.2 Compression

Police forces use automatic fingerprint identification systems to match fingerprints in order to identify individuals during criminal investigations. In these systems, fingerprint image compression is essential because AFIS data bases may contain several million fingerprint images. The Discrete Wavelet Transform (DWT) is widely used in image analysis and coding (see [81] and references therein). The compact support of the basis functions of the DWT implies an ability to adapt to local image structures. The U.S. Federal Bureau of Investigation (FBI) has specified an adapted wavelet method for their own use in large fingerprint data bases [69].

Potential improvements of this compression schemes could be based on the ridgelet transformation - which, for its application requires orientation information. Ridgelet compression is different from wavelet compression in that sense that it exploits the high directional redundancy in fingerprint images [71].

Another possibility results from [18] who show that fingerprint texture can be almost perfectly approximated using a so called AM-FM modulation. In this scheme, a fingerprint image is modelled as a 2D frequency modulation signal whose phase consists of the continuous part and a spiral part (for modelling minutiae). The authors show that the reconstructed fingerprints contain only few spurious minutiae. Again, the reconstruction ('de-compression') of the fingerprint requires the precise OF of the fingerprint.

A natural consequence of such a compression schemes is the requirement of an OF model which uses the smallest possible number of coefficients.

1.5.3 Classification and Indexing.

Formally, fingerprint classification refers to the problem of assigning a fingerprint to a class in consistent and reliable way [44]. Generally, fingerprint classification is based on global features, such as global ridge flow and singular points. These features can only be computed from the fingerprints OF. Allusively, all existing classification techniques depend on a correctly estimated OF.

Other shortcomings of many systems which are based on the Henry classification schema (see Figure 1.4) are that the different classes are unevenly distributed among a small number (typically 6) of Henry classes. Consequently, a single fingerprint class can still contain a huge number of entries to search among. Moreover, some fingerprints are naturally ambiguous which makes them difficult for exclusive classification. To address these issues, a continuous classification has been proposed in literature [60]. In the continuous classification schema fingerprints are not partitioned into disjoint classes, but instead indexed by numerical vectors that summarize their main features. The numerical vectors constitute a multidimensional feature space, in which each fingerprint corresponds to a single point. Retrieval assumes that similar fingerprints tend to cluster together in the feature space. Thus, a query fingerprint should find its match within a given distance by checking against its closest neighbours projected in the feature space. Model coefficients would represent a suitable feature for the described continuous classification and indexing method.

1.5.4 Intrinsic coordinate systems for the extraction of Finger Codes

Parametric models which are invariant under an Euclidean transformation can be used to define a coordinate system which is invariant to translation and rotation. Such a intrinsic coordinate system could immensely simplify the matching process. When using the intrinsic coordinates instead of pixel coordinates, minutiae are defined with respect to their position in the OF. Instead of the common practice of treating the OF and the minutiae as two separate descriptions, of which one is used for classification and the other one for matching, one feature vector would be used. Using intrinsic coordinates the matching of fingerprints would depend mainly on the enrolment module, but less on the matcher itself. Such a matcher would query fingerprints in the feature space, where a matching between fingerprints corresponds to searching nearest neighbours in a feature space. Thus, a intrinsic coordinate system could simplify the matching to the above mentioned indexing task. On top of the mentioned advantages, an intrinsic coordinate system would aid biometric key extraction and biometric encryption methods. In conclusion, the intrinsic coordinate system can be seen as the link between the two feature levels (minutiae and flow pattern) and would simplify and speed up fingerprint matching. We want to note, that up today there does not exist a method fulfilling the above requirements.

1.6 Outline

This Dissertation is structured as described in the follwing:

1. This Chapter gives an general introduction on biometrics. The motivation and an outline of the thesis is given.

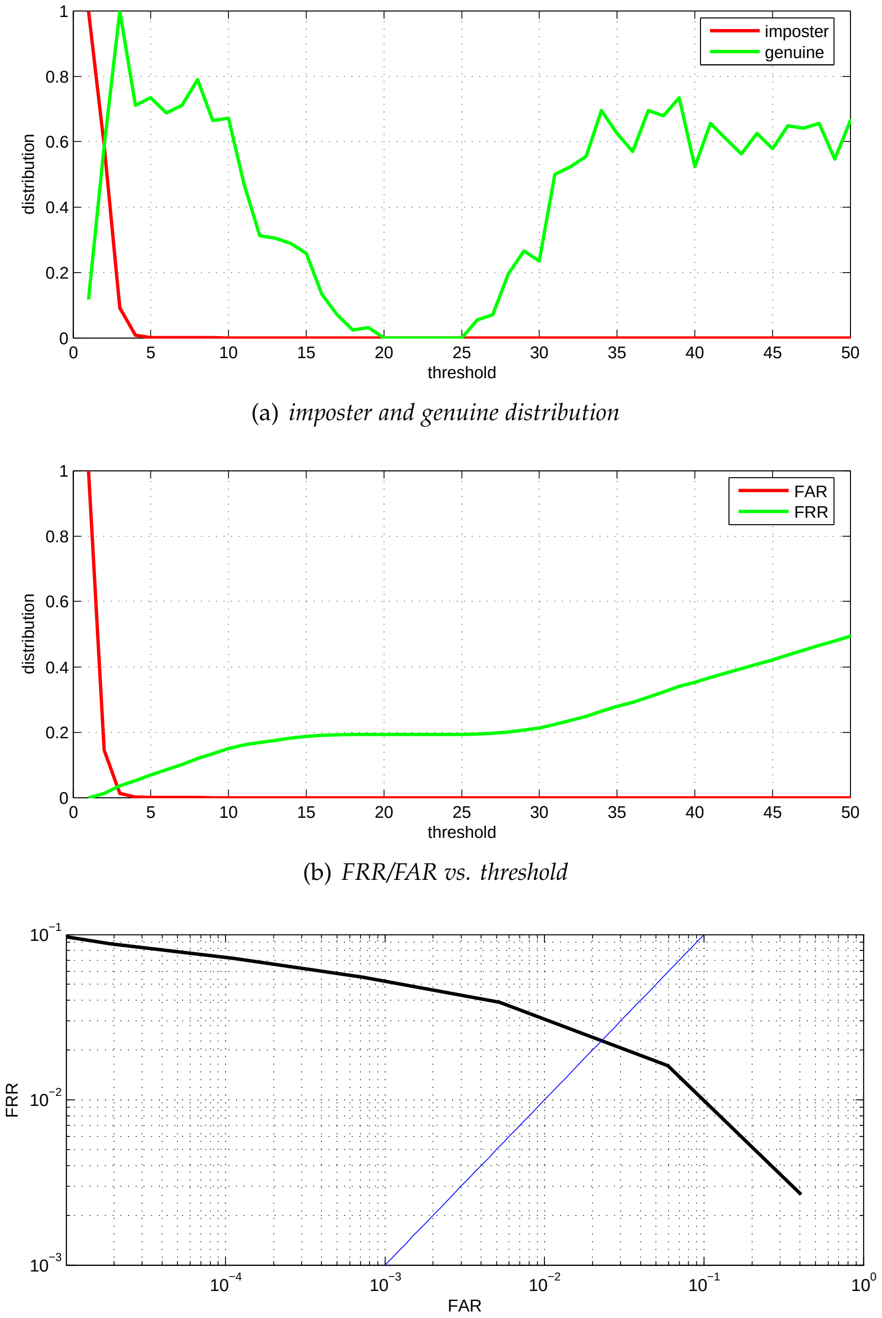
2. Chapter 2 discusses related work on fingerprint ridge orientation modelling. We will discuss the different representations of orientation and mention the problems which arise due to orientation being a π cyclic quantity. Furthermore a comparison of available fingerprint orientation models is given.

3. Chapter 3 proposes a fingerprint ridge orientation model which addresses the problem of smoothing orientation data while preserving high curvature areas.

4. Chapter 4 applies a priori knowledge to the fingerprint ridge orientation models which are proposed in chapter 3. The main contribution of this chapter is the application of a priori knowledge to the process of fingerprint ridge orientation. The model constraints the orientation to vary only in ways as they occur in nature.

5. Chapter 5 describes enhancement algorithms in order to suppress noise in the input images. We choose an available method from literature and demonstrate the effectiveness of the fingerprint ridge orientation estimation models proposed in the previous chapters.

6. Chapter 6 concludes the thesis, summarizing the main results obtained and outlining future research lines.

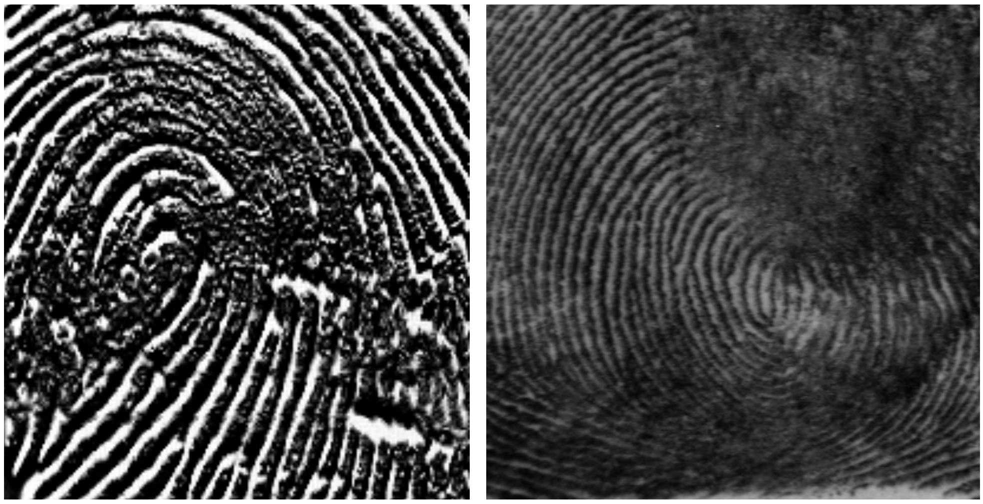


(a) *imposter and genuine distribution*

(b) *FRR/FAR vs. threshold*

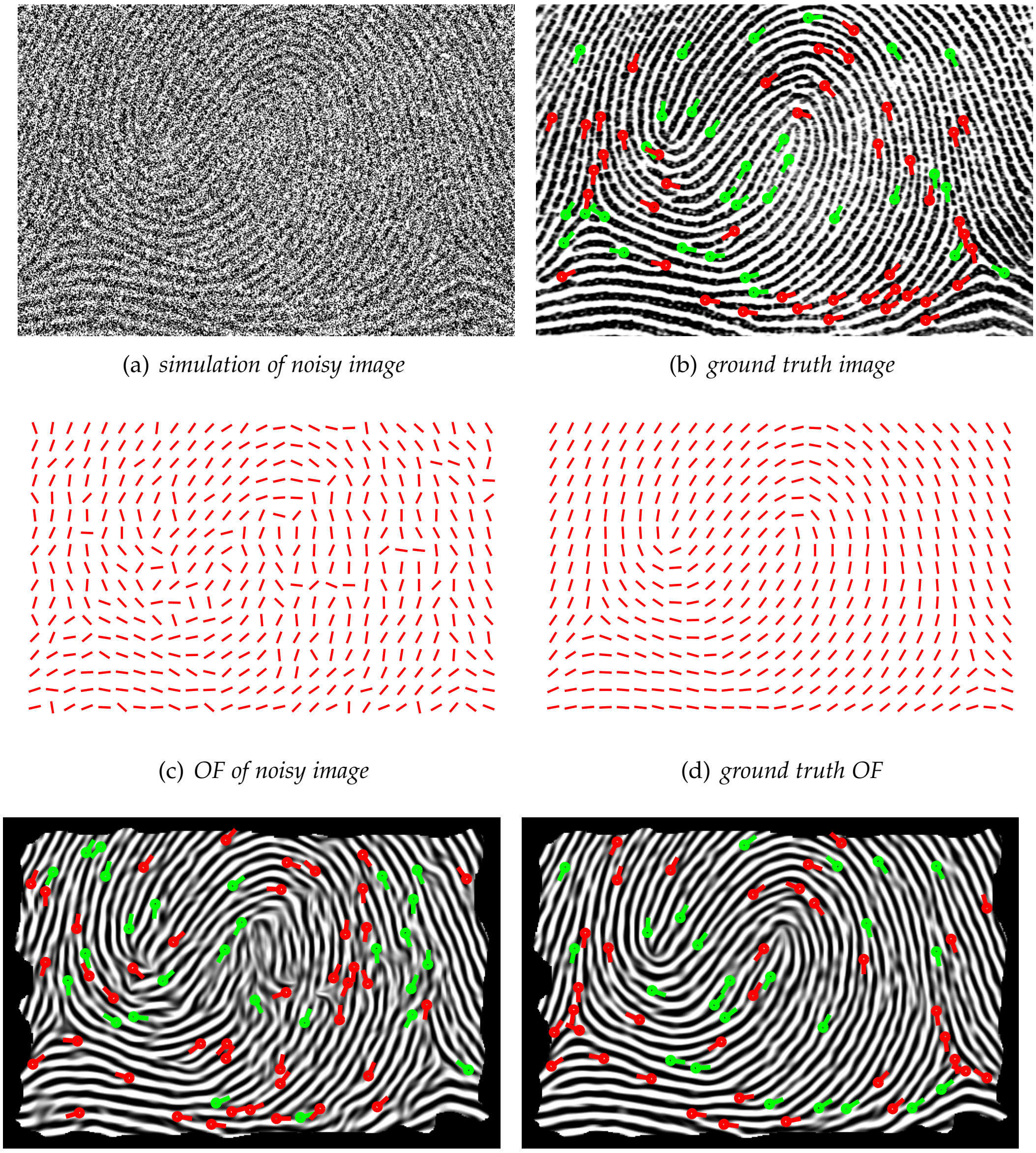
(c) *Receiver Operating Characteristic (ROC)*

Figure 1.9: **ROC curve**. Evaluation of a fingerprint verification algorithm using the FVC2004db2a dataset. First, genuine and imposter distributions were computed using the matcher (Figure (a)). FRR and FAR are derived from the score distribution, shown in Figure (b). Finally, the ROC curve is derived from the FRR and FAR distributions in Figure (c) and is eliminates the use of any thresholds.



(a) (b)

Figure 1.10: **Examples for low quality fingerprint images.** Figure (a) shows the effects of a wet finger on a low cost sliding sensor (Atmel FingerChip AT77C101B). Figure (b) shows a off-line scan from NIST (special database 4 [91]). Low image quality results in ridges and valleys which can not be well separated. Note that this is a prerequisite for orientation estimation, hence it is not possible to compute the orientation at the affected regions. As mentioned in the section above, correctly estimated OF is a prerequisite for successful image enhancement.



(a) *simulation of noisy image*

(b) *ground truth image*

(c) *OF of noisy image*

(d) *ground truth OF*

(e) *enhancement of noisy image*

(f) *enhancement of noisy image using ground truth OF*

Figure 1.11: **Simulated noise scenario.** In this experiment we simulated low quality by adding synthetic noise to a given image. This simulated scenario, where the 'ground truth' OF is available, shows the benefits of correctly estimated ridge orientation. Figure (e) and (f) are enhanced versions of Figure (a). While for the noisy OF the matcher reported 30% similarity to the original image, it reported more than 65% for the image enhanced using the ground truth OF. Furthermore, this illustration also shows how the large number of spurious minutiae can be drastically reduced by using the ground truth OF in Figure (f).

Chapter 2

Related Work

Contents

Abstract

This chapter discusses prior work related to this thesis. First, we will give an overview of the orientation estimation process. We will discuss the different representations of orientation, which are necessary because orientation is a π cyclic quantity. In the second section of this chapter we give a list of available fingerprint ridge orientation models in literature. This literature will be classified based on different aspects. The third part of this chapter shows a deep analysis of the standard orientation averaging method which is based on vectorial orientation smoothing. Furthermore, we will discuss the Bias-Variance trade-off applicable to the orientation smoothing.

2.1 Orientation Estimation, Averaging and Representation

As we have discussed in the introduction (Chapter 1) fingerprint matching inherently depends on correctly estimated fingerprint ridge orientation. This section discusses the steps necessary for fingerprint OF estimation.

Image gradient directions can be extracted for the full 360 degree range. Ridge orientation is orthogonal to these image gradients and therefore is defined only for 180 degrees. Figure 2.3 shows a simplified sketch illustrating this fact. An example, using a real fingerprint image is given in Figure 2.3.

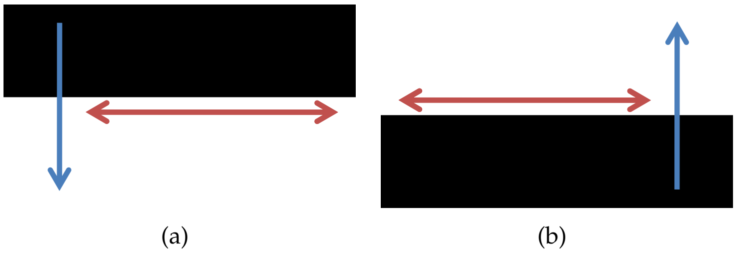


Figure 2.1: Two examples showing the gradient direction (blue) and orientation (red). Note that although the image gradients in this two examples point in the opposite direction, the resulting orientation is the same.

Due to noise in images, orientation estimation can be understood as the task of estimating the 'average' orientation of a bunch of measured vectors.

In the vast majority of existing schemes, the described 'bunch of measured vectors' is extracted by the analysis of grey value gradients. It is well known that the unbiased computation of gradients from discrete signals is a difficult problem in itself. There has been much work performed for computing derivatives with high precision. We refer the reader to [59] for a detailed overview on gradient estimation techniques.

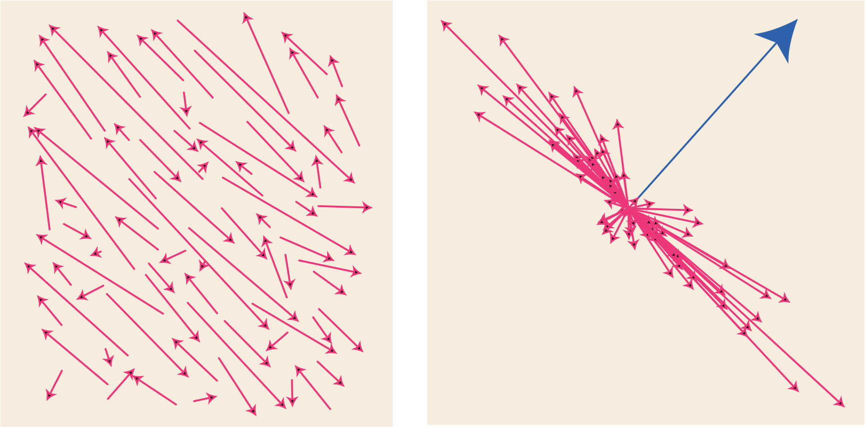


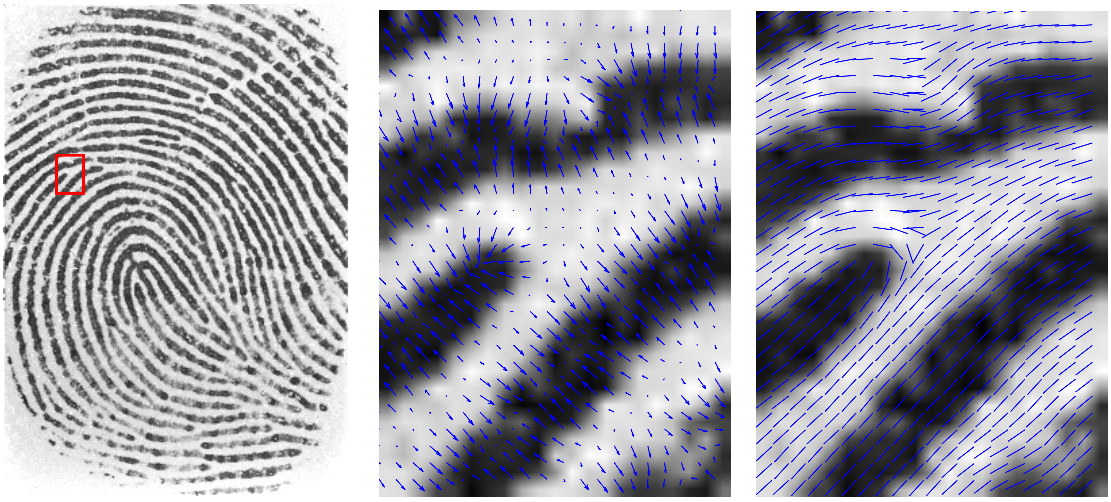
Figure 2.2: **central question:** How can the estimated gradients be 'averaged' to one orientation? Illustration taken from Mester and Mühlich [64], who argue that orientation averaging is a Least Squares Problem.

Almost all established techniques for orientation averaging are based on the same criterion, disguised in several apparently different formulations [65] , and finally leading to an eigenvalue problem. In the following, we list the most prominent (from the fingerprint community point of view) approaches, which are all equivalent:

- Witkin and Kass [48] proposed the doubling of the orientation angle. After doubling the angles, opposite gradient vectors will point in the same direction and therefore will reinforce each other, while perpendicular gradients will cancel. Due to its simplicity (see Figure 2.7 and 2.7) this method has been heavily adopted by the fingerprint community and is applied by the majority of the literature. Note though, that many authors (especially outside of the biometrics community) cite Granlund [31] as the one who invented the double angle notation in a very early paper in 1978. See [4,44] for an detailed overview of this method.

- Bazen and Gerez [4] propose a PCA based approach to orientation smoothing. A proof is given by the authors that their method is equivalent to the above mentioned method.

- Bigün and Granlund [5] used the knowledge that orientation can be determined by either analysing the joint statistics of gradient vectors, or the angular distribu-



(a) *typical fingerprint image* (b) *gradients* (c) *orientation*

Figure 2.3: **direction versus orientation:** Image gradient directions can be extracted for the full 360 degree range. Ridge orientation is orthogonal to these image gradients and therefore is defined only for 180 degrees. Figure (a) shows a fingerprint image. The red rectangle is enlarged in Figure (b) and the image gradients are plotted. Figure (c) shows the computed orientation.

> tion of signal power in its energy spectrum. Interestingly, this idea resembles the approach of Witkin and Kass [48] which has been published in the same year.

For the two dimensional case, the double angle representation (which is a vector representation), is equivariant to the tensor representation of orientation as proposed by Knutsson in [52]. This fact has been proven by Nordberg et al. in [68].

In [74] Perona extended the idea of diffusions to orientation like quantities. His paper presents details for a discrete implementation and explores the problem analytically and experimentally. A number of open problems are proposed at the end of this paper.

Farnebäck [17] describes a method of estimating the orientation tensor more accurately. It is based on a local polynomial model of the signal, using polynomials up to the second order. The approach is based on making a least squares approximation of a polynomial model to the local signal. Through a so-called theory of normalized convolution, the parameters can be computed as filter responses where the filters are given by dual basis functions relative to the polynomial basis.

Another interesting idea for achieving higher accuracy in orientation averaging algorithms is available from Mühlich and Mester in [66,65,64]. They show that averaging is a Total Least Squares problem, which will agree to our argumentation (formulated in the last section of this chapter) that the optimization has to be performed with respect to the important entity, (i.e. the sought orientation angle), but not with respect to intermediate entities.

2.2 Literature on Fingerprint Ridge Orientation Models

This section discusses available methods in literature. The motivation of available methods is manifold. All of them tempt to solve at least one problem mentioned in the introduction. In the next section we will outline criteria on which these methods can be differentiated.

- **Solution to the inverse problem.** Some models are applicable only for the qualitative explanation of fingerprint flow patterns, meaning given the model parameters a fingerprint flow pattern can be generated but not vice versa. On the other hand, there exist models which can solve the inverse problem of finding those model parameters which fit the model to the given data. Since we are interested in model coefficients for various applications (indexing, classification, see Chapter 1), our main interest lies in such methods.

- **Chicken egg problem.** Many existing models can only be applied when the fingerprints SPs class and position is given. Since the computation of SPs is depending on the OF itself, it is obvious that these models are stuck in a 'chicken egg problem' for low quality images.

- **Prior Information about Fingerprints.** Another issue is the prior information about fingerprint flow patterns. Briefly, this is the ability of the model to generate (only) pattern of real fingerprint flow patterns and therefore correcting noisy OFs. This is especially important in relation with interpolation and extrapolation methods. Currently, almost all available methods comprising prior information are using differential equations whose trajectories have the structure of fingerprint ridges. We will discuss this issue in more detail in chapter 4.

- **Assumption of smoothness.** One fact, not directly stated by most of the authors, is the assumption that fingerprints always form an almost smooth pattern. While this is true for almost all of the fingerprint area, at closer examination discontinuities (singular points) can be seen. In order to remove noise a orientation model must spatially average orientation. On the other hand, it should be noted that smoothing usually will harm the high curvature information, especially SPs. This is an issue which is known as the Bias-Variance dilemma in approximation theory [6] and usually there is no circumvention to this trade off.

- **Parmetric vs free form.** Note that there exist also parameter free methods for the estimation and/or smoothing fingerprint ridge orientation. Such methods are typically based around partial differential equations (PDE) which then are solved numerically.

Below, we will give a brief overview of available methods in literature.

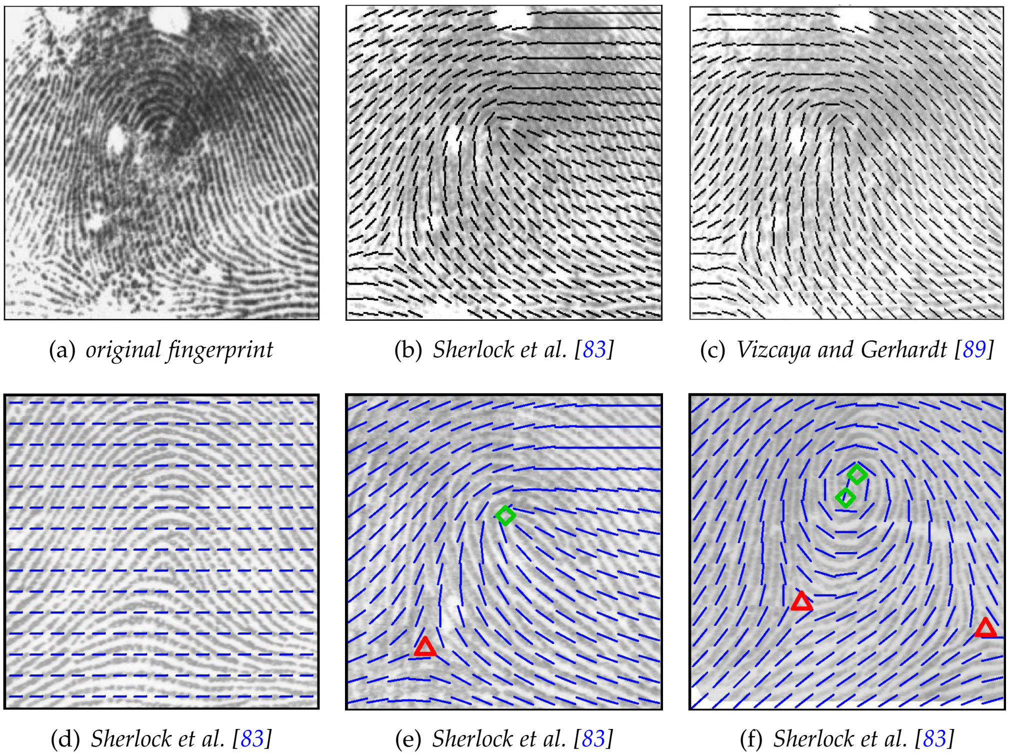
2.2.1 Models for the synthesis of fingerprint orientations

Smith [84] was one of the first to model orientation fields using by differential equations. His work was later refined by Mardia et al. [63]. A more recent work is available from Kücken et al. [54] who have presented physically motivated differential equations explaining the formation of fingerprints. All these studies aimed at a qualitative explanation of fingerprint patterns and cannot be used for our purpose.

2.2.2 Parametric models for the inverse problem

Sherlock and Monroe [83] model the orientation using a so-called zero-pole model. Unfortunately, this orientation model is too simple and fails describing the ridge orientation accurately. Vizcaya and Gerhardt improve on this model in [89] by using a piecewise linear model around the singular point. Since the distances and orientations between singular points are not modelled correctly the mentioned approach can not be used successfully for accurate orientation modelling. Note that, despite obvious limitations this model (and variants of it) can be successfully used for the synthesis of fingerprints (i.e. in the prominent SFINGE [44] approach of Capelli et al.) A remarkable feature of the zero-pole approach is the elegant way of modelling SP by the use of zeros and poles. See Figure 2.4 for illustrations on the zero-pole model.

Capelli et al. [7] describe a method for the reconstruction of fingerprints from standard minutiae templates. For orientation reconstruction the authors propose a method based on the before mentioned zero-pole model [83]. Similiar to Vizcaya and Gerhardt [89] the model is improved using an a piecewise linear function. Given the positions and types of the SPs, these piecewise functions can be computed. Each piecewise function is defined only for a quadrant of the image. Due to this reason the final cost function for the full fingerprint contains eight so called control points. The mentioned cost function is then optimized using a Nelder-Mead simplex [67] optimization while


(a) *original fingerprint*

(b) *Sherlock et al. [83]*

(c) *Vizcaya and Gerhardt [89]*

(d) *Sherlock et al. [83]*

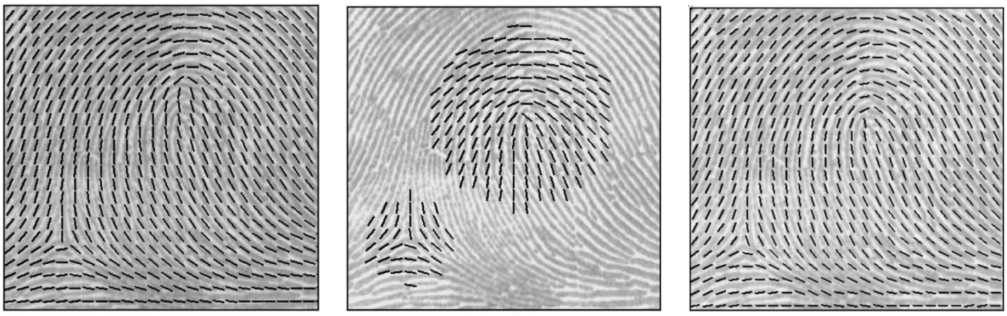
(e) *Sherlock et al. [83]*

(f) *Sherlock et al. [83]*

Figure 2.4: **First row:** Comparison of the original zero-pole model (Sherlock et al. [83], shown in Figure (b)) with one of its variants (piecewise linear model by Vizcaya and Gerhardt [89], shown in Figure (c)). Note that, despite obvious limitations the latter model can be successfully used for the synthesis of fingerprints (i.e. in the prominent SFINGE [44] approach of Capelli et al.) **Second row:** the zero-pole model fitted to fingerprints. Another obvious limitation of this method is that arch type fingerprints (no SPs) can not be modelled. A remarkable feature of the zero-pole approach is the elegant way of modelling SP by the use of zeros and poles (visible in Figure (e) and (f)). Original images courtesy of [103,39].

simultaneously applying heuristics. In the last stage this approach performs post processing in order to remove possible interpolation artefacts.

In a very recent paper Huckemann et al. [41] propose a global OF model-based on Quadratic Differentials. Quadratic Differentials are a mathematical tool for extremal problems for mappings and moduli of complex domains. The authors approach can be seen as specifically tailored Quadratic Differentials for the purpose of fingerprint ridge orientation modelling. The coefficients (typicall five) of the model are geometrically interpretable and have a clear meaning. Typically, such approaches create flow patterns which do not occur in natural fingerprints. Another drawback of this method is the abil-



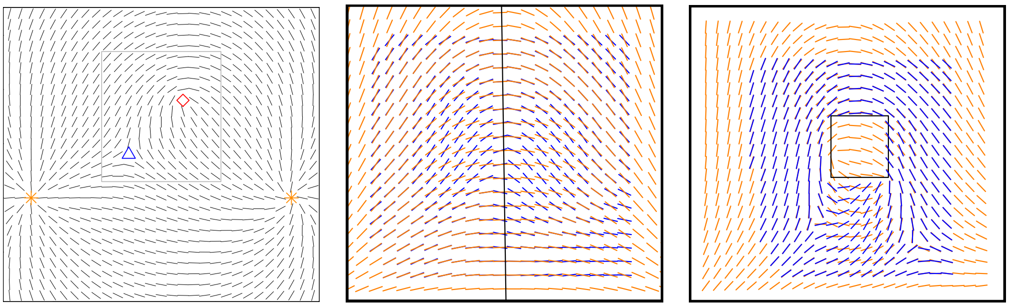
(a) *global approximation* (b) *local approximations* (c) *combination of both*

Figure 2.5: **Combination Model (images courtesy of [103]).** Many available models in literature proceed as shown in the above images. First, the global field is approximated. High curvature areas (typically around SPs) are modelled separately. In a third step the local and global orientations are combined.

ity to not generalize well for all types of fingerprints. This is especially true for arches. However, it should be noted that in comparison to other approaches, this approach has been evaluated on a large enough database which affirms numerical stability as well as the practical applicability of the method. Figure 2.6 shows the model fitted to some fingerprint OFs using the approach of [41].

A very prominent method is described by Zhou and Gu in [103, 102]. In a first step, the authors propose to model the global orientation using power series. Since the global model only poorly describes singularities, in a second step the authors propose to model the singularities using a so called point-charge model [102]. No solid rules are defined for using this combination method in larger databases. Furthermore the algorithm depends on reliable detection of SP - a problem which itself is proven to be hard to solve. Figure 2.5 shows an example of the method as described in [103].

Li [57] et al. model the orientation of fingerprints using higher order phase portraits. Therefore the method first divides the fingerprint into several regions and approximates them using piecewise linear phase portraits. In a further step their method computes a global model using the piecewise linear phase portraits. Unfortunately, the algorithm is very difficult for an application in practice. Three reasons can be outlined : a) robust detection of SPs b) robust separation of fingerprints into predefined regions c) non-linear methods having impractical runtimes. Recently, the same authors have proposed [58] a modification of their algorithm and gave a thorough stability analysis. Although some of the above mentioned issues have been addressed, the method still lacks the proof of usefulness on larger scale applications (eg. tested only on 38 pre-selected fingerprint images). Furthermore the model still demands constraints for each of the singularities.



(a) (b) (c)

Figure 2.6: **Approach of Hotz et al.** Subfigure (a) shows a orientation field generated by Quadratic Differentials. The hypothetical fingerprint area is shown as a grey window. The artificial poles, essential for this approach, are marked by orange stars. Note also that the lower image plane needs to be 'blanked' out for the application to real OFs. Subfigure (b) shows the model (orange) fitted to a arch type fingerprint (blue). Apparently the model is not able to perfectly generalize. In Figure (c) the interpolation ability of the model is shown. The interpolation quality is high only for certain areas and certain fingerprints. Furthermore the method often fails to generate plausible fingerprint flow patterns. All illustrations taken from [39].

Wang et al. [90] presented a Fingerprint Orientation Model based on trigonometric polynomials. Their approach (named FOMFE) does not require prior knowledge of singular points. One mentioned application is the restoration of poor quality fingerprint areas through contextual filtering. Another application is database indexing based on the model parameters.

2.2.3 Non-parametric models

An axiomatic approach for interpolation and smoothing OFs based on partial differential equations (PDEs) has been presented by Chessel [9]. Their method is able to preserve singularities. Two applications of the so called Absolutely Maximizing Lipschitz Extension (AMLE) operator are shown, which take advantage of its faithful and continuous interpolation of orientations to extract curves in poorly contrasted image. The authors suggested that one of the application of their method is the orientation interpolation in biological images.

Another method not directly related to orientation modelling, but orientation computation for reconstruction of fingerprint images from minutiae is presented by Ross et al. in [80]. In their schema the orientation field is computed using minutia triplets. First a weight which is depending on the distance to each of the three minutiae is computed.

This weight is then used to assign a value to each pixel in the minutiae triplet. In the last stage the resulting orientation is smoothed using an averaging filter. A disadvantage of the method is clearly that it can not be applied to large regions (or regions with missing minutiae triplets) and usually gives rather poor performance for high curvature areas.

Table 2.1 gives an overview of the above cited methods. These methods can be differentiated by being parametric or free form and by the ability of solving the inverse problem. Furthermore the table shows whether the chicken-egg problem (the necessity for computing SPs before the model coefficients) is solved and if the possibility for applying prior knowledge is given. The table also sums up the principles exploited for OF modelling. DE denotes to 'differential equation', RF stands for 'rational function' where the orientation is modelled using a nominator and a denominator. Many methods also apply vectorial data (VD) computed using the sine and cosine from the OF for modelling a fingerprints OF.

literature overview					
Authors	Method	parametric	inverse problem	SPs necessary	prior information
Smith [84]	DE	no	no	-	-
Mardia et al. [63]	DE	no	no	-	-
Kücken and Newell [54]	DE	no	no	-	-
Sherlock and Monroe [83]	RF	yes	yes, with restrictions	yes	no
Vizcaya and Gerhardt [89]	RF + piecewise linear functions	yes	yes, with restrictions	yes	no
Capelli et al. [7]	RF + piecewise linear functions	yes	yes, with restrictions	yes	yes
Zhou and Gu [103, 102]	RF using VD, (global + local)	yes	yes	yes	no
Li et al. [57]	DE,RF (linear phase portraits)	yes	yes	yes	yes
Wang et al. [90]	RF using VD	yes	yes	no	no
Huckemann et al. [41]	DE,RF (quadratic differentials)	yes	yes	no	yes
Ross et al. in [80]	minutiae triplets	no	yes	no	no
Chessel et al. in [9]	PDE	no	yes	no	no
proposed	RF using VD	yes	yes	no	optional using PCA

Table 2.1: Overview of available literature on modelling fingerprint ridge orientation. DE=differential equation, PDE=partial differential equation, RF=rational function, VD = vectorial data (computed using sine and cosine function), PCA = principal component analysis

2.3 Modelling Fingerprints using the Double Angle Representation

Obviously, one of the simplest ways for modelling fingerprint ridge orientation is using the double angle representation as described by Rao and Schunck in [78]. As we have described above, the double angle representation can be used to apply standard numerical approaches to orientation data, e.g. convolution for smoothing (averaging) orientation data. Equation 2.1 sums up the mathematical context of this method.

$$O(x,y)_s = \frac{1}{2}\arctan\frac{\sin(2O(x,y)) * G}{\cos(2O(x,y)) * G} \tag{2.1}$$

$O(x,y)_s$ denotes the smoothed orientation, which can be obtained by convolving the sine and cosine part of the doubled angle orientation by a convolution mask G. For more details on this method we refer the reader to the Appendix. However, it is also possible to fit a set of polynomials to the orientation data, as this representation solves the discontinuity between zero and π. See Figure 2.7 and 2.8 for an illustration.

An approximation of a (discrete) two dimensional function $f(x,y)$ can be done using use linear combinations of n basis functions $\phi(x,y)$ (usually polynomials).

$$f(x,y) \approx \sum_{j=0}^{n} a_j \phi_j(x,y) \tag{2.2}$$

This technique is also known as regression [6]. We will introduce this technique in detail in Chapter 3.

For the task of fingerprint ridge orientation modelling, linear regression can be applied to the double angle representation. In application, this means that instead of using convolution for smoothing the orientation data, polynomials are used for smoothing and approximation of this data:

$$sin(2O(x,y)) \approx \sum_{j=0}^{n} a_j \phi_j(x,y) \qquad cos(2O(x,y)) \approx \sum_{j=0}^{n} b_j \phi_j(x,y) \tag{2.3}$$

where $2O(x,y)$ represents the doubled angle of the orientation at the coordinate (x,y). Finally, the modelled orientation can be back converted using the following formula:

$$O(x,y) = \frac{1}{2}\arctan\frac{\sum_{j=0}^{n} a_j \phi_j(x,y)}{\sum_{j=0}^{n} b_j \phi_j(x,y)} \tag{2.4}$$

The above described modality, is applied in several approaches [90,103,102,57,58] available in literature. We found that this straightforward approach for modelling fingerprint OFs leads to problems which are summarised in the next two subsections.

2.3.1 Issues with Non-linearity and Error Propagation

From Equation 2.4, one can observe that any two coefficients a_j and b_j influence the final orientation in a non-linear way. Errors in the approximation of the individual data parts could either sum up or cancel each other in the final quantity (the orientation). Especially notable is the fact, that the actual amplitude of each polynomial can be cancelled by the division - thus making smaller amplitudes more sensitive to approximation errors. Moreover, one should note that the final quantity is further obtained trough the use of the non-linear $\arctan$ function. These facts are a clear argument against the linear and independent approximation of the vectorial orientation data.

Note that this argumentation agrees with Mester et al. [66, 65, 64] who show that orientation estimation and averaging is a Total Least Squares problem and the spatial smoothing using the double angle representation approach has drawbacks.

The second important issue, concerning the parameter estimation is error propagation. The approximation of vectorial orientation data using the above mentioned scheme does not deal with this phenomenon. An illustration showing this effect is given in Figure 2.9.

The real measure for fitting a ridge orientation model to a fingerprint's orientation should be directly computed by using the orientation angle as opposed to its vectorial parts. In order to minimize the least square error (of the orientation), the following non-linear function must be minimized:

$$\underset{a_j,b_j}{min} \quad \sum_{j=1}^{i} \omega_j \left[\arctan \frac{\sum_{j=0}^{n} a_j \phi_j(x,y)}{\sum_{j=0}^{n} b_j \phi_j(x,y)} - 2O(\mathbf{x}_j) \right]^2 \tag{2.5}$$

Where a_j and b_j are the desired coefficients of the polynomial approximation. i represents the number of pixels and n is the total number of basis functions. It should be noted that Equation 2.5 is not usable as formulated, due to the discontinuity problem at 0 and π.

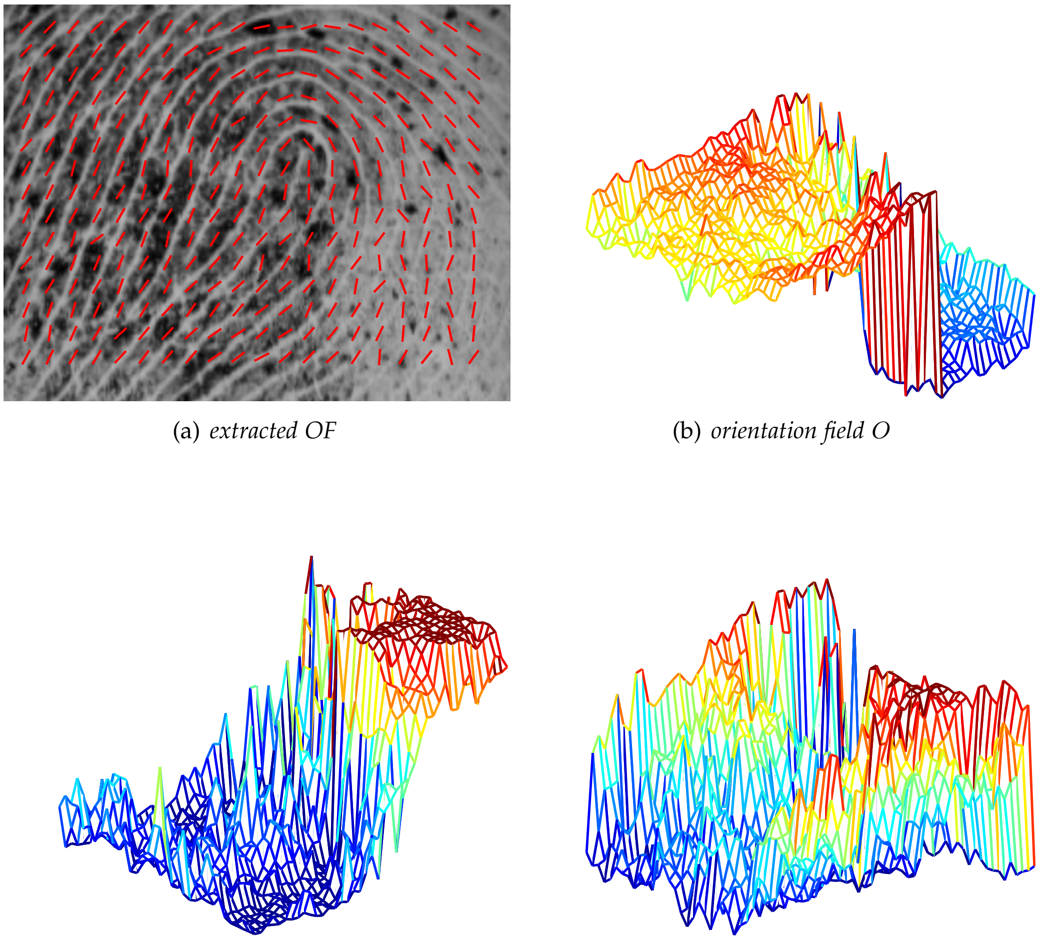
The problem of this discontinuity at 0 and π can be solved as described in the following. In an early paper Rao and R. Jain [76] proposed the sine as a distance measure for non-linear parameter estimation in linear phase portraits. In their optimization routine, they minimize the absolute values of this measure. Later, Ford and Strickland [25] suggested that the sum of squares of these distances should be minimized. One should note that in these references the authors intention is to use directly the orientation to obtain the model's parameters. As we carry out our modelling already in the doubled angle space, we have to half the angle for correct determination of the error functional. Then, rewriting the cost function results in:

$$\underset{\mathbf{a},\mathbf{b}}{min} \quad \sum_{j=1}^{i} \omega_j \left[\sin\left(\frac{1}{2} \arctan \frac{\sum_{j=0}^{n} a_j \phi_j(x,y)}{\sum_{j=0}^{n} b_j \phi_j(x,y)} - O(\mathbf{x}_j) \right) \right]^2 \tag{2.6}$$

2.3.2 Impact of Spatial Smoothing on Singular Points

SPs cause discontinuities in the OF. This discontinuities are also present in the vectorial data. (see Figure 2.10 and 2.11 for an example). Such discontinuities represent a problematic case for data approximations using polynomials, as one needs a large number of parameters for their approximation. On the other hand, low order polynomials are preferred as they have better smoothing capabilities.

A visual explanation for the problem is given in Figure 2.11, where the vectorial orientation data of a loop type fingerprint is given. In the centre, these two surfaces contain a discontinuity, a jump from -1 to 1. The presence of this discontinuity is important, because the SP is defined by the roots of this discontinuity. This fact can be easily verified by back conversion of the vectorial data using the formula in Equation 2.4. Modelling the vectorial data using a second order Fourier-Series (25 parameters each) results in shifting their zero-poles and hence also in shifting of the SP. Furthermore, the Fourier series straightens the discontinuity, which results in false orientation around the SP (see Figure 2.10). This problem plaques not only standard numerical methods for smoothing but also every potential ridge orientation modelling approach. It is not important to precisely approximate the individual discontinuities. Because the appearance of a SP is strictly defined by the roots and the ratio of the two shown surfaces. Existing approaches put high emphasize on the exact and separate approximation of the vectorial data. Our argumentation is that the approximation should be done in a coupled way in order to precisely model the orientation data. This would reposition the roots of the vectorial data back to the correct location. Furthermore, one can use lower order polynomials, which will result in better smoothing properties and fewer coefficients.



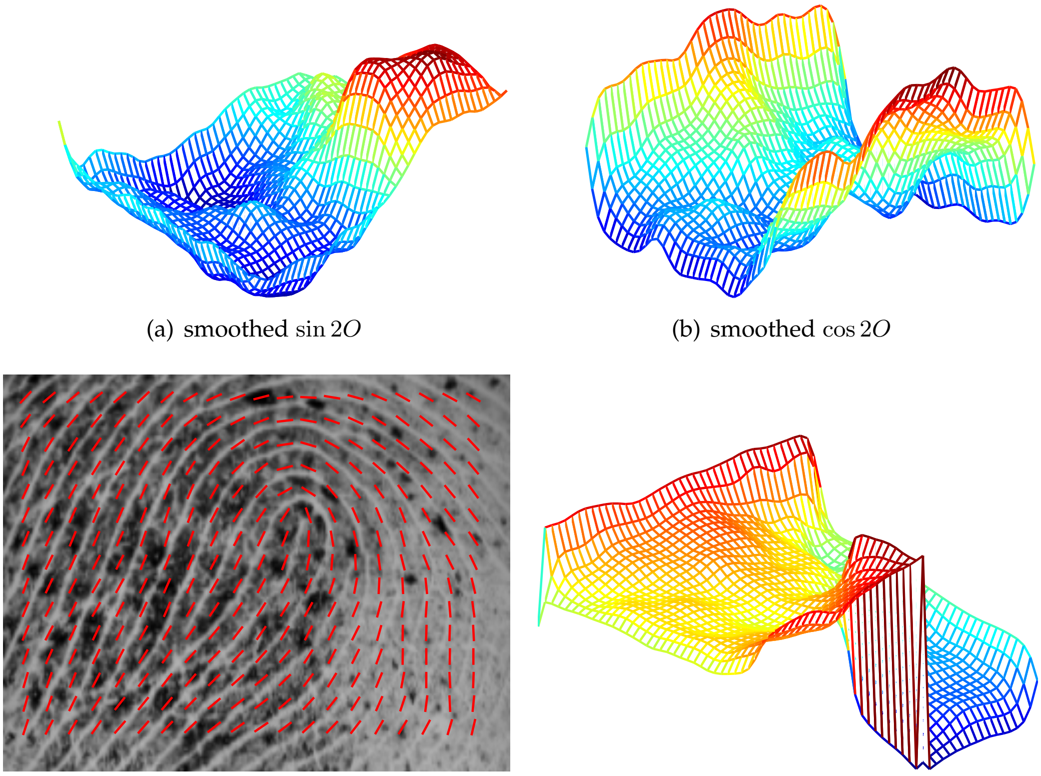
(a) *extracted OF*

(b) *orientation field O*

(c) $\sin 2O$

(d) $\cos 2O$

Figure 2.7: **The double angle representation.** Orientation modelling directly using angles as quantity faces the problem of the discontinuity between the values 0 and π (see Subfigure (b)). A popular approach [48] for solving this problem is the doubling of the orientation angle and splitting it up into a vectorial representation. This procedure removes the discontinuity problem at π and zero and allows standard numerical methods to be applied to orientation data (eg. averaging, smoothing using convolution, approximation using polynomials, ...). In Subfigures (c) and (d) the double angle representation is shown as meshgrid.



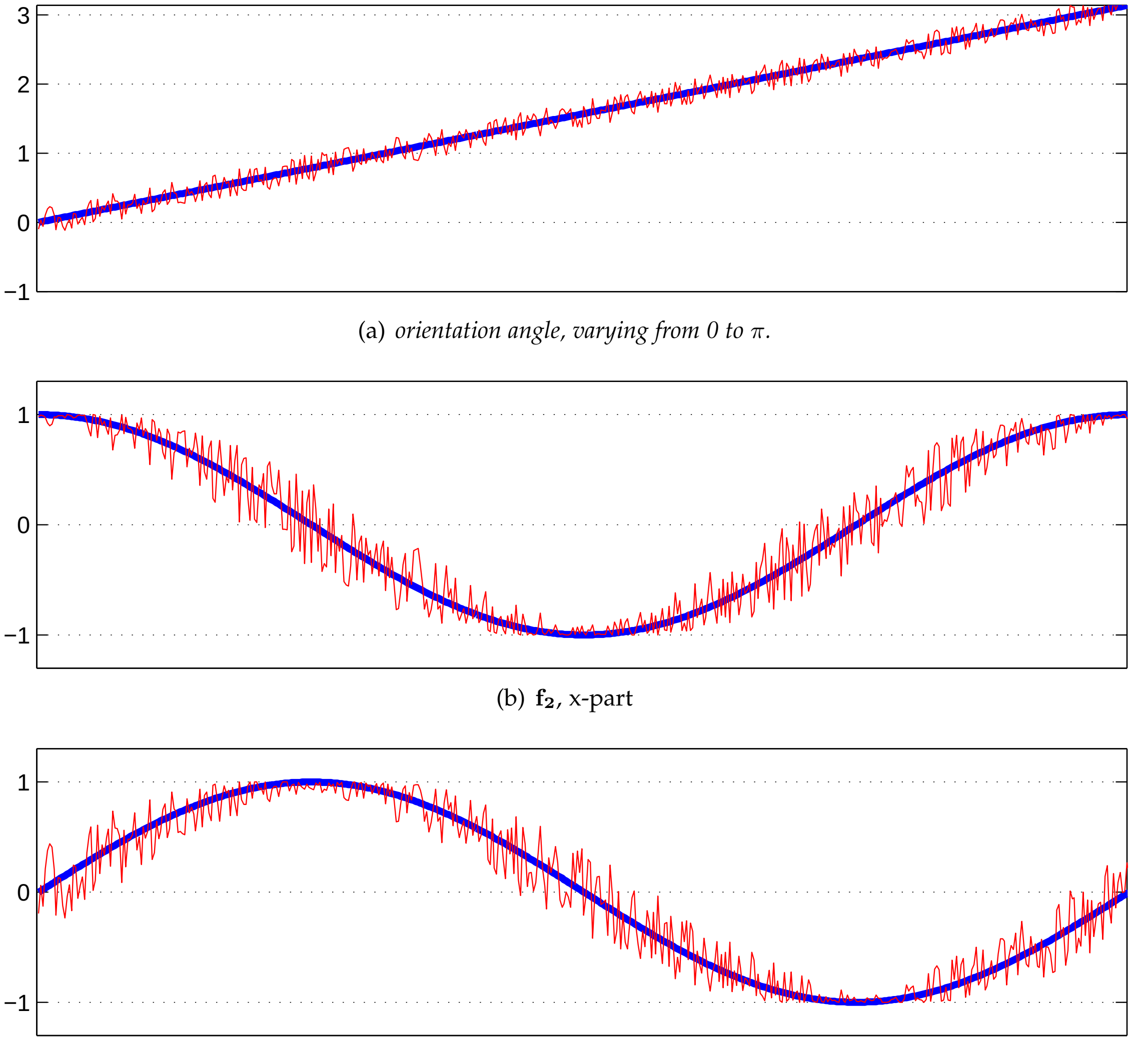
(a) smoothed $\sin 2O$

(b) smoothed $\cos 2O$

(c) smoothed OF overlaid

(d) smoothed orientation

Figure 2.8: **Modelling orientation using the doubled angle representation.** This illustration shows how the OF is modelled using the doubled angle representation. Subfigure 2.7(c) and 2.7(d) are approximated from the data given in Figure 2.7 using polynomials. Subfigures (a) and (b) show the sine and cosine parts smoothed using polynomials. Note how the back-converted OF again contains the discontinuity between 0 and π.

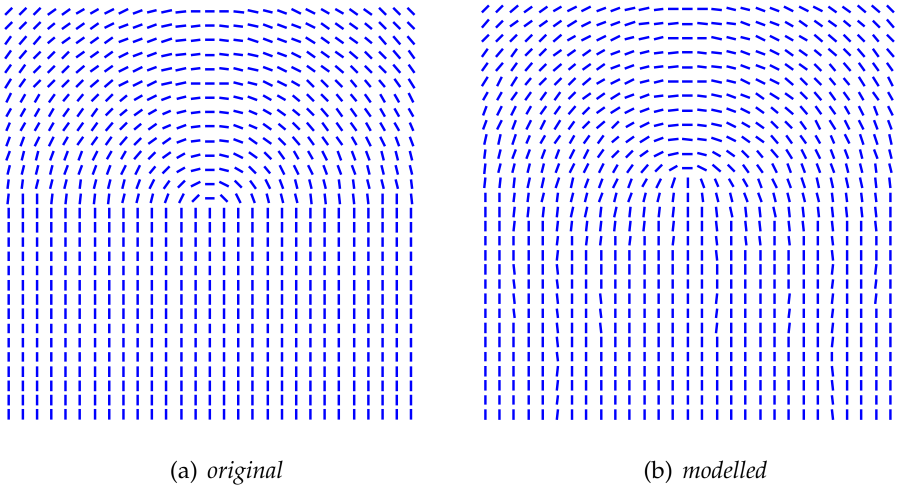


(a) *orientation angle, varying from 0 to π.*

(b) $\mathbf{f_2}$, x-part

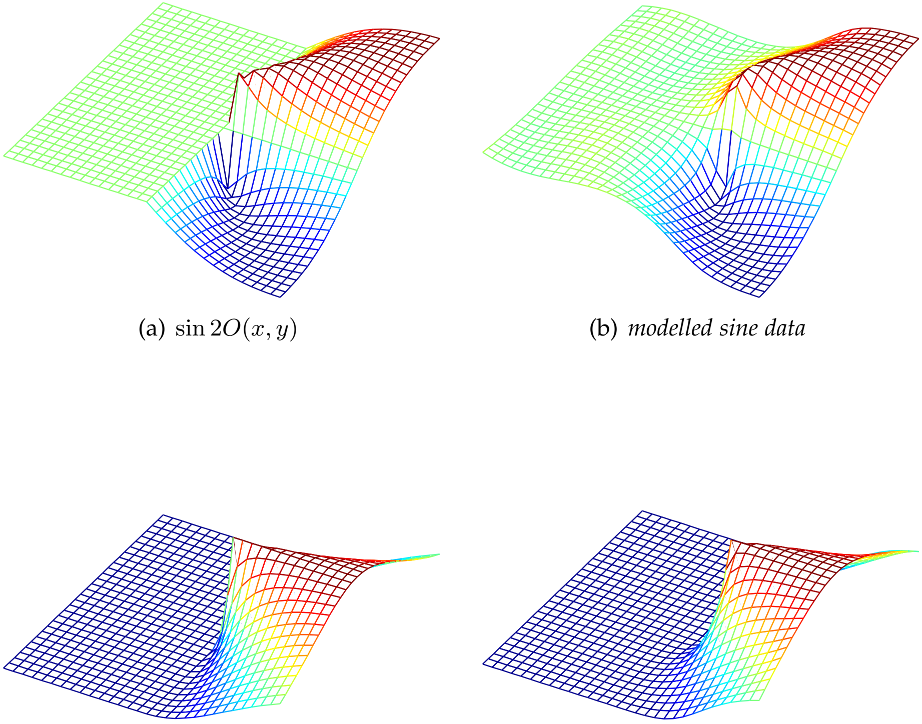
(c) $\mathbf{f_1}$, y-part

Figure 2.9: Illustration of error propagation and non-linearity. Synthetic noise has been added to the orientation data (subfigure 2.9(a)). Due to the non-linear connection of orientation and vectorial data, errors propagate further in a non-linear way. Note how the amplitude of the error changes. Independent modelling of orientation data using its vectorial parts does not account for this fact.



(a) *original* (b) *modelled*

Figure 2.10: In 2.10(a) the orientation field of a loop type fingerprint image can be seen. This image has been synthetically generated. In Figure 2.10(b) the OF is approximated by a second order Fourier polynomials. This modelling approach is akin to the approach of Witkin and Kass [48], where one separately models/smoothes the vectorial data using Fourier series. Note how the SP is shifted away from the low curvature area.



(a) $\sin 2O(x, y)$

(b) *modelled sine data*

(c) $\cos 2O(x, y)$

(d) *modelled cosine data*

Figure 2.11: In 2.11(a) and 2.11(c) the vectorial data of a loop type fingerprint can be seen (as shown in Figure 2.10). In 2.11(b) and 2.11(d) these surfaces have been modelled using a second order Fourier series. An explanation why the SP shifts after modelling can be given by looking at the zero-poles of these surfaces, which define the position of the SP. The shift of the zero-pole causes also the SP to shift.

Chapter 3

Modelling Fingerprint Ridge Orientation using Legendre Polynomials

Contents

Abstract

Based on the findings from Chapter 2.3 this chapter proposes a solution for fingerprint ridge orientation modelling. One of the main problems the presented method addresses is smoothing orientation data while preserving details in high curvature areas, especially singular points. We show that singular points, which result in a discontinuous orientation field, can be modelled by the zero-poles of Legendre Polynomials. The model parameters are obtained in a two staged optimization procedure. Another advantage of the proposed method is a extremely compact representation of the orientation field, using only 56 coefficients. We have carried out extensive experiments using a commercial fingerprint matcher and a singular point detector. Moreover, we compared the proposed method with other state-of-the-art fingerprint orientation estimation algorithms. We can report significant improvements in both, singular point detection as well as matching rates.

3.1 Generating Legendre Polynomials

The proposed method uses Legendre Polynomials as basis functions for fingerprint ridge orientation modelling. These polynomials are orthogonal in the interval $[-1, 1]$, a necessary property which improves the stability of the optimization process by creating good conditioned linear equation systems. Furthermore, this basis functions are fast to evaluate through the use of few multiplications and additions and are simple to construct and to generalize to higher dimensions.

Legendre Polynomials $\phi(x)$ are named after the French mathematician Adrien-Marie Legendre (1752 - 1833). Mathematically, the orthogonality can be described as [53].

$$\int_{-1}^{1} \phi_n(x)\phi_m(x)dx = 0$$

Where $n, m \in \mathbb{Z}$ and $m \neq n$. Each univariate Legendre polynomial $\phi_n(x)$, can be computed using Rodrigues formula [28]:

$$\phi_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} \left[(x^2 - 1)^n\right] \tag{3.1}$$

Another iterative (and faster) method to compute the Legendre polynomials uses the so called three-term recurrence relation:

$$\phi_{n+1}(x) = \frac{2n+1}{n+1}\phi_n(x) - \frac{n}{n+1}\phi_{n-1}(x) \tag{3.2}$$

where the first two Legendre polynomials are $\phi_0(x) = 1$ and $\phi_1(x) = x$. The first six basis functions are given in Table 3.3.

$$\begin{aligned}
\phi_0(x) &= 1\\
\phi_1(x) &= x\\
\phi_2(x) &= \frac{3x^2-1}{2}\\
\phi_3(x) &= \frac{5x^3-3x}{2}\\
\phi_4(x) &= \frac{35x^4-30x^2+3}{8}\\
\phi_5(x) &= \frac{63x^5-70x^3+15x}{8}\\
\phi_6(x) &= \frac{231x^6-315x^4+105x^2-5}{16}\\
\phi_7(x) &= \frac{429x^7-693x^5+315x^3-35x}{16}
\end{aligned} \tag{3.3}$$

Another important property of Legendre polynomials is the simple generalization to higher dimensions by using the method of separable variables [86]. Consider the Legendre polynomials $\phi_{n-m}(x)$ and $\phi_m(y)$ in the two variables x and y. Then one can compute the set of basis functions for the k^{th} order Legendre polynomial expansion as:

$$\phi_{nm} = \phi_{n-m}(x)\phi_m(y) \begin{pmatrix} n = 0, 1, 2, \ldots, k \\ m = 0, 1, 2, \ldots, n \end{pmatrix}$$

See table 3.1 for details on the number of parameters.

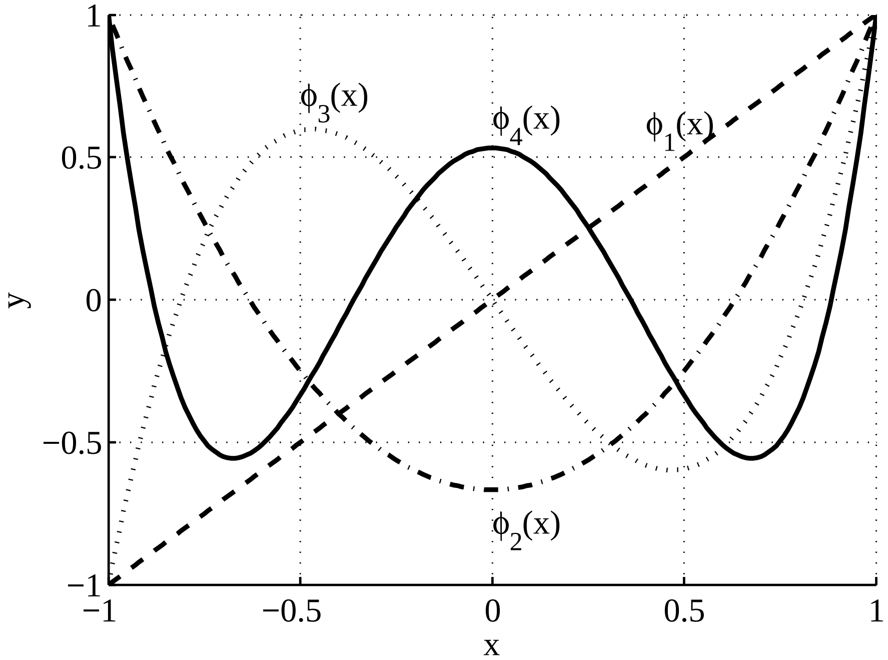


Figure 3.1: Function plot of some Legendre polynomial basis functions as given in equation 3.3.

Remarks on the Choice of Basis Functions

In the simplest case the basis functions $\Phi(x,y)$ can be constructed using the monomials known from 'common' polynomials:

$$\begin{aligned}\Phi(x,y) &= [1 \quad x \quad y \quad xy \quad \ldots \quad x^{n-k}y^{k}] \\ \text{with } n &= 0,1,2,\ldots \qquad k = 0,1,2,\ldots,n\end{aligned}$$

Unfortunately, the use of these monomials can lead to computational problems. As described in the next subsection, the approximation of a discrete function requires a solution to an overdetermined linear system of equations. When using non orthogonal polynomials this system becomes ill-conditioned. Moreover, it gets worse with higher orders of the polynomial. In such a case the solution of the overdetermined linear system of equations $\mathbf{V}\mathbf{x} = \mathbf{f}$ is overly sensitive to perturbations in $\mathbf{V}$ or $\mathbf{f}$. A numerical solution to an ill-conditioned system of equations is difficult (if not impossible) to determine. Additionally, the solution of large equation systems could introduce enormous round-off errors. In the worst case, higher order Taylor polynomials may oscillate around the desired values. One can improve the stability of the optimization process by choosing

number of parameters									
Order	2	3	4	5	6	7	8	9	10
Legendre	12	20	30	42	56	72	90	110	132
FOMFE	50	98	162	242	338	450	578	722	882

Table 3.1: The number of parameters given for a particular order of the polynomials. Note that Legendre polynomials allow a finer graduation for the number of parameters in comparison to Fourier series as described in [90].

the basis functions $\Phi(x, y)$ to be members of an orthogonal set. This convenience guarantees that the round off errors gets small and the optimization method becomes stable, even for large datasets and high order approximations. An additional advantage is that the resulting parameter space is Euclidean.

Wang et al. [90] propose the use of Fourier series for orientation modelling. Trigonometric functions naturally obey the property of being orthogonal. Fourier approximations substantially reduce the problem of ill conditioned equation systems, but are slow to compute and evaluate. Furthermore, they are still subject to error for higher terms. A major advantage in behalf of Legendre Polynomials is that the discretization error is minimized.

Furthermore, Legendre Polynomials are known for their fast evaluation [99] through the use of recurrence relations (Equation 3.2). In comparison to Fourier-Series [90], Legendre Polynomials allow a finer graduation of the number of parameters. See Table 3.1 for a detailed overview.

3.2 Application to Orientation Data

This section describes the model architecture. A function $f(x, y)$ can be approximated using a number of Legendre polynomials $\approx \sum_{j=0}^{n} a_j \phi_j(x, y)$ as described above. Usually, one seeks for the parameters a_j so that the equation is fulfilled. Since the number of observed data points usually is much larger than the number of parameter, this equation leads to a overdetermined system of equations. A common method for the solution of such an overdetermined equation system is the Pseudoinverse technique. This method is described in the Appendix 6.2.

In the following, we will discuss how this method can be used to model orientation data. The given (doubled) orientation is denoted with $2O(x, y)$. In digital images, the gradients direction can be extracted for the full 360 degree range. Ridge orientation, which is orthogonal to these image gradients, can be determined only up to 180 degrees. Smoothing orientation is not straightforward. Orientation vectors cannot be averaged in their local neighbourhood since opposite orientation vectors would cancel each other, even if they correspond to the same orientation. This is caused by the fact that local orientations remain unchanged when rotated for 180 degrees. Witkin and Kass [48] proposed the doubling of the orientation angle (equivalent to squaring the orientation vector, see [4,78]). After doubling the angles, opposite gradient vectors will point in the same direction and therefore will reinforce each other, while perpendicular gradients will cancel. This procedure guarantees an continues occurrence (meaning no discontinuities at zero and π) of the vectorial data and thus enables standard filtering procedures (e.g. low pass filter) to be applied on orientation data. Note also that variants of this method are used throughout the literature to model orientation data. More details are given in the Appendix.

Using the above mentioned method, one can model orientation data as described in the following:

$$\begin{aligned}\mathbf{a} &= (\mathbf{V}^T\mathbf{W}\mathbf{V})^{-1}\mathbf{V}^T\mathbf{W}\mathbf{f_1} \\ \mathbf{b} &= (\mathbf{V}^T\mathbf{W}\mathbf{V})^{-1}\mathbf{V}^T\mathbf{W}\mathbf{f_2}\end{aligned} \tag{3.4}$$

Where $\mathbf{a}$ and $\mathbf{b}$ are the parameters of a Legendre expansion of the vectorial orientation data **f1** and **f2**. The vectorial data $\mathbf{f_1}$ and $\mathbf{f_2}$ can be computed from the orientation data O using the sine and cosine function respectively, see Equation 3.10. The size of the system matrix $\mathbf{V}$ is determined by the number of coordinate points i and the number of basis functions n. $\mathbf{W} = diag(\omega_1, .., \omega_i)$ is the diagonal weighting matrix containing the weights for every coordinate.

Finally, the modelled orientations can be back converted using the following formula:

$$O(\mathbf{x}_j) = \frac{1}{2}\arctan\frac{\Phi(\mathbf{x}_j)\mathbf{a}^T}{\Phi(\mathbf{x}_j)\mathbf{b}^T} \tag{3.5}$$

As described in Section 2.3 the solution vectors $\mathbf{a}$ and $\mathbf{b}$ are coupled and influence the final orientation in a non-linear way. Errors which result from this simplification can be seen especially around SPs. This makes it very difficult to compute the parameters independently and linear as described in Equation 3.4.

The real measure for fitting a ridge orientation model to a fingerprint's orientation should be directly computed by using the orientation angle as opposed to its vectorial parts. In order to minimize the least square error (of the orientation), the following non-linear function must be minimized:

$$\underset{\mathbf{a},\mathbf{b}}{min} \quad \sum_{j=1}^{i} \omega_j \left[\arctan \frac{\Phi(\mathbf{x}_j)\mathbf{a}^T}{\Phi(\mathbf{x}_j)\mathbf{b}^T} - 2O(\mathbf{x}_j) \right]^2 \tag{3.6}$$

Where $\mathbf{a}$ and $\mathbf{b}$ are the desired parameter vectors for the Legendre approximation. The coordinates are $\mathbf{x_j}$ and i the number of points. It should be noted that Equation 3.6 is not usable, due to the discontinuity problem at 0 and π.

The problem of this discontinuity at 0 and π can be solved as described in the following. In an early paper Rao and R. Jain [76] proposed the sine as a distance measure for non-linear parameter estimation in linear phase portraits. In their optimization routine, they minimize the absolute values of this measure. Later, Ford and Strickland [25] suggested that the sum of squares of these distances should be minimized. One should note that in these references the authors intention is to use directly the orientation to obtain the model's parameters. As we carry out our optimization already in the doubled angle space, we have to half the angle for correct determination of the error functional. Then, rewriting the cost function results in:

$$\underset{\mathbf{a},\mathbf{b}}{min} \quad \sum_{j=1}^{i} \omega_j \left[\sin\left(\frac{1}{2} \arctan \frac{\Phi(\mathbf{x}_j)\mathbf{a}^T}{\Phi(\mathbf{x}_j)\mathbf{b}^T} - O(\mathbf{x}_j) \right) \right]^2 \tag{3.7}$$

3.3 Parameter Fitting

The optimization of the above mentioned functional (Equation 3.7) can only be carried out using a non-linear technique. A single non-linear optimization would consume too much time for optimization. Furthermore, such a method needs special treatment against local minima. This is the reason why we propose a hybrid optimization method which delivers accurate parameters and is still reasonable fast.

In the first optimization step we roughly approximate the model's parameters using a closed form solution (Pseudoinverse, see Appendix). In a second step, a non-linear refinement of Equation 3.7 delivers the accurate parameters. Our approach is not depending on other, prior computed data such as SPs.

3.3.1 First Step

In the first step, we propose to independently model the vectorial data ($\mathbf{f_1}$ and $\mathbf{f_2}$) of the given orientation field $O(x, y)$. To obtain the parameters, it is necessary to compute the weighted pseudo inverse $\mathbf{V}_{\mathbf{W}}^{+}$ of the system matrix $\mathbf{V}$. For details on the pseudo inverse technique we refer to the Appendix.

$$\mathbf{V}_{\mathbf{W}}^{+} = (\mathbf{V}^T\mathbf{W}\mathbf{V})^{-1}\mathbf{V}^T \tag{3.8}$$

Note that the coordinates x and y need to be normalised to the range $[-1, 1]$. The weighting matrix $\mathbf{W}$ is computed using fingerprint segmentation, $\omega = 0$ for background and $\omega = 1$ for foreground pixels. The parameter vector $\mathbf{a}$ and $\mathbf{b}$ for the sine and cosine approximation can be computed as described in the following:

$$\begin{aligned} \mathbf{a_0} &= \mathbf{V}_{\mathbf{W}}^{+}\mathbf{W}\mathbf{f_1} \\ \mathbf{b_0} &= \mathbf{V}_{\mathbf{W}}^{+}\mathbf{W}\mathbf{f_2} \end{aligned} \tag{3.9}$$

$\mathbf{f_1}$ and $\mathbf{f_2}$ contain the sine and cosine data.

$$\mathbf{f_1} = [\sin 2O(\mathbf{x}_1), \quad \sin 2O(\mathbf{x}_2), \dots, \quad \sin 2O(\mathbf{x}_i)]^T \tag{3.10}$$

$$\mathbf{f_2} = [\cos 2O(\mathbf{x}_1), \quad \cos 2O(\mathbf{x}_2), \dots, \quad \cos 2O(\mathbf{x}_i)]^T \tag{3.11}$$

3.3.2 Second step for refinement

The minimization of the cost function in Equation 3.7 is done using the Levenberg-Marquard-Algorithm (LMA).

The LMA is an iterative technique that locates a local minimum of a multivariate function that is expressed as the sum of squares of several non-linear, real-valued functions. It has become a standard technique for non-linear least-squares problems, widely adopted in various disciplines for dealing with data-fitting applications. LMA can be thought of as a combination of Steepest Descent and the Gauss-Newton method. When the current solution is far from a local minimum, the algorithm behaves like a steepest descent method: slow, but likely to converge. When the current solution is close to a local minimum, it becomes a Gauss-Newton method and exhibits fast convergence.

A LMA iteration can be given as:

$$\mathbf{c_{iter+1}} = \mathbf{c_{iter}} + (\mathbf{H} - \lambda_{iter}\mathrm{diag}[\mathbf{H}]^{-1})\mathbf{d} \tag{3.12}$$

Where $\mathbf{c}$ consists of the concatenated parameter vectors $\mathbf{c} = [\mathbf{a}, \mathbf{b}]$. $\mathbf{H}$ is the Hessian and $\mathbf{d}$ is the derivative of the cost function. Notice that while $\mathbf{d}$ is exactly the average error gradient, $\mathbf{H}$ is not the true Hessian of the function. Instead, $\mathbf{H}$ is an approximation to the Hessian which is obtained by averaging outer products of the first order derivative.

Note, that a detailed description of the LMA is beyond the scope of this thesis and we refer the interested reader to [67,51] for more extensive treatments.

We use numerical evaluations of the cost function in order to compute $\mathbf{d}$ and $\mathbf{H}$. The initial parameter $\mathbf{c_0}$ is computed using the closed form solution and provides a good starting point. Therefore the maximum number of iterations is set to 15. If the minima is detected before the maximum number of iterations has been exceeded, the algorithms quits and returns the current parameters. This process uses typically two seconds on a state of the art computer (Intel Xeon 5160, 3 GHz, Matlab 7.2, single threaded) for typical image sizes of 388x374. It should be mentioned that the computation can be carried out much faster using lower resolutions and that this task can be parallelized.

3.4 Robust Estimation

The above mentioned procedure relies on the method of least squares for estimation of the parameters in the model. Least-squares estimators assume that the noise corrupting the data is Gaussian distributed with zero mean. A common problem that is encountered in the application of least squares is the presence of outliers in the data. An outlier is an observation that is numerically distant from the rest of the data. In the worst case, even one outlier can destroy the least squares estimation, resulting in parameter estimates that do not provide useful information for the majority of the data. Robust methods have been developed as an improvement to least squares estimation in the presence of outliers.

There basically exist two approaches in order to solve the problem:

- Regression diagnostics, where certain quantities are computed from the data with the purpose of pinpointing influential points, after which these outliers can be removed or corrected, followed by an least squares analysis on the remaining cases.
- Robust regression, which tries to devise estimators that are not so strongly affected by outliers

In the following, we give a short summary of related work on robust parameter estimation procedures. This summary represents an excerpt from [35, 100]. For a more thorough overview, we refer the reader to [79].

Let

$$r_j = \sin\left(\frac{1}{2}\arctan\frac{\Phi(\mathbf{x}_j)\mathbf{a}^T}{\Phi(\mathbf{x}_j)\mathbf{b}^T} - 2O(\mathbf{x}_j)\right) \tag{3.13}$$

be the residual for the coordinate $\mathbf{x}_j)$. Then we have the following, possibilities to extract the parameters robustly:

1. **Least squares.** Least squares is the most common estimator. One noteworthy advantage of this estimator is that for common regression tasks, there exists a closed form solution through the pseudo inverse (see Appendix 6.2).

   $$\underset{\mathbf{a},\mathbf{b}}{min} \quad \sum_{j=1}^{i} r_j^2$$

   In many real-world situations, the assumption of the Gaussian noise characteristics is not relevant. Applying the least squares estimator for such data can result in completely erroneous estimates.

2. **L-estimators.** The class of L-estimators is based on linear combinations of order statistics. A widely known variant is the least absolute values regression, also known as L_1 regression. It can be determined by

   $$\underset{\mathbf{a},\mathbf{b}}{min} \quad \sum_{j=1}^{i} |r_j|$$

   Other variants are the $alpha$-regression quantile and the $alpha$-trimmed estimator.

3. **M-estimators.** M-estimators are based on the idea of replacing the squared residuals used in least square estimation by another function of the residuals r_j, yielding

   $$\underset{\mathbf{a},\mathbf{b}}{min} \quad \sum_{j=1}^{i} \rho(r_j)$$

   where ρ is a symmetric function with a unique minimum at zero. Differentiating this expression with respect to the regression coefficients yields

   $$\sum_{j=1}^{i} \psi(r_j)\mathbf{x}_j = 0$$

   where ψ is the derivative of ρ, and $\mathbf{x}$ is the row vector of explanatory variables of the jth observation. The M-estimate is obtained by solving a system of p non-linear equations. The solution is not equivariant with respect to scale. Thus, the residuals should be standardized by means of some estimate of the standard deviation σ so that

$$\sum_{j=1}^{i} \psi(r_j/\sigma)\mathbf{x}_j = 0$$

where σ must be estimated simultaneously. One possibility is to use the median absolute deviation scale estimator:

$$\sigma = Cmed(|r_i - medr_i|) \tag{3.14}$$

where $C = 1.4826$ if Gaussian noise is assumed. There exist various M-estimators in literature. Widely known are the Huber [40], Tukey and Hampel estimators. M-estimators are statistically more efficient (at a model with Gaussian errors) than L_1 regression, while at the same time they are still robust with respect to outlier values. Generalized M-estimators add the possibility to weight the different residuals.

4. **R-estimators.** R-estimators are based on the ranks of the residuals. If R_j is the rank of r_j, then the objective is to

$$\underset{\mathbf{a},\mathbf{b}}{min} \quad \sum_{j=1}^{i} a_i(R_j)r_j$$

where the score function $a_i(R_j)$ is monotone and satisfies

$$\sum_{j=1}^{i} a_i(j) = 0 \tag{3.15}$$

Some possibilities for the scores $a_i(j)$ are the Wilcoxon scores, the Van der Waerden scores, the median scores as well as the bounded normal scores [79].

An important advantage of R-estimators compared to M-estimators is that they are automatically scale equivariant.

5. **S-estimators.** S-estimators form a class of affine equivariant estimators. They are defined by minimization of the dispersion of the residuals:

$$\underset{\mathbf{a},\mathbf{b}}{min} s(r_1(\mathbf{a},\mathbf{b}), ...r_i(\mathbf{a},\mathbf{b}))$$

with final scale estimate

$$\sigma = s(r_1(\mathbf{a},\mathbf{b}),....r_i(\mathbf{a},\mathbf{b})) \tag{3.16}$$

The dispersion $s(r_1(\mathbf{a},\mathbf{b}),....r_i(\mathbf{a},\mathbf{b}))$ is defined as the solution of

$$K = \frac{1}{n}\sum_{j=1}^{i}\rho(\frac{r_j}{s})$$

Where K is often put equal to $E_{\Phi[\rho]}$. The function ρ must satisfy the following conditions. It must be symmetric and continuously differentiable, $\rho(0) = 0$ and that there exists a $c > 0$ such that ρ is strictly increasing on $[0, c]$. An example ρ-function is called Tukey's biweight function [79]. S-estimators are computationally expensive, because of the iteratively computation of non-linear equations. The main difference between M and S-estimators is that the scale is estimated a certain type of robust M-estimator.

6. **Least Median of Squares.** Least median of squares (LMS) estimation is defined by

$$\underset{\mathbf{a},\mathbf{b}}{min} \quad \underset{j}{med} \quad r_j^2$$

Some properties of the LMS estimator are:

- There always exist a solution for the LMS estimator.
- The LMS estimator is regression equivariant, scale equivariant and affine equivariant.
- High breakdown point (50%).

The main disadvantage of the LMS method is the lack of efficiency, when errors would really be normally distributed. The convergence rate of the LMS method is bad. Efficiency of the method can be improved - usually by using a LMS estimation as starting value for computing a one-step M-estimate.

7. **Least Trimmed Squares (LTS).** The LMS estimator performs poorly from the point of view of asymptotic efficiency. One possibility to repair this is to use the Least

Trimmed Squares (LTS) estimator given by

$$\underset{\mathbf{a},\mathbf{b}}{min} \quad \sum_{j=1}^{h}(r_j^2)_{j:i}$$

where $(r^2)_{1:i} \leq \cdots \leq (r^2)_{i:i}$ are the ordered squared residuals. The formula is very similar to LS, the only difference being that the largest residuals are not used in the summation. Some properties of the LTS estimator are:

- There always exist a solution for the LTS estimator.
- The LTS estimator is regression equivariant, scale equivariant and affine equivariant.
- Same high breakdown point as the LMS with the given conditions.
- The LTS has the same asymptotic efficiency at the normal distribution as the M-estimator.

The main drawback of the LTS method is that the objective function requires sorting of the squared residuals.

8. **Iteratively Reweighed Least Squares**. Iteratively Reweighed Least Squares (IRLS) is mainly used in M-estimation and should not be used with S-estimators, LMS and LTS.

   IRLS requires initialization which is usually performed with the least squares estimator. The procedure is as following:

   - Begin with a initial estimate $(\mathbf{a}, \mathbf{b})$
   - Form the residuals r_j
   - Update the estimate $\sigma = 1.4826 med(|r_j - medr_j|)$
   - Define weights

   $$w_j = \frac{\psi(r_j/\sigma)}{r_j/\sigma}$$

   - Update the estimate $(\mathbf{a}, \mathbf{b})$ by performing weighted least squares using the new weights w_j.
   - Iterate until convergence.

   Using IRLS one obtains a L_1 treatment of large residuals and a L_2 treatment of small residuals.

9. **Hough transform.** One of the oldest robust methods used in image analysis and computer vision is the Hough transform [85]. The idea is to map data into the parameter space, which is appropriately quantized, and then seek for the most likely parameter values to interpret data through clustering. A classical example is the detection of straight lines given a set of edge points. Because of its nature of global search, the Hough transform technique is robust, even when there is a high percentage of gross errors in the data. Due to runtime reasons, this technique can usually not be applied to solve problems having more than three unknowns.

   How many subsamples should be considered? In principle, one could repeat the procedure for all possible subsamples of size p, of which there are

10. **RANSAC.** The RANSAC (Random Sample Consensus) [23] is a paradigm for fitting a model to experimental data developed within the computer vision community. It is closely related to the robust estimators and the random resampling algorithm.

    The RANSAC paradigm is formally stated as follows:

    (a) Given a model that requires a minimum of p data points to instantiate its free parameters, and a set of data points A such that the number of points in A is greater than p, randomly select a subset S_1 of p data points from A and instantiate the model.

    (b) Use the instantiated model M_1 to determine the subset S_1^* of points in A that are within some error tolerance of M_1. The set S_1^* is called the consensus set of S_1.

    (c) If #(S_1^*) is greater than some threshold t, which is a function of the estimate of the number of gross errors (outliers) in A, use S_1^* to compute (possibly using least squares) a new model M_1^*.

    (d) If #(S_1^*) is less than t, randomly select a new subset S_2 and repeat the above process.

    (e) If, after some predetermined number n of trials, no consensus set with t or more members has been found, either solve the model with the largest consensus set found, or terminate in failure.

    There are three unspecified parameters: the error tolerance, the number of subsets to try, and the threshold t. The principal difference between robust estimation and RANSAC is that the robust estimators try to find the solution that minimizes some dispersion measure, but in RANSAC the objective is to find the (largest) consensus set of the observations.

The main advantage of RANSAC is its ability to do robust estimation of the model parameters, i.e., it can estimate the parameters with a high degree of accuracy even when outliers are present in the data set.

A disadvantage of RANSAC is that there is no upper bound on the time it takes to compute these parameters. When an upper time bound is used (e.g. a maximum number of iterations) the solution obtained may not be the optimal one. Another disadvantage of RANSAC is that it requires the setting of problem-specific thresholds.

The number of iterations n can be computed using the following formula :

$$n = \frac{log(1-z)}{log[1-(1-\epsilon)^m]} \tag{3.17}$$

Where z is the confidence level, m is the number of model parameters to be estimated and ϵ is the outlier proportion. It should be noted that for models with a large number of parameters, the number of iterations becomes impractically high for common confidence levels. e.g for 56 parameters, a confidence level of $z = 0.95$ and an outlier proportion of $\epsilon = 0.2$ the number of iterations becomes 80000.

11. **Outlier diagnostics.** Outlier diagnostics are statistics that focus attention on observations having a large influence on the least squares estimators. Several diagnostic measures have been designed to detect individual cases or groups of cases that may differ from the bulk of the data.

    Outlier diagnostics is often based on observing the diagonal elements of the LS projection matrix H also called the hat matrix

    $$H = X(X^TX)^{-1}X^T$$

    which transforms a observed vector y into its least square estimate

    $$\hat{y} = Hy$$

    The diagonal elements h_{ii} of the hat matrix can reveal the leverage points in the data. When h_{ii} is large, that is near 1, then variance of the ith residual is almost zero. The main cause for this may be that the case in question has an unusually large influence on the LS regression coefficients.

3.4.1 Proposed Robust Method

In order to remove the sensitivity to outliers, we propose a robust estimation technique which is described below. Apparently, the worst possible outlier in orientation data is one which is perpendicular to the models estimation (90 degree). This is in contrast to other scenarios, where an outlier can be arbitrarily away from the model estimation. Therefore replacing the L_2 norm and using other robust norms (e.g. L_1) may not significantly improve the parameter extraction.

The non robust method uses least squares. The residuals are

$$\underset{\mathbf{a},\mathbf{b}}{min} \quad \sum_{j=1}^{i} r_j^2$$

The proposed method computes the residuals r_j as the sine of the differences of the model compared to the observed value.

$$r_j = \sin\left(\arctan \frac{\Phi(\mathbf{x}_j)\mathbf{a}^T}{\Phi(\mathbf{x}_j)\mathbf{b}^T} - 2O(\mathbf{x}_j) \right) \tag{3.18}$$

This Equation 3.18 contains the sine-function as cost function. In order to minimize the influence of outliers, we propose the use of $sin(2\alpha)$ function instead of the $sin(\alpha)$ function (see Figure 3.2). This change in the equation is very important, as overly large residuals (90 degree outliers) yield in small values. The final robust energy functional can be derived from equation 3.7, and can be given as:

$$\underset{\mathbf{a},\mathbf{b}}{min} \quad \sum_{j=1}^{i} \omega_j \left[\sin\left(\arctan \frac{\Phi(\mathbf{x}_j)\mathbf{a}^T}{\Phi(\mathbf{x}_j)\mathbf{b}^T} - 2O(\mathbf{x}_j) \right) \right]^2 \tag{3.19}$$

Unlike other robust functions from literature, there are no additional added costs for the computation.

3.5 Evaluation

In order to evaluate the proposed ridge orientation model, we conduct a large number of experiments.

This section is structured as described in the following. The first two subsections 3.5.1 and 3.5.2 define the datasets and the performance measures for the evaluations. The optimal number of iterations for the non-linear optimization, as well as the optimal order of Legendre polynomials is determined in the experiments described in subsection

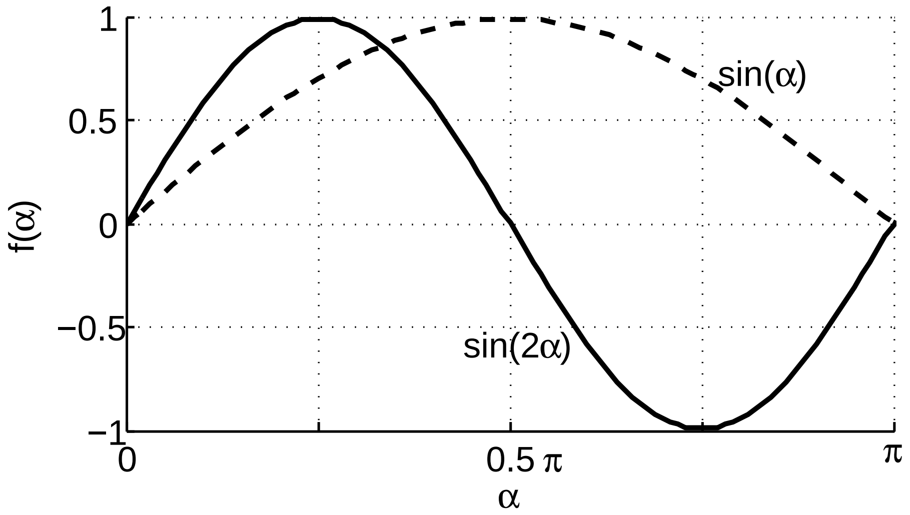


Figure 3.2: Two cost functions. non-robust $sin(\alpha)$ and robust $sin(2\alpha)$. Orientation π and zero result in the same cost. The robust cost function $sin(2\alpha)$ eliminates the influence of outliers.

3.5.3. Exemplary results are given in Subsection 3.5.4. In Subsection 3.5.5 we examine the SP detection capability of the proposed method in comparison to other methods. Note that all the SPs in the images have been manually labelled. In subsection 3.5.6 we examine how the proposed methods affects the minutiae matching performance of a commercially available fingerprint matcher. Subsection 3.5.7 compares the proposed robust parameter fitting method to a normal least squares method.

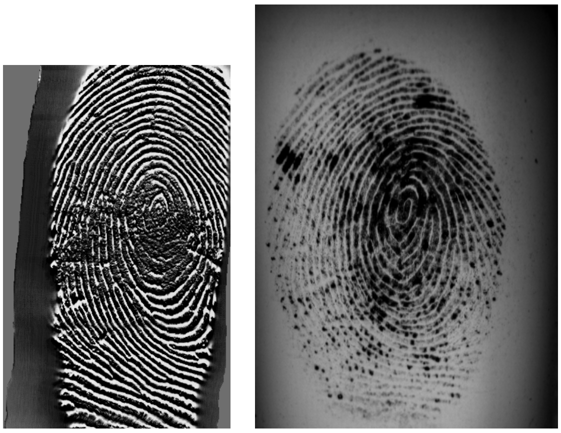
3.5.1 Datasets

We use the following two publicly available databases (shown in Figure 3.3) for our experiments:

- FVC2004 DB3A [62]: a database containing 800 fingerprints of low quality. These images have been created by a low cost thermal sweeping sensor (FingerChip FCD4B14CB by Atmel). Most of the included images contain low quality regions, thus the ridge orientation interpolation ability can be tested on this database.

- FVC2006 DB2A [20]: a database containing 1680 fingerprints recorded using an optical sensor. The population is more heterogeneous and the database includes fingerprint images of manual workers and elderly people. In order to reduce the computational load, we used the first 67 fingerprints for our experiments. Using the available 12 impressions per fingerprint this results in 804 images.

Raw orientation estimation as well as the segmentation into foreground/background is done using a industrial fingerprint matcher from Siemens (Siemens IT Solutions and Services, Biometrics Center).



(a) *FVC2004, db3a* (b) *FVC2006, db2a*

Figure 3.3: Typical examples of images in the two databases used for evaluation. FVC2004 [62], database 3 (Figure 3.3(a)) consists of 800 images sized 300x480 pixels. The FVC2006 [20] database 2 dataset (Figure 3.3(b)) contains 1680 images of fingerprints with the size 400x560. In order to reduce the computational load, we used 804 images of this dataset.

3.5.2 Performance Measures

Singular Point Detector

For measuring the performance of a SP detection algorithm, the two quantities of interest are clearly the number of correct detections and the number of spurious detections. Obviously, the ultimate goal is to maximize the number of correct detections and to minimize wrong detections.

Unfortunately, there is no established 'standard' in literature for evaluating SP detection approaches. Although most authors give true positive and false positive numbers, there seems to be no consensus on how large the threshold should be, on which this decision is based on. In this publication, we intend to vary this threshold and display the precision and recall figures as a function of it. The definition of the performance measures is given in the following:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3.20}$$

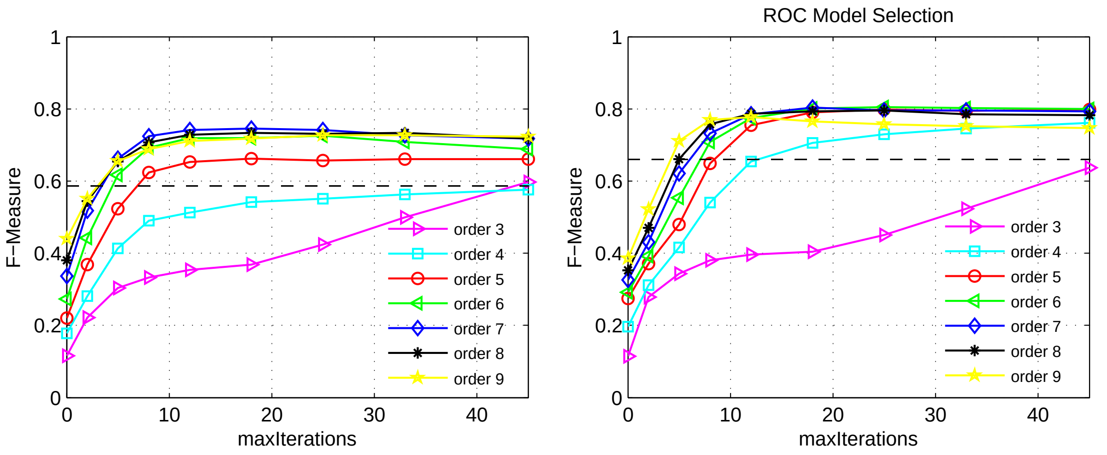
$$\text{Precision} = \frac{TP}{TP + FP} \tag{3.21}$$

$$\text{F-Measure} = \frac{2 \cdot \text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}} \tag{3.22}$$

Where TP=true positive, FN=false negative and FP=false positive. The first quantity of interest, namely, the proportion of SPs that are detected, is given by the recall. The second quantity of interest is the number of correct detections relative to the total detections made by the system is given by the precision. Note that we are also interested in the threshold parameter that achieves the best trade-off between the two quantities. This will be measured by the F-measure which summarizes the trade-off between recall and precision, giving equal importance to both.

Fingerprint Matcher

The final performance of a fingerprint matcher is assessed by two indices: False Acceptance Rate (FAR) and the False Rejection Rate (FRR). The FAR index is defined as the general percentage of an imposter being falsely accepted by the system. On the other hand, the FRR index is the percentage of how many genuine users are falsely rejected. The System performance is reported by a Receiver Operating Characteristic (ROC) curve plotting FAR versus FRR at various matching thresholds. Besides the above ROC curve there also exist some indices which summarize the accuracy of a verification system (see [44]). Most prominent is the Equal-Error-Rate (EER), which will be reported in our experiments. The EER denotes the error rate at a threshold for which FRR and FAR are identical (FRR=FAR). In order to conduct the experiments we match every fingerprint with every other fingerprint (except itself) in the database. The matches are carried out in a symmetric way, eg. A is matched with B and B is matched with A. For a database of 800 images with 100 fingerprints and 8 impressions per fingerprint this results in $799 * 800 = 639200$ matches. From these matches $7 * 800 = 5600$ are genuine pairs and $8 * 99 * 800 = 633600$ imposter pairs. Note that these figures are slightly different for the FVC2006db2a dataset, since we use 804 images here. Note that this choice was done in order to reduce the runtime necessary for matching all fingerprints with each other in the database. The computed statistical significance level (see Section 3.5.8) suggests that this is a valid choice.



(a) *FVC2004db3a, SP detection results*

(b) *FVC2006db2a, SP detection results*

Figure 3.4: Model Selection using the FVC2004db3a and FVC2006db2a dataset. SP detection is done on the smoothed OF using a PI based method. The models order has been varied from 3 to 9. The maximum number of iterations are varied from zero to 33. The F-Measure is computed for a threshold of 15 pixels (FVC2004db3a) and 20 pixels (FVC2006db2a) respectively. The black dotted line shows results on the original OF.

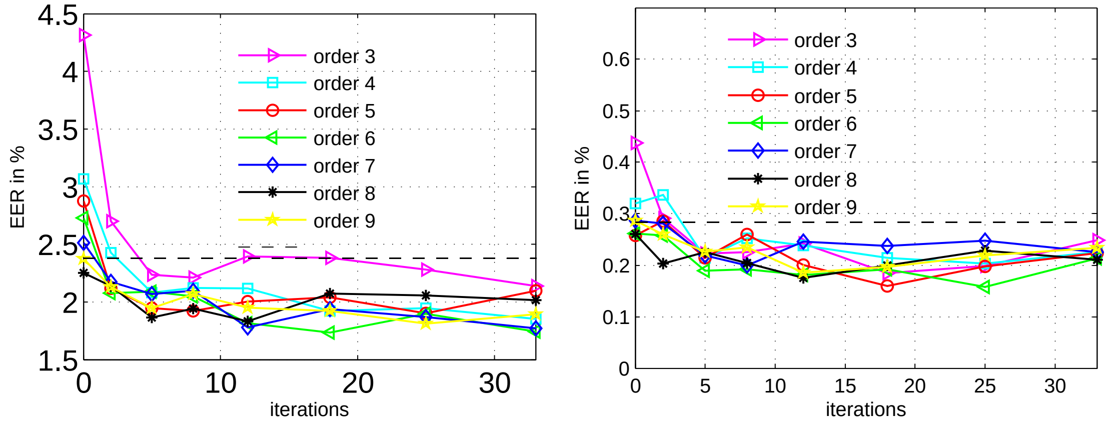
3.5.3 Model Selection

Formally, model selection is the task of selecting a statistical model from a set of potential models, given the data. In our particular case we have to find the optimal number of parameters (order of the Legendre polynomial) for the proposed method. The measure for determining the best number of parameter (goodness of fit) is based on both, the singular point detection as well as on the matching quality. More details and and a formal definition of model selection is given in the Appendix.

Model Selection using Singular Points

Here we conduct an experiment in order to determine SP detection capability of the proposed model. We vary the polynomial order from 3rd to 9th order. The maximum number of iterations for the LMA is varied from zero to 33. The F-Measure is computed using a threshold of 15 pixels (FVC2004 db3a) and 20 pixels (FVC2006 db2a) respectively. This value is adjusted to twice the typical ridge valley distance in both databases. Figure 3.4 shows the results for these two datasets.

As can be seen Legendre polynomials with order of 6-8 score the best results. Furthermore, it can be seen that there is only a mere improvement after 15 iterations.



(a) *FVC2004db3a, Matching results*

(b) *FVC2006db2a, Matching results*

Figure 3.5: Model Selection using the FVC2004db3a and FVC2006db2a dataset. This plot shows how the EER of the fingerprint matcher varies versus the number of iterations. This evaluation is done using different orders of the Legendre polynomials. The black dotted line shows results using the original OF.

Model Selection using Minutiae

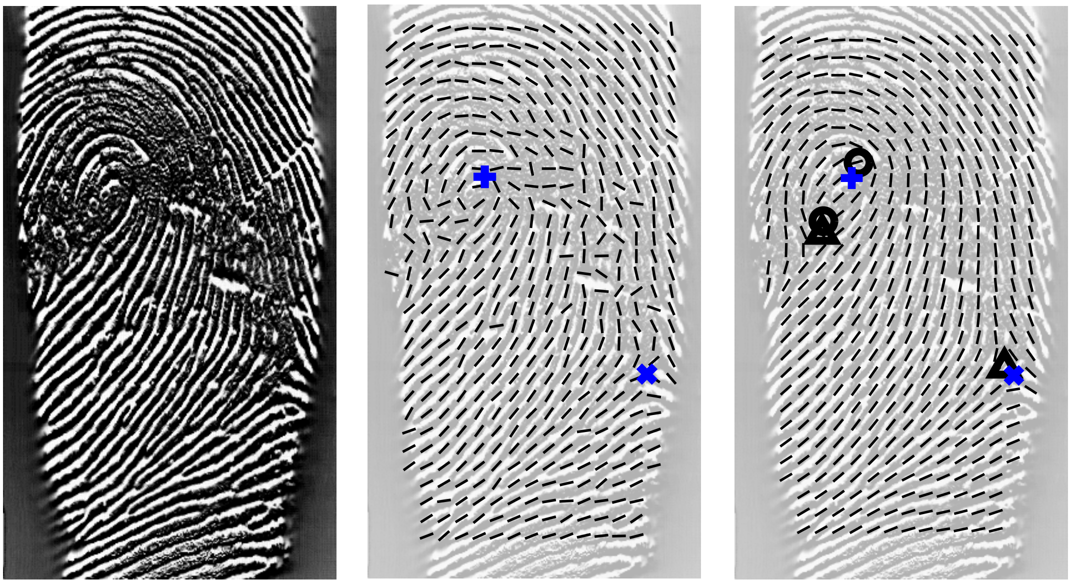
In order to determine the best order for the model we conduct fingerprint matching. Again, the polynomial order is varied from 3 to 9. The maximum number of iterations for the LMA is varied from zero to 33. On the y-axis we plot the EER. Similar to the SP detection results, Legendre polynomials with orders 6-8 result in the lowest error rates. Note the black dotted line, which represents the results from the original OF of the matcher.

This experiment shows, that the order of the polynomial is not a critical parameter. Polynomials with order from 5 - 8 give very good results. Furthermore, on can see that the non-linear approach gives good results after 5 iterations already.

3.5.4 Exemplary Results

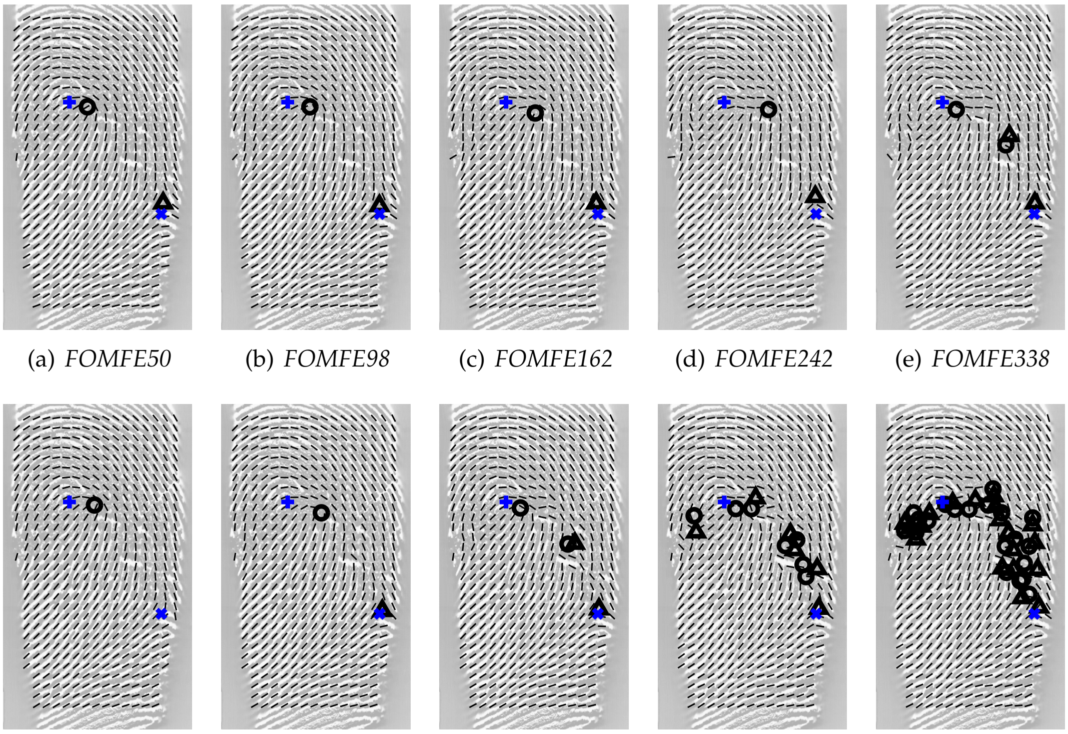
In Figure 3.7 the orientation smoothing capability of different methods is illustrated. The image shown is of Central Pocket Loop type. This image contains both, high curvature areas (tiny whorl) and noise, making it the ideal benchmark for orientation smoothing algorithms. Methods which independently smooth the vectorial data, will be plagued by the Bias-Variance Trade-off [6]. Using our method we can circumvent this problem to some extend.

Ridge orientation 3.6(b) has been extracted using a gradient based method from the original image 3.6(a). Subfigures 3.7(j)-3.7(f) show the orientation field smoothed using Gaussian convolution. While low smoothing ($sigma = 5, 8$) causes wrong detections, too much smoothing ($sigma = 17, 25$) usually causes a smoothing of the high curvature area and results in a poor overall orientation field. Subfigures 3.7(a)-3.7(e) show the orientation modelling capability of the FOMFE approach [90]. In Figure 3.7(a) and 3.7(b) we use a low order trigonometric polynomial (k=2 and k=3, resulting in 50 and 98 parameters). The reconstructed orientation in the noise affected area is not satisfactory. In Figure 3.7(c) the optimal number of 162 parameters is used and in Figure 3.7(d) and Figure 3.7(e) we used 338 parameters for describing the orientation. Note that in all cases wrong singularities are detected. In Subfigure 3.6(c) the computed orientation of our proposed method is shown. Emphasize should be paid on the correct reconstruction of the orientation in the noise affected region and to the accurate determination of the SPs position.



(a) *original image* (b) *extracted* (c) *proposed56*

Figure 3.6: **Orientation estimation of a Central Pocket Loop type pattern.** In Subfigure 3.6(c) the proposed method (6th order Legendre polynomial, resulting in 56 parameters) after 15 iterations is shown. It should be emphasized that our approach was able to reconstruct the fine details (eg. small whirl) while being able to eliminate the influence of noise.



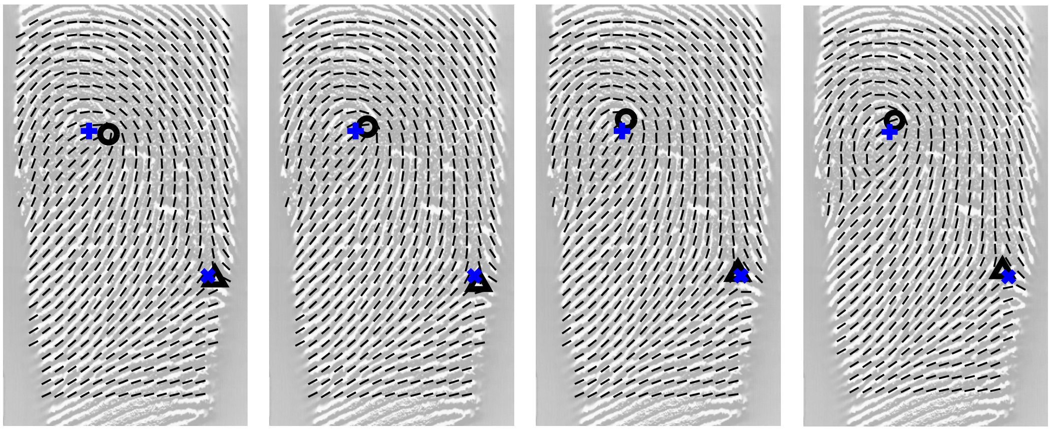
(a) *FOMFE50* (b) *FOMFE98* (c) *FOMFE162* (d) *FOMFE242* (e) *FOMFE338*

(f) *Gaussian*, $\sigma = 25$ (g) *Gaussian*, $\sigma = 17$ (h) *Gaussian*, $\sigma = 12$ (i) *Gaussian*, $\sigma = 8$ (j) *Gaussian*, $\sigma = 5$

Figure 3.7: **Illustration of the Bias-Variance Dilemma in standard orientation estimation techniques.** Starting from the gradient based orientation estimation shown in Figure 3.6(a), the orientation is smoothed using the FOMFE approach [90] (first row) and using a convolution based smoothing as described in [4] (named GAUSSIAN here). In the first row the approximation capability of the FOMFE approach can be seen. The order of the Fourier polynomials have been changed from k=2 to 3,4,5 and 6. This results in 50, 98, 162, 242 and 338 parameters. The images are labelled using this number. In the second row, the smoothing capability of Gaussian convolution based methods is demonstrated. Here we varied σ between 5, 8, 12, 17 and 25. Note that using these two approaches it was not possible to accurately estimate the ridge orientation of the shown FP.

3.5.5 Singular Point Detection Results

In this subsection the results from a Poincare Index (PI) [44] based SP detector are given. We compare the proposed method with two other OF estimation methods available in literature [90,4]. Both methods are varied between heavy smoothing, normal smoothing (values as proposed by the authors) and low smoothing. The first method is the Fourier expansion [90] of a OF denoted as FOMFE. Here we vary the order of the two dimen-



(a) *start* (b) *iteration 1* (c) *iteration 5* (d) *iteration 8*

Figure 3.8: Illustration of the second optimization step (using the same input data as shown Figure in 3.7). In (a) the output of the closed form optimization can be seen. In (b), (c) and (d) it is illustrated how the non-linear method converges towards the final solution, which is shown in Figure 3.6(c).

sional Fourier series between 2,4 and 6 (resulting in 50, 162 and 338 parameters). The second method is the smoothing of the orientation using a Gaussian convolution mask. This method is described by Bazen and Gerez in [90,4] and denoted as 'Gaussian'. The parameter σ is varied between the values 5, 12 and 25.

The results, obtained using the manually labelled position of SPs, are given in Figure 3.9 for the FVC2004 db3a and in Figure 3.10 for the FVC2006 db2a database. The results can be summarised as in the following.

- **low smoothing using conventional methods:** Recall rates are very good, mainly due to the fact that no real SPs are 'smoothed out' using low smoothing. Note that noisy OFs contain many random detections which will be counted as true positive and hence 'improve' the recall rates. On the other hand, low smoothing results in poor precision rates, meaning many wrong detections are present. Both this findings can be observed for the two datasets, depicted in Figure 3.9. In defence of the FOMFE approach, one should mention that it yields much better precision rates than simple Gaussian smoothing.

- **heavy smoothing using conventional methods:** ideally, one would expect that more smoothing results in better precision figures and in worse recall figures. Unfortunately, this findings can not be observed in practice, since heavy smoothing shifts the positions of core type SPs beyond any usable threshold and destroys all high curvature information. Therefore, heavy smoothing results in bad precision and bad recall numbers. For deltas, it can be observed that heavy smoothing does not result in as bad figures.

- **best trade off using conventional methods:** The best trade off between the two above mentioned scenarios is a medium smoothing ,denoted with 'FOMFE162' and 'Gaussian $\sigma = 12$'. These values have been suggested by the corresponding authors [90,4].

- **proposed method:** The proposed method has very high precision and recall figures. The trade off, smoothing artefacts while preserving high curvature areas, has been significantly improved for the FVC2004db3a in comparison to other methods. We can report an increase of 10% for the F-Measure.
  For the FVC2006db2a database there is no significant improvement to SP detection in comparison to competing methods. This is due to the fact that this database is of very high quality and therefore there is no improvement in smoothing (also reflected by the very low EER of the matcher).

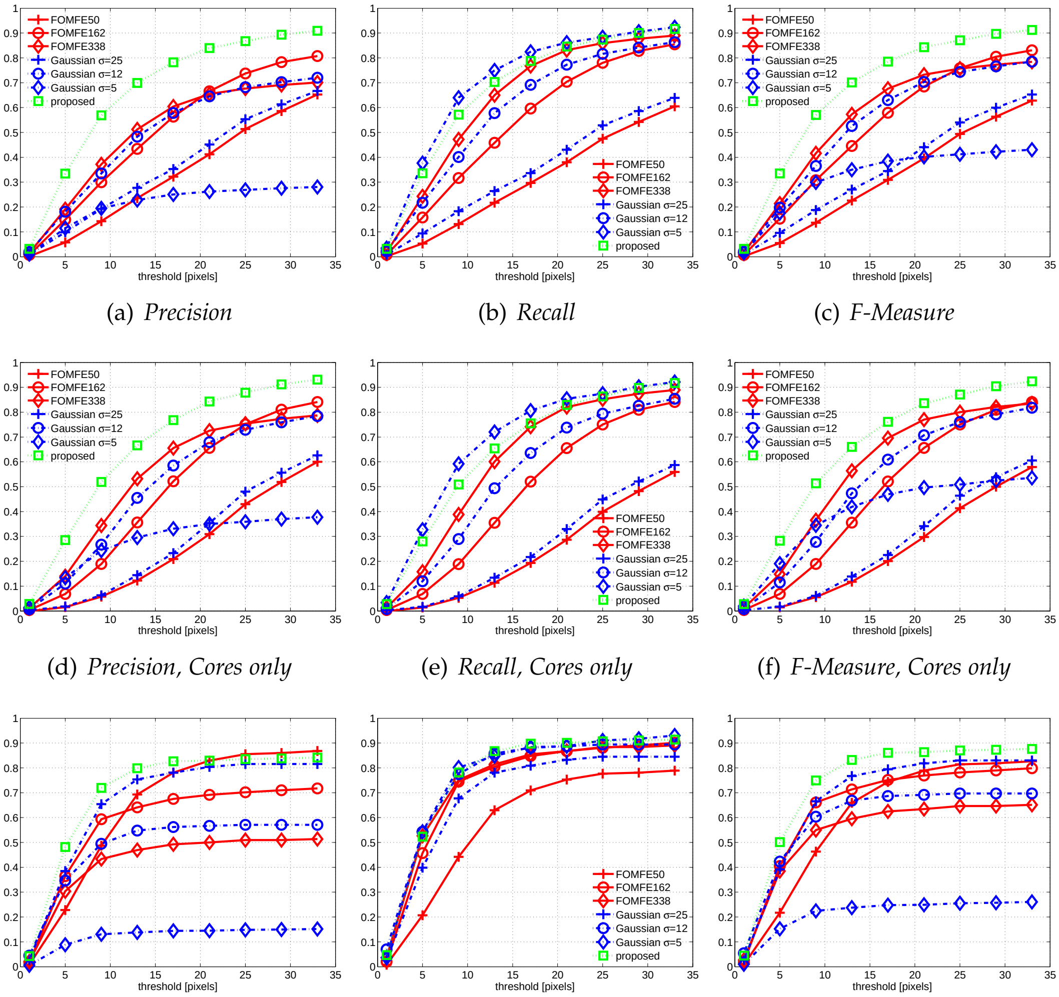
3.5.6 Matcher Results

In this subsection we present the fingerprint matching results using the industrial matcher from Siemens IT Solutions and Services, Biometrics Center. Everything else being equal, we only replace the original orientation processing/smoothing method. The matcher uses the following features for matching two fingerprints [20] : Minutiae, Ridges or Pseudo-Ridges, OF, Raw fingerprint image or parts of it and the global ridge frequency. Please note that the results presented here are not directly comparable with the results published in [62] and [20] since the number of imposters is much smaller in this competitions, mainly due to computational reasons.

The results (shown in Figure 3.11(a) and 3.11(b)) can be summed up as described in the following:

- For the FVC2004db3a the EER of the original matcher is 2.38%. Using the FOMFE approach this measure can be reduced to 2.28%. Using the proposed method we can reduce the EER to 1.81%, a relative improvement of 24%.

- For the newer database FVC2006db2a the original matcher has an EER of 0.284%. The FOMFE based OF results in an EER of 0.284%. In comparison we can achieve an EER of 0.18% using the proposed method. This is a relative improvement of 37%. In conclusion, the use of this database shows a good generalisation ability of the proposed algorithm.



(a) *Precision*

(b) *Recall*

(c) *F-Measure*

(d) *Precision, Cores only*

(e) *Recall, Cores only*

(f) *F-Measure, Cores only*

(g) *Precision, Deltas only*

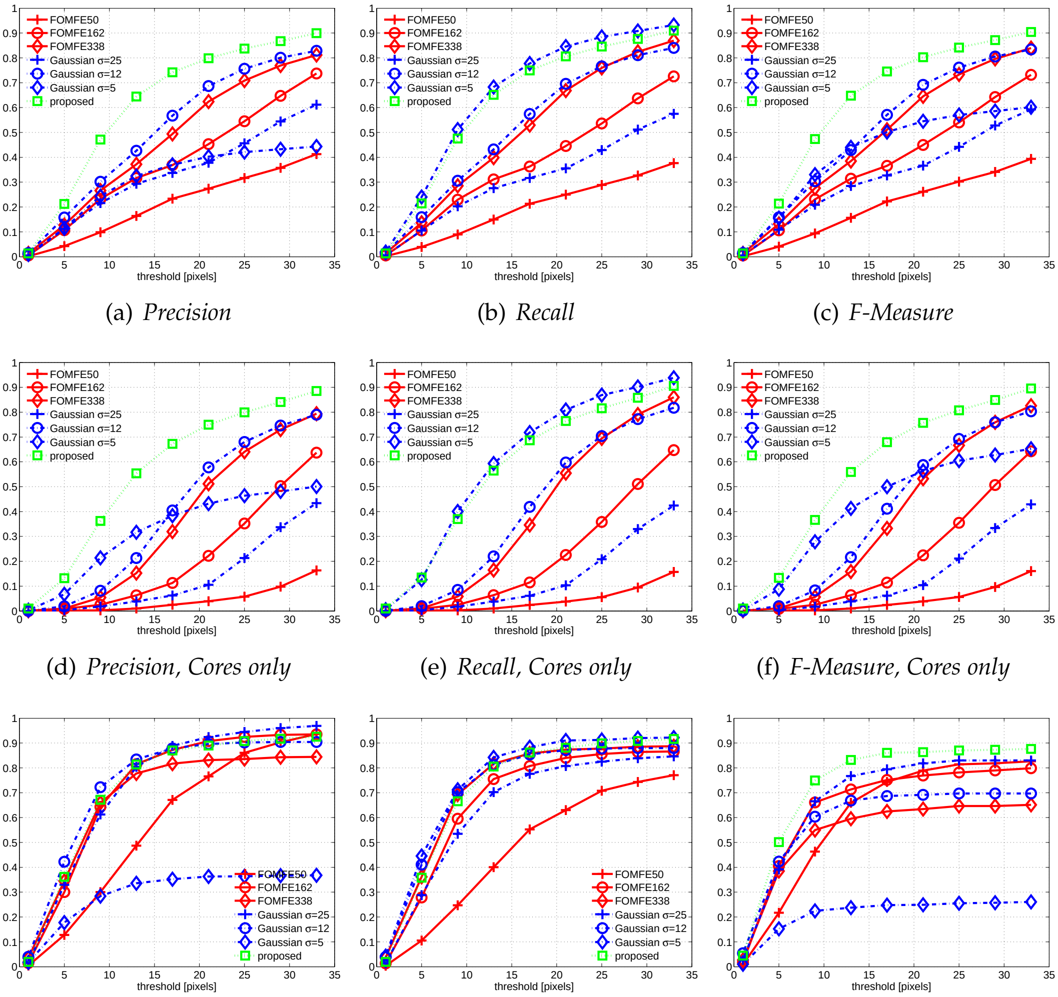
(h) *Recall, Deltas only*

(i) *F-Measure, Deltas only*

Figure 3.9: SP detection results using the FVC2004 database 3a. Recall, Precision and the F-Measure evaluated for different Fingerprint Ridge Orientation smoothing approaches. On the y-axis, the threshold for obtaining the mentioned performance measures is given.

3.5.7 Robust versus Non-Robust Parameter Estimation

In this subsection we will give a small comparison of the robust parameter fitting method with the classic least squares method. Both, the non robust fitting method, as well as the robust fitting method use polynomials of 6th order (50 parameters) and use 15 iterations for the non-linear parameter estimation. As can be seen from Figure 3.16 and 3.15 the robust parameter estimation has a significant impact on the matching performance. For the FVC2006db2a dataset, the EER can be decreased from 0.26 to 0.18



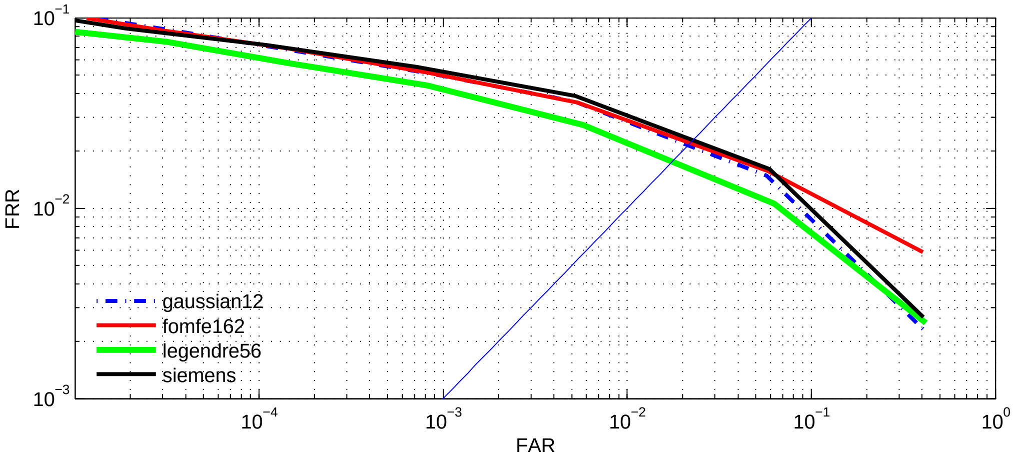
(a) *Precision* (b) *Recall* (c) *F-Measure*

(d) *Precision, Cores only* (e) *Recall, Cores only* (f) *F-Measure, Cores only*

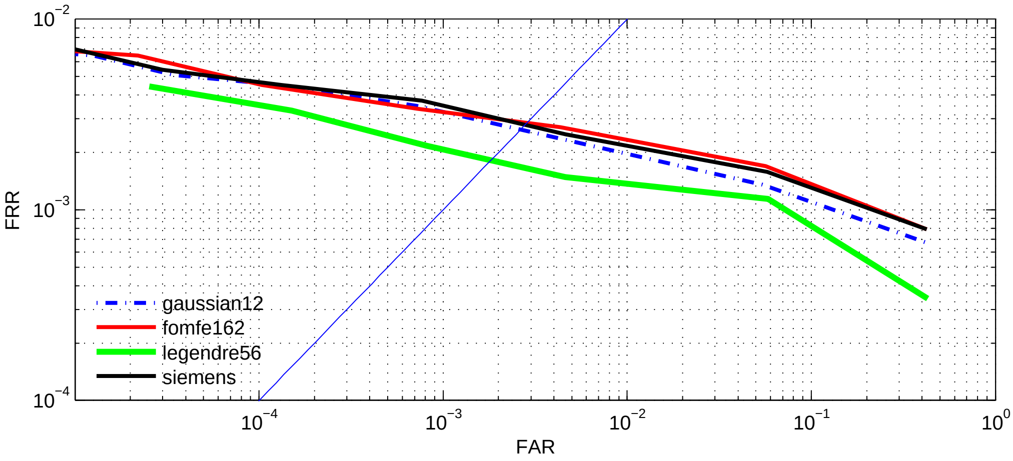
(g) *Precision, Deltas only* (h) *Recall, Deltas only* (i) *F-Measure, Deltas only*

Figure 3.10: SP detection results using the FVC2006 database 2a. Recall, Precision and the F-Measure evaluated for different Fingerprint Ridge Orientation smoothing approaches. On the y-axis, the threshold for obtaining the mentioned performance measures is given.

%. The robustness has a smaller impact on the FVC2004 db3a dataset, where the EER can be reduced from 1.96% to 1.81% using the robust algorithm. The robust version can only gain little in performance. The main reason for this fact is that the orientation data in the mentioned databases contains only a small amount of Gaussian noise. This represents the best case for a Least-Squares Estimator. In any other cases (e.g. many regions with biased orientation data caused by ghosting, scars or crinkles) a robust parameter



(a) *FVC2004db3a*

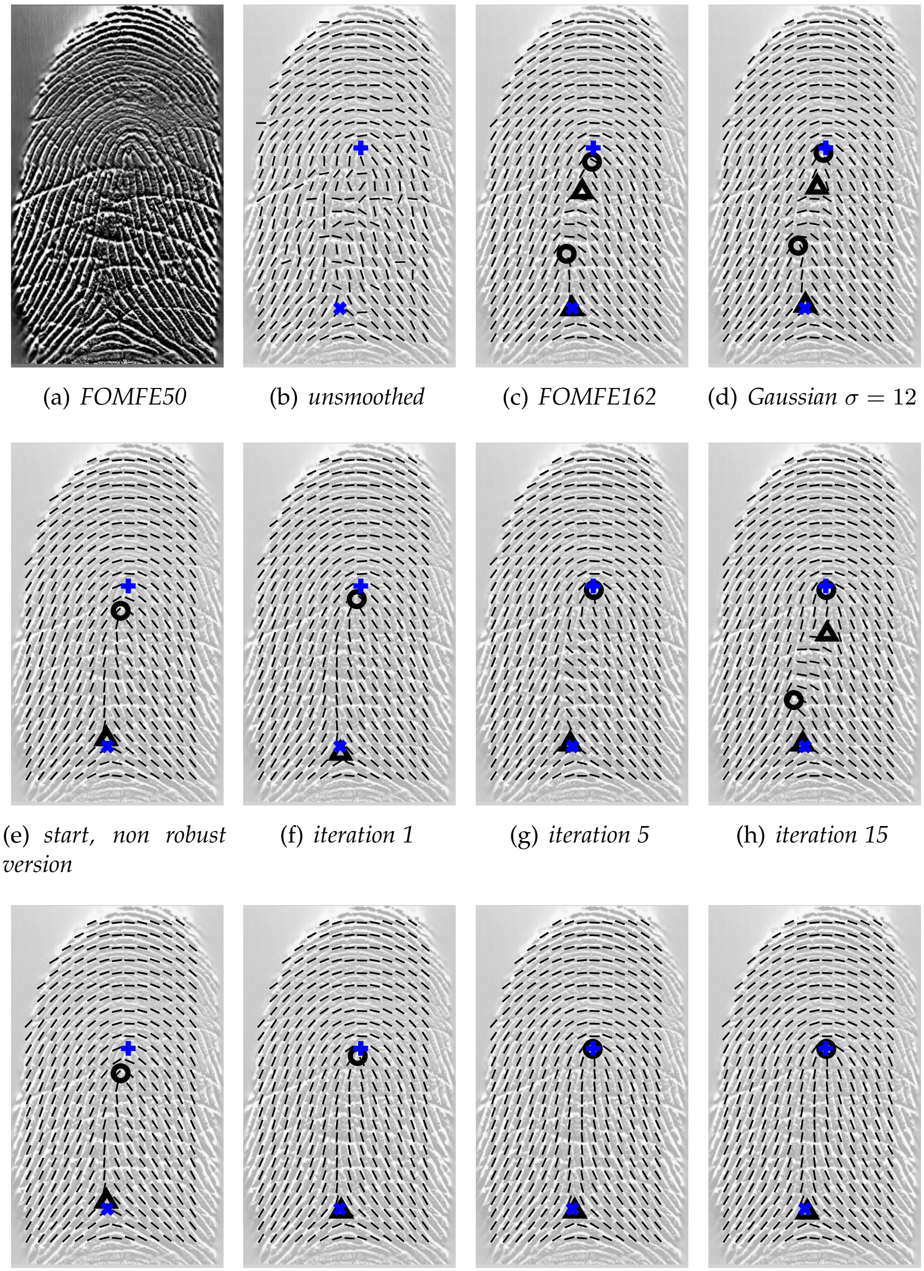


(b) *FVC2006db2a*

Figure 3.11: ROC Curves, using commercial matcher. For the FVC2004db3a the EER of the original matcher is 2.38%. Using the FOMFE approach this measure could be reduced to 2.28%. Using the proposed method we can reduce the EER to 1.81%, a relative improvement of 24%. For the FVC2006db2a database the original matcher has an EER of 0.284%. The FOMFE based OF results in an EER of 0.284%. In comparison we can achieve an EER of 0.18% using the proposed method. This is a relative improvement of 37%.

estimation will give significant better results.

In Figure 3.12 a visual example comparing the non-robust version with the robust parameter fitting method is given. As can be seen, the robust version is persistent to the occurring outliers.

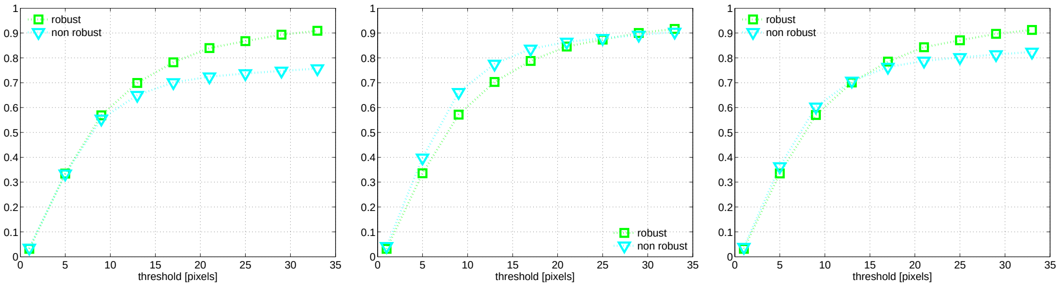


(a) *FOMFE50* (b) *unsmoothed* (c) *FOMFE162* (d) *Gaussian* $\sigma = 12$

(e) *start, non robust version* (f) *iteration 1* (g) *iteration 5* (h) *iteration 15*

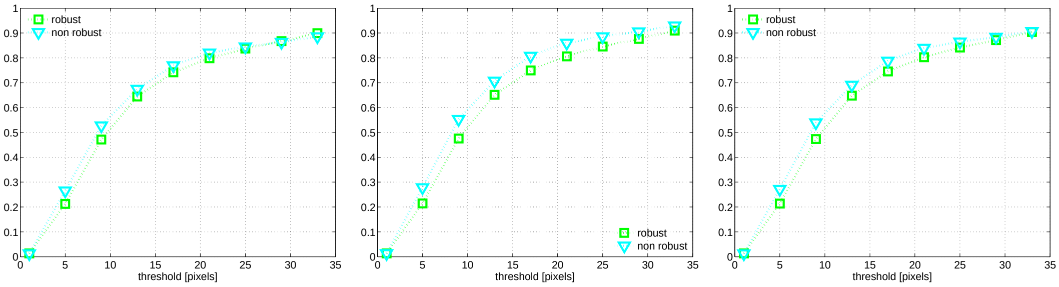
(i) *start, robust version* (j) *iteration 1* (k) *iteration 5* (l) *iteration 15*

Figure 3.12: **Robust vs non robust parameter estimation.** The image in 3.6(a) results in a noisy orientation field (seen in 3.6(b)). The distribution of the noise is non-Gaussian, representing bad data for classic least squares algorithms. Subfigure 3.12(c) and 3.12(d) show the results of the FOMFE model and of the smoothing capability of a Gaussian low pass. The second row shows our approach with a normal (non robust) least squares parameter estimation technique. In the third row, one can see the proposed robust Legendre polynomial fitting. Again, we used a 6th order Legendre polynomial and 15 iterations for the non-linear fitting.



(a) *Precision* (b) *Recall* (c) *F-Measure*

Figure 3.13: **SP detection using FVC2004db3a:** Robust vs non robust parameter fitting.



(a) *Precision* (b) *Recall* (c) *F-Measure*

Figure 3.14: **SP detection using FVC2006db2a:** Robust vs non robust parameter fitting.

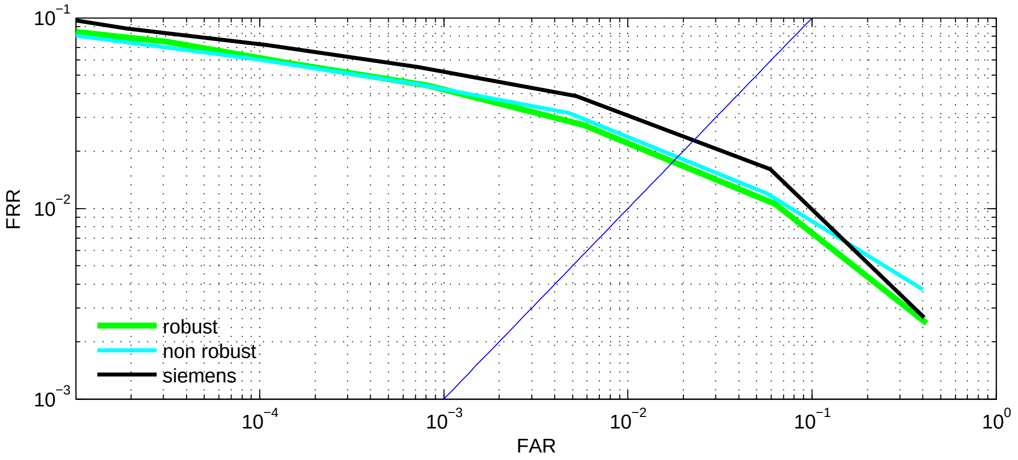


Figure 3.15: FVC2004db3a. ROC comparing the non robust with the proposed robust parameter fitting method. The EER decreases from 1.96% to 1.81%.

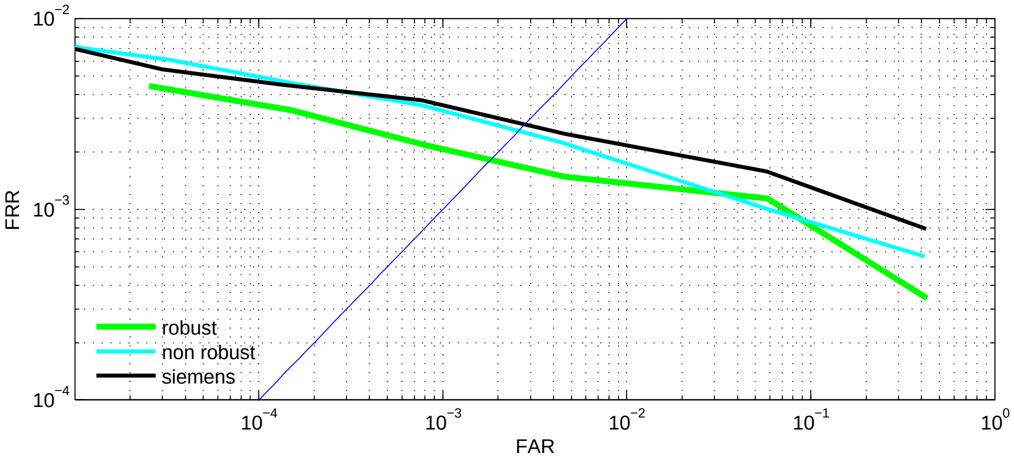


Figure 3.16: FVC2006db2a. ROC comparing the non robust with the proposed robust parameter fitting method. EER decrease from 0.26 to 0.18 %.

3.5.8 Statistical significance of experimental results

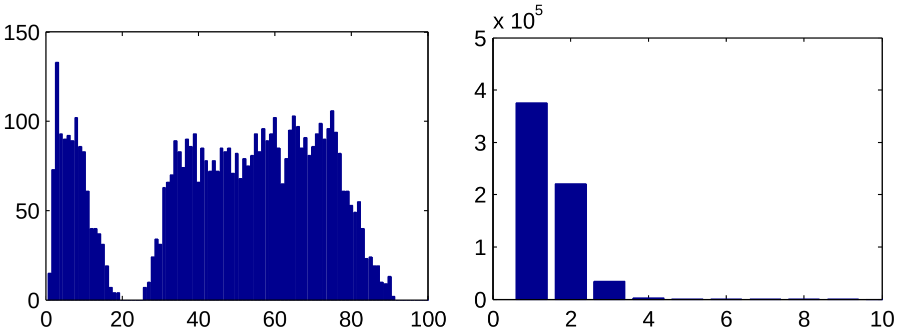
This subsection evaluates the statistical significance of the achieved results.

Guyon et al. [33] address the problem of determining the size of the testing samples to guarantees statistical significance in a recognition task. It is assumed that the errors of both evaluations are distributed according to the binomial law, which is approximated by the Normal law. For the estimation of the test set size which allows comparing the performance of two 'recognizers', Guyon [33] states:

$$n = \left(\frac{z_\alpha}{\beta}\right)\frac{2}{p} \tag{3.23}$$

where $p = (p_1 + p_2)/2$ and $\beta = (p_2 - p_1)/p$. p_1 and p_2 are the empirical error rates of the two evaluations to be compared. z_α represents the z-value of the given confidence level α and can be obtained from the Normal (Gaussian) distribution.

Wu and Wilson [98] from the National Institute of Standards and Technology (NIST) illustrated that the discrete probability distribution functions of the match and non-match similarity scores, generated by using fingerprint matching algorithms on data sets, have no definite underlying distribution functions. See Figure 3.17 for the genuine and imposter match score distributions of the FVC2004db3a dataset. Clearly, it can be seen that the distributions are non Gaussian. As a consequence, the mentioned approach of Guyon [33] et al. is not feasible here. Instead a non-parametric approach (as desribed in [98]) must be employed in the analysis of the fingerprint similarity scores.



(a) *genuine* (b) *genuine*

Figure 3.17: **FVC2004db2a.** Genuine and imposter score distributions.

Furthermore Wu and Wilson [98] argue that although the sizes of fingerprint data sets are much larger than the sizes of the data sets that are dealt with in the medical practice, the same significance evaluations [34] can be performed on fingerprint matcher statistics.

One of the most significant argument in the mentioned paper [98] is that the area under an ROC curve stands for the probability that the score obtained for the genuine match is higher than the score assigned for the impostor match given both genuine match and impostor match assuming the score is a continuous random variable. Therefore, to evaluate a fingerprint matching algorithm, an ROC curve as a whole rather than an ROC curve at a specific point or within a chosen region should be taken into account. Even if a part of an ROC curve produces lower error values, this does not guarantee that the ROC curve as a whole is better.

Furthermore, the variance of the area under an ROC curve can be obtained by calculating the variance of the Mann–Whitney statistic. In addition, the Mann–Whitney statistic is asymptotically normally distributed regardless of the distributions of the match and non-match similarity scores thanks to the Central Limit Theorem. Thus, the Z statistic can be formulated. The Z statistic can be computed in a conservative way depending on how to deal with the correlation coefficients. Therefore, the Z statistic hypothesis test offers a systematic way to detect the statistical significance of differences between two underlying ROC curves, namely, differences between performances of two fingerprint matching algorithms. The method investigated in [98] provides the information on if the model based OF produces better results, as well as the information about whether the difference is real or just by chance at a quantified significance level.

The straightforward way to test the significance of the difference between two areas under ROC curves is the Z-test. The Z statistic is defined as the difference of two areas divided by the square root of the variance of two-area difference [34], and it is subject to the standard normal distribution with zero expectation and a variance of one. The Z statistic can be expressed as:

$$Z = \frac{A_1 - A2}{\sqrt{SE^2(A_1) + SE^2(A_1) - 2 * r * SE(A_1) * SE(A_2)}} \tag{3.24}$$

where A_1 and A_2 are two areas under ROC the curves, $SE(A_1)$ and $SE(A_2)$ are two standard errors of these areas, respectively. r is the correlation coefficient between two areas. In order to estimate r it is necessary to compute the Kendall τ correlations between the paired ratings C_{IMP} (imposters) and C_{GEN} (genuine). r can be estimated by computing the mean of the correlations $r = (r_{IMP} + r_{GEN})/2$ and reading out the corresponding entry of the table given in [34].

The computed values for the FVC2004 db3a and for the FVC2006 db2a are listed in Table 3.2 and 3.3. We computed the Kendall τ correlations C_{GEN} and C_{IMP} using the Matlab statistical toolbox. It should be noted that for the given large number of imposter

database	A_1	A_2	$SE(A_1)$	$SE(A_2)$
FVC2004db3a	0.9950	0.9963	0.0007	0.0006
FVC2006db2a	0.9979	0.9988	0.0003	0.0003

Table 3.2: The areas under the ROC curve and the corresponding standard errors.

database	C_{gen}	C_{imp}	C	r	Z	$p_{onetail}$	$p_{twotail}$
FVC2004db3a	0.7649	0.4749	0.6199	0.45	1.89	2.87%	5.74%
FVC2006db2a	0.8590	0.4892	0.6741	0.49	2.97	0.13%	0.26%

Table 3.3: Kendall τ correlations of the imposter and genuine distributions. The computed Z values and the one tail/two tail probabilities.

match scores (640.000) it may take several hours for the computation of this measures, therefore [98] suggest a stochastic approach. For the computation of the areas under the ROC curve we used a available Matlab script from Mathworks [29].

For the FVC2006db2a database the two tailed p value is 0.26% and for the FVC2004db3a the two tailed p value is 5.74%. Note that for the relative comparison of two fingerprint matchers, it is suggested [98] to use the one tail distribution. The values for the one tail p value are 2.87% and 0.13%, respectively. The borderline value for assuming significantly better performance between two ROC curves is $p \approx 5\%$. Thus, for both databases these values suggest that the observed difference may not be random. In other words, the performances of the corresponding algorithms are most likely different - the impact of the model based OF is statistically significant.

3.6 Conclusion

In this chapter we proposed a novel method for fingerprint ridge orientation modelling using Legendre Polynomials. The method proceeds in two steps. In the first step we roughly estimate the parameters using a closed form solution. In the second optimization step, we propose to use a non-linear optimization technique for more precise parameter estimation. We showed, that only five iterations of the Levenberg-Marquardt algorithm lead to a significantly improved orientation field. Due to the small number of necessary iterations, the runtime is typically two seconds for a fingerprint, using unoptimized Matlab code and high resolution orientation fields.

We want to point out that despite similarities to [90] our method enjoys important additional benefits. Apart the different basis functions, we model singularities more effectively. In [90] discontinuities, which result from singular points, are directly modelled. Due to the fact that Fourier series and polynomials are not well suited for the approximation of discontinuous signals, this achievement is obtained by a very high order approximation. On the other hand, using a high amount of basis functions leads to poor noise suppression and to poor interpolation properties. This problem is known as the Bias-Variance dilemma [6] in approximation theory. For further details we refer the reader to the Appendix. To solve this fundamental problem, we utilize the fact that singular points can be modelled by the zero-poles of the vectorial orientation data, without the necessity to directly model any discontinuities. As we have explained in detail, this can be achieved by using a non-linear parameter fitting technique. As a result, the proposed method enjoys a significantly higher smoothing fidelity in comparison to other methods available in literature.

Another significant attribute of the proposed model is the ability to represent any fingerprint ridge orientation field with a comparatively small number of parameters (typically 56).

For evaluation of the proposed method we perform feature extraction on publicly available databases (FVC2004db3a and FVC2006db2a). We compare our model based method to other state of the art methods and can report higher performance rates. Singular points are detected using a Poincáre-Index based detection algorithm and were compared to manually annotated data. Furthermore, for testing the improvement in orientation extraction, we replaced the orientation field of a state of the art fingerprint matcher. The matching results show improvements of 24% for the FVC2004db3a and 37% for FVC2006db2a databases, respectively. These improvements have been confirmed to be statistical significant.

Chapter 4

Applying Prior Knowledge to Fingerprint Ridge Orientation Models

Contents

Abstract

In this chapter we apply a priori knowledge to the fingerprint ridge orientation model from the previous chapter. Using a priori knowledge, the OFs can be constrained by the model to vary only in ways as they occur in nature. In the first part of this chapter, we will describe the idea of flexible templates models. This models, usually used for shape analysis in literature, can be used to apply prior knowledge within pattern recognition algorithms. Furthermore, we will give an overview of these methods and describe some of the available techniques. In an another section, we will describe how a similar technique can be used to estimate OFs in noisy fingerprints as well as to interpolate larger OF parts. The proposed method does not depend on any pre-alignment or registration of the input image itself. The training can be done fully automatic without any user interaction. We evaluated both, the generalisation as well as the prediction capability of the proposed method.

4.1 Introduction

The constructivist theorists of cognitive psychology believe that the process of seeing is an active process in which our world is constructed from both the retinal view and prior knowledge [75]. A visual example is given in Figure 4.1, where the black blobs can be deciphered only using a priori knowledge (hint: a Dalmatian is shown). This fact constitutes the motivation of all learning-based computer vision methods. A Widely used and established technique is based around so-called deformable templates models, who enable the user to apply a priori knowledge.

4.1.1 Deformable Template Models

In the last decade, the model-based approach towards image interpretation has proven very successful. This is especially true in the case of images containing objects with large variability. A deformable template model can be characterized as a model, which under an implicit or explicit optimization criterion, deforms a shape to match a known object in a given image [24].

Deformable template models are capable of dealing with a variety of shape deformations and variations, while maintaining a certain structure. The deformable models have wide applications in pattern recognition and computer vision, including image/video database retrieval, object recognition and identification, image segmentation, restoration

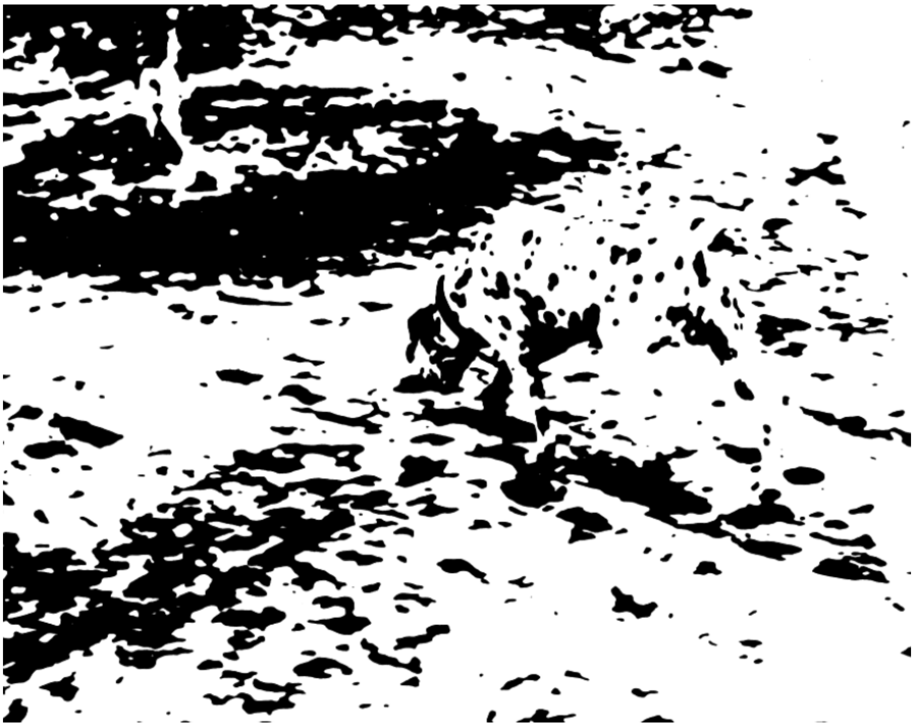


Figure 4.1: **Image interpretation using a priori knowledge.** The black blobs in the famous image from Preece et. al [75] could never be deciphered without a priori knowledge. This is the main assumption behind the 'constructivist approach' as described in [75], namely, that visual perception involves the intervention of representations and memories such as 'dog', 'park' etc.

and object tracking (for a detailed overview see [46] and references therein).

A representative problem for the application of deformable template models in computer vision is shape matching. Early research in this area concentrated mainly on rigid shape matching, where the matched shapes were obtained by applying simple transformations such as translation, rotation, scaling, and affine transformation [12] to the model template, which can be recovered using correlation-based matching or the Hough transform [16, 3]. Because of the rigidness of the above mentioned approaches, their practicability is limited. In most applications, an exact geometric model of the object is not available because of the variability in the imaging process and inherent within-class variabilities. Deformable template matching is more versatile and flexible in dealing with the deficiencies of rigid shape matching. The concept of deformable templates was introduced to computer vision simultaneously in the year 1973 by Widrow [96, 97] with the 'rubber masks' and Fischler and Elschlager [22] with the spring-loaded templates.

Deformable template matching is a more powerful technique because of its capability to deal with shape deformations and variations (see Figure 4.2). A deformable model is **active** in the sense that it can adapt itself to fit the given data. It is a useful shape model because of its flexibility, and its ability to both impose geometrical constraints on the shape and to integrate local image evidence. Therefore, the objective function consists of two parts:

1. The internal energy, **prior**, or geometrical information is related only to the geometric shape of the deformed template which is an intrinsic property of the template, independent of the input image.

2. The **likelihood term**, in all the cases, pertains to the input image data. Using this term, the deformable model interacts with the image, being attracted to the desired salient image features. This term measures the **fidelity**, or goodness of fit, of the template to the input image.

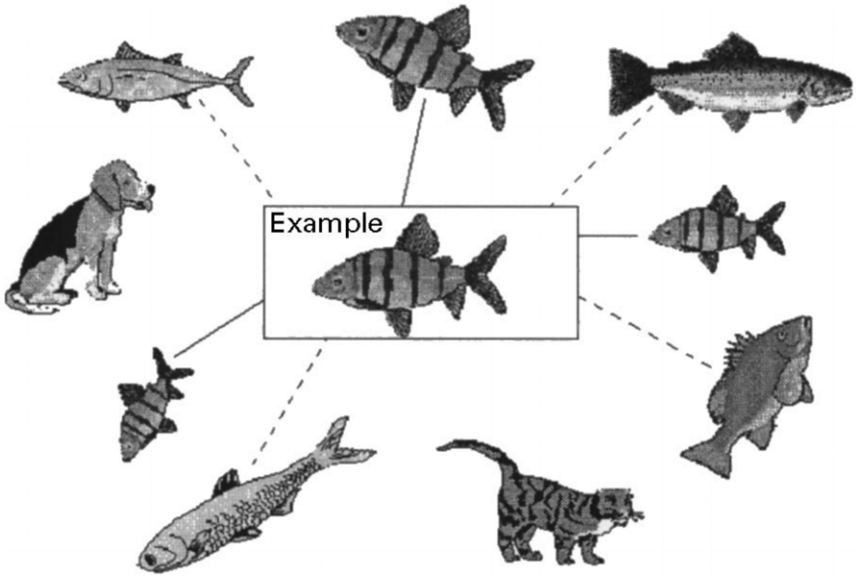


Figure 4.2: **An example of template matching.** The example fish can be matched to only 3 fishes (solid lines) with rigid template matching using translation, scaling and rotation. Ideally, it can be matched to all the fishes using a deformable template model (dashed lines). Illustration courtesy of [46].

Deformable template models can be partitioned into two classes. First, the freeform models who can represent any arbitrary shape as long as some general regularization constraints (continuity, smoothness, etc.) are satisfied. They are generally called active contours. On the other hand, parametric deformable models are capable of encoding a specific characteristic shape and its variation. The shape can be characterized by a parametric formula or using a prototype and deformation modes. Figure 4.3 shows a classification schema of various template models.

Of particular interest in context with this work are the parametric deformable template models. This type of flexible models can be used when some prior information of the geometrical shape is available, which can be encoded using a small number of parameters. It should be noted, that although a direct application of existing shape models is not possible for fingerprint OFs, some of the insights can be used to create flexible templates for OFs of fingerprints.

For parametric flexible templates, there exist generally two ways to parametrize the shape class and its variations [46]:

- The **analytical deformable template** are defined by a set of analytical curves (e.g. ellipses). The geometrical shape of the template can be changed by using different values of the parameters. Variations in the shape are determined by the distribution of the admissible parameter values. This representation requires that the geometrical shapes are well structured. Such 'hand crafted methods' can capture detailed knowledge of expected shapes, but usually this approach lacks generality. It is necessary to design both a new model and a scheme for fitting to images for each application.

- The **prototype-based deformable templates** are defined around a so called 'standard', 'prototype' or 'generic' templates which describes the 'most likely', 'average' or 'characteristic' shape of a class of objects (e.g. hands) which has a global conforming structure and possibly individual deviations. Each instance of the shape class is derived from the 'prototype' via a parametric mapping. The use of different parameter values again gives rise to different shapes. Variations in the shape are also determined by the distribution of the admissible parameter values of the mapping.

An example for an early and very well known deformable template models is the Active Contour Model (also known as Snakes) proposed by Kass et al. [49]. Snakes represent objects as a set of outline landmarks upon which a correlation structure is forced to constrain local shape changes. In order to improve specificity, many attempts at hand crafting a priori knowledge into a deformable template model have been carried out.

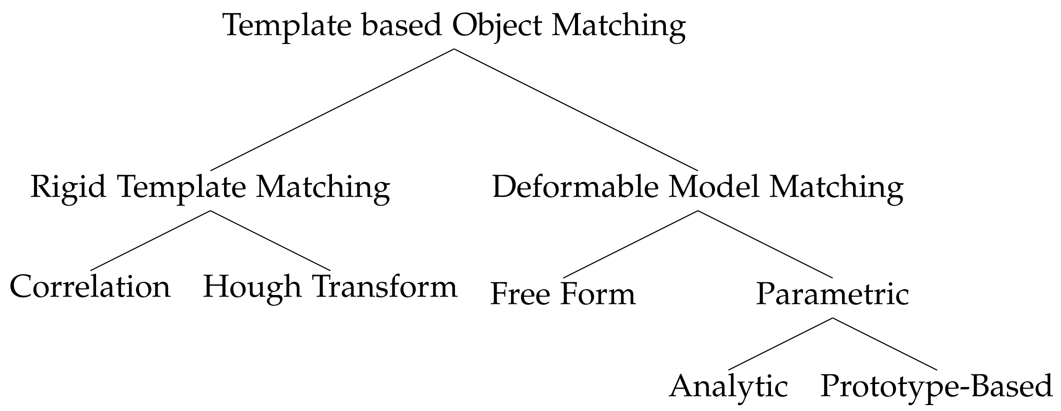


Figure 4.3: An overview of the template matching techniques as described in [46]

In a more general approach, while preserving specificity Cootes et al. [13] proposed the Active Shape Models (ASM) where shape variability is learned through observation. In practice, this is accomplished by a training set of annotated examples followed by a expert analysis [30] combined with a principal component analysis. A direct extension of the ASM approach has lead to the Active Appearance Models [14]. Besides shape information, the texture information, i.e. the pixel intensities across the object, is included into the model. For further information on deformable template models, the reader is referred to the surveys given in [46,24]

4.1.2 Prior Knowledge within Fingerprint Ridge Orientation Models

When studying available literature on fingerprint ridge orientation models, one will note that many authors discuss the same issues as described by flexible template models. The fundamental underlying concepts sound similar (a flexible template for a fingerprint class), with the strong distinction that instead of shapes, OFs are necessary. Surprisingly, this similarity has not been studied by researchers in the biometrics community in the past.

Besides exploiting the smoothness constrain in fingerprint images, many authors of existing work apply some type of parametric flexible templates. Usually, these methods exploit the fact that trajectories of certain differential equations (e.g. linear phase portraits, quadratic differentials) resemble the structure of fingerprint ridges. The underlying problem of such methods is the restricted adaptability to real fingerprint flow patterns. Generally, there exists a large trade off between generalisation and specificity. To sum up, these methods are:

- **not general enough:** while it is possible to fit the models to many classes of fingerprints, some fingerprints can not be modelled (eg. in case of linear phase portraits and quadratic differentials arch type fingerprints are impossible to model).

- **not specific enough:** many analytic parametric models can not only be used for the creation of fingerprint like pattern but also other geometric patterns (e.g. 'improper nodes' or 'saddles' in case of linear phase portraits). In many cases it is necessary to use heuristics in order to prevent the occurrence of these patterns, which overall leads to a less robust method.

- **not able to model a entire fingerprint:** because most analytic parametric template models can only describe a part of the fingerprint, several instances of a model need to be combined in order to model a entire fingerprint. Other methods (i.e. Zhou and Gu [32] use a point charge model) use only parametric analytic templates for SP modelling. Again, overall such combinations lead to a less robust method.

Studying the recent literature on orientation modelling suggests that a solution to the above mentioned problem lies in more suitable analytic expressions which result in a more specific and more general model than the existing ones. On the other hand, these 'hand crafted' analytic expressions together with the suggested heuristics are becoming more and more complex while still being of limited usefulness. In general, the mechanisms which give rise to variability are insufficiently well understood to allow a theoretical model of deformability to be proposed.

The only feasible apporach is to 'learn' specific patterns of variability from a representative training set of the OFs to be modelled. Therefore, this chapter proposes a method which uses a parametric deformable template model (as described in chapter 3 but applies a parametric prototype-based method. This contrasts all other existing fingerprint models and eliminates the need for a 'perfect' analytic expression.

The word 'active' in literature usually serves in two ways. First, 'active' refers to the ability of a model to adapt itself to fit the given data. Second, and more importantly, the word active is used to emphasize that a method 'actively' learns OFs from training samples. Due to the similarity to such flexible template models the proposed method in this chapter is called Active Fingerprint Ridge Orientation Model (AFROM).

4.1.3 Training and application of prototype-based parametric deformable models

The success of deformable template model approaches depends on the accurate description of the shape class - the expected shape instances and their variations. Some existing work on shape modelling has focused on the active learning of the shape models from training samples. In this existing work shape classes are described by learning both - the **’representative’** shape and the **’variability’** in the shape class [13,55].

A common technique which has been shown to be very useful in many computer vision applications, is principle component analysis (PCA). This usefulness is mainly due to its capability to reduce the dimensionality and to ’extract’ the important dimensions in terms of the amount of variations they explain. As a result, PCA plays an important role in learning object representations (e.g., eigenfaces [88]). A very prominent example for the application of PCA is given by Cootes et al. [13] who have adopted the method of learning deformable template models. Their method, called ’active shape models’, learns the prototype shape and its deformations from a collection of correctly annotated example shapes. This completely contrasts other methods where instead the parametric form is ’hand crafted’ for the given shape class. Basically, polygonal representations are used for modelling the shapes. Therefore, it is necessary to manually aligning the training set, i.e., establishing the correspondences between the ’landmark points’ (nodes) of training samples of the same class and to calculate the mean position and variation of each node from the training shapes. The mean shape is used as the generic template of the class of shapes. A number of modes of variation, i.e., the eigenvectors of the covariance matrix, are determined for describing the main factors by which the exemplar shapes tend to deform from the generic shape. A small set of linearly independent parameters are used to describe the deformation. In this way, their shape model allows for considerable meaningful variability, but is still specific to the class of structures it represents. The major contribution of their work is that the active shape model is able to learn the characteristic pattern of a shape class and can deform in a way which reflects the variations in the training set. The limitations of the approach are its sensitivity to partial occlusion, and its inability to handle large scale and orientation change.

4.2 Training the Model

We use a commercial fingerprint software from Siemens (Siemens IT Solutions and Services, Biometrics Center) for local OF estimation and for the segmentation of the image into foreground and background pixels. Note that no other processing, i.e. registration or alignment has been employed. For the training step, the extracted raw OF is smoothed using the method described in Chapter 3. Using this method it is possible to estimate the OF also in the corners of the image (background). Furthermore, the method suppresses artefacts (contained in many images).

4.2.1 Representation of Fingerprint Flow Patterns

For the representation of fingerprints OFs, we use 12^{th} order Legendre Polynomials similar as described in Chapter 3. Alternatively, one could also use the parametric OF representation as described in [90] by Wang et al. In the following, we give a short overview of the used OF approximation method. Let $2O(x, y)$ be the doubled orientation and

$$\Phi(\mathbf{x}) = [\phi_0(\mathbf{x}) \dots \phi_n(\mathbf{x})] \tag{4.1}$$

the row vector containing Legendre polynomials $\phi(\mathbf{x})$ evaluated for a given coordinate $\mathbf{x} = (x, y)$. The system matrix is given as $\mathbf{V}$ and consists of the row vectors $\Phi(\mathbf{x})$,

$$\mathbf{V} = \begin{pmatrix} \Phi(\mathrm{x}_1) \\ \Phi(\mathrm{x}_2) \\ \vdots \\ \Phi(\mathrm{x}_i) \end{pmatrix} = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \dots & \phi_n(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \dots & \phi_n(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_i) & \phi_1(\mathbf{x}_i) & \dots & \phi_n(\mathbf{x}_i) \end{pmatrix} \tag{4.2}$$

$\mathbf{f_x}$ and $\mathbf{f_y}$ contain the vectorial orientation data (computed using sine and cosine function from $2O(x, y)$). Then one can compute the parameter vector $\mathbf{c} = [\mathbf{a}, \mathbf{b}]$ for the vectorial approximation as described in the following:

$$\mathbf{a} = \mathbf{V}_\mathbf{w}^+ \mathbf{W} \mathbf{f_y} \qquad \mathbf{b} = \mathbf{V}_\mathbf{w}^+ \mathbf{W} \mathbf{f_x} \tag{4.3}$$

Where

$$\mathbf{V}_\mathbf{w}^+ = (\mathbf{V}^T \mathbf{W} \mathbf{V})^{-1} \mathbf{V}^T \tag{4.4}$$

is the pseudo-inverse of the system matrix $\mathbf{V}$. For details on the computation of the pseudo-inverse we refer to the Appendix in chapter 6.2. The diagonal weighting matrix $\mathbf{W}$ is computed using fingerprint segmentation, where the diagonal elements are $\omega = 0$ for background and $\omega = 1$ for foreground pixels.

4.2.2 Computing a Subspace

Suppose now we have s sets of parameters $\mathbf{c}_i = [\mathbf{a}_i, \mathbf{b}_i]$ which were generated from s fingerprints as described above. These vectors form a distribution in the n dimensional space. If one can model this distribution, one can generate new examples similar to those in the original training set. Furthermore, one can decide whether a given OF is a plausible fingerprint flow patterns. We apply Principal Component Analysis (PCA) to the set of parameters in order to model flow variations of naturally occurring fingerprints. Therefore, we compute the mean vector $\overline{\mathbf{c}}$

$$\overline{\mathbf{c}} = \frac{1}{s}\sum_{i=1}^{s}\mathbf{c}_i \tag{4.5}$$

and the covariance matrix $\mathbf{S}$

$$\mathbf{S} = \frac{1}{s-1}\sum_{i=1}^{s}(\mathbf{c}_i - \overline{\mathbf{c}})(\mathbf{c}_i - \overline{\mathbf{c}})^T \tag{4.6}$$

of the data, followed by the eigenvectors

$$\mathbf{e} = [e_1, e_2, \dots, e_t] \tag{4.7}$$

and the corresponding eigenvalues

$$\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \dots, \lambda_t] \tag{4.8}$$

of $\mathbf{S}$ (sorted largest first). Let Ω be the space of all possible parameters and Ψ the linear subspace spanned by the PCA. Then we can project parameters from Ω to Ψ using the linear projection φ:

$$\mathbf{d}_i = \varphi(\mathbf{c}_i) \qquad = \mathbf{e}^T(\mathbf{c}_i - \overline{\mathbf{c}}) \qquad \text{projection } \varphi \tag{4.9}$$

$$\mathbf{c}_i = \varphi^{-1}(\mathbf{d}_i) \qquad = \overline{\mathbf{c}}_i + \mathbf{e}\mathbf{d}_i \qquad \text{inverse projection } \varphi^{-1} \tag{4.10}$$

here $\mathbf{c}_i$ represents a point in the high dimensional space Ω and $\mathbf{d}_i$ the same point projected in to the linear subspace Ψ. The number of eigenvectors t to retain should be chosen so that the model represents a sufficiently large proportion of the total variance. Thus, the original high dimensional data can be approximated using a model with much fewer parameters. In Figure 4.4 the eigenvalue spectrum of 2000 fingerprint vectors (NIST4, f-prints database) is shown. Note that these fingerprints were not registered nor aligned in any other form. Only image cropping according to a segmentation has been performed.

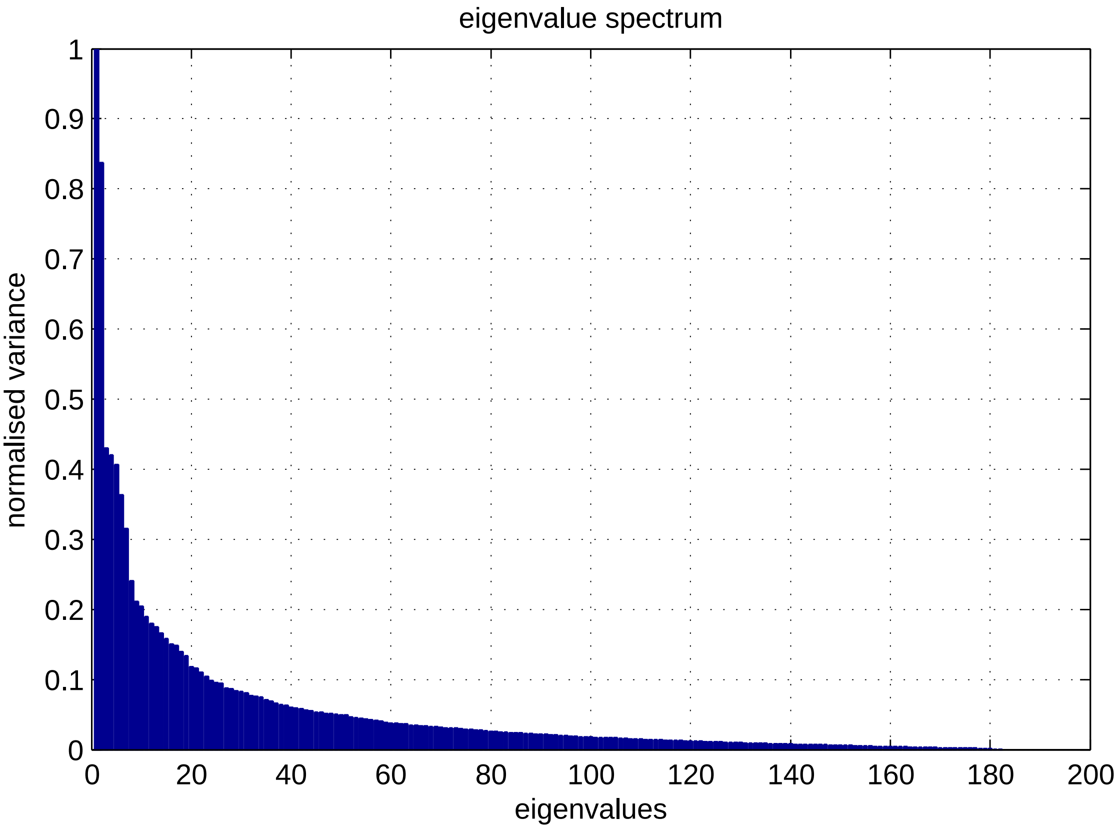


Figure 4.4: Eigenvalue spectrum of 2000 fingerprint vectors (NIST4 f-prints database). No registration or alignment to this database has been applied. From the figures (normalised variance), it can be seen that the first 40 eigenvalues accord to the majority (95%) of the total variance. Another important observation worth mentioning is that the first two eigenvalues dominate the eigenvalue spectrum.

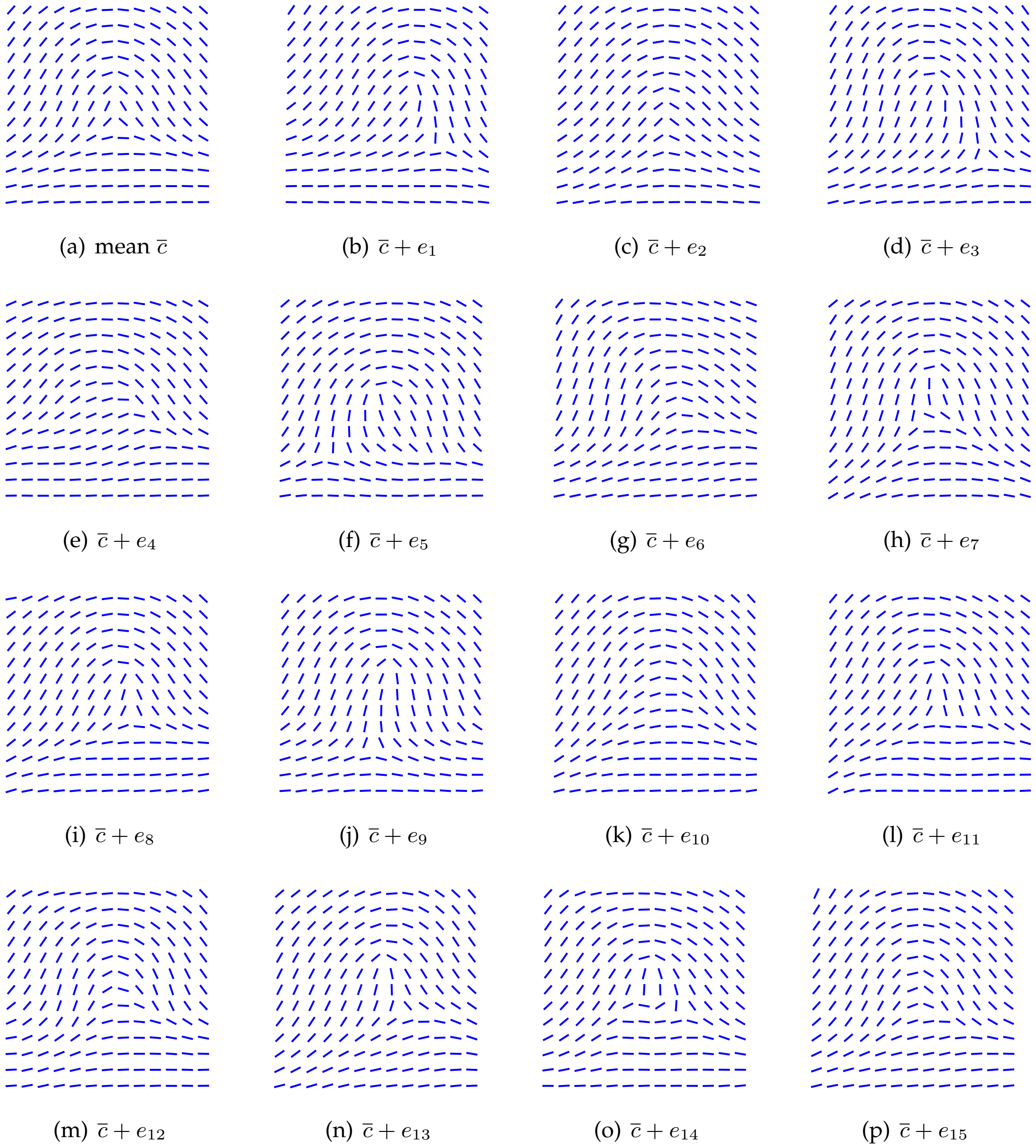
4.2.3 Interpretation of 'Eigen-orientations'

Informally, Eigen-orientations can be seen as a set of 'standardized orientation ingredients', derived from statistical analysis (PCA as described above) of many fingerprints. Any fingerprint can be considered to be a combination of these standard orientations. This interpretation is similar to other methods available in literature (e.g. Eigenfaces as described by Turk and Pentland in [88]). A major distinction to the above mentioned method is that we do not have pixel based representation, the eigen-orientations are coefficients of Legendre polynomials. In other words, the mean fingerprint can be modified using previously computed ’perturbation’ functions.

Regarding the visualization of Eigenmodes there are limitations in comparison to Eigenfaces. The relation between coefficients and the orientation vectors is non-linear (arctan function, fraction), therefore it is not possible to directly visualize a individual Eigen-orientation. Instead we have to show the impact of each Eigen-orientation on the mean vector $\mathbf{c}_i$. From a different point of view, the Eigen-orientations can be seen as ’orientation ingredients’ which, when applied, can be used to modify a standard fingerprint in order to generate every other possible fingerprint. Figure 4.5(a) shows the mean of the coefficient vectors as described in Equation 4.5. The first 15 Eigen-orientations are shown in Subfigure 4.5(b)-4.5(p). All of this Eigen-orientations form smooth patterns of fingerprints.

In Figure 4.6 it is shown how the fingerprint flow pattern changes when moving along the principal axis. As can be seen, it is possible to generate left and right loop type fingerprint flow patterns. In Figure 4.7 the same procedure is done for the second axis. This illustration shows that using the second Eigen-orientation it is possible to model the patterns of whorls and arch type fingerprints.

Using the first two eigenvalues, it is possible to model five types of fingerprints (the mean can be interpreted as tented arch pattern). This fact explains also the high dominance of the first two Eigen-orientations in the eigenvalue spectrum. However, it should be noted that the foremost two eigen-Orientations can be used only for a ’rough’ global approximation of fingerprint OFs. For a finer modelling of fingerprints it is necessary to apply a higher amount of eigen-Orientations. For details on the selection of the optimal number of eigen-Orientations see Section 4.4.3.



(a) mean $\overline{c}$

(b) $\overline{c} + e_1$

(c) $\overline{c} + e_2$

(d) $\overline{c} + e_3$

(e) $\overline{c} + e_4$

(f) $\overline{c} + e_5$

(g) $\overline{c} + e_6$

(h) $\overline{c} + e_7$

(i) $\overline{c} + e_8$

(j) $\overline{c} + e_9$

(k) $\overline{c} + e_{10}$

(l) $\overline{c} + e_{11}$

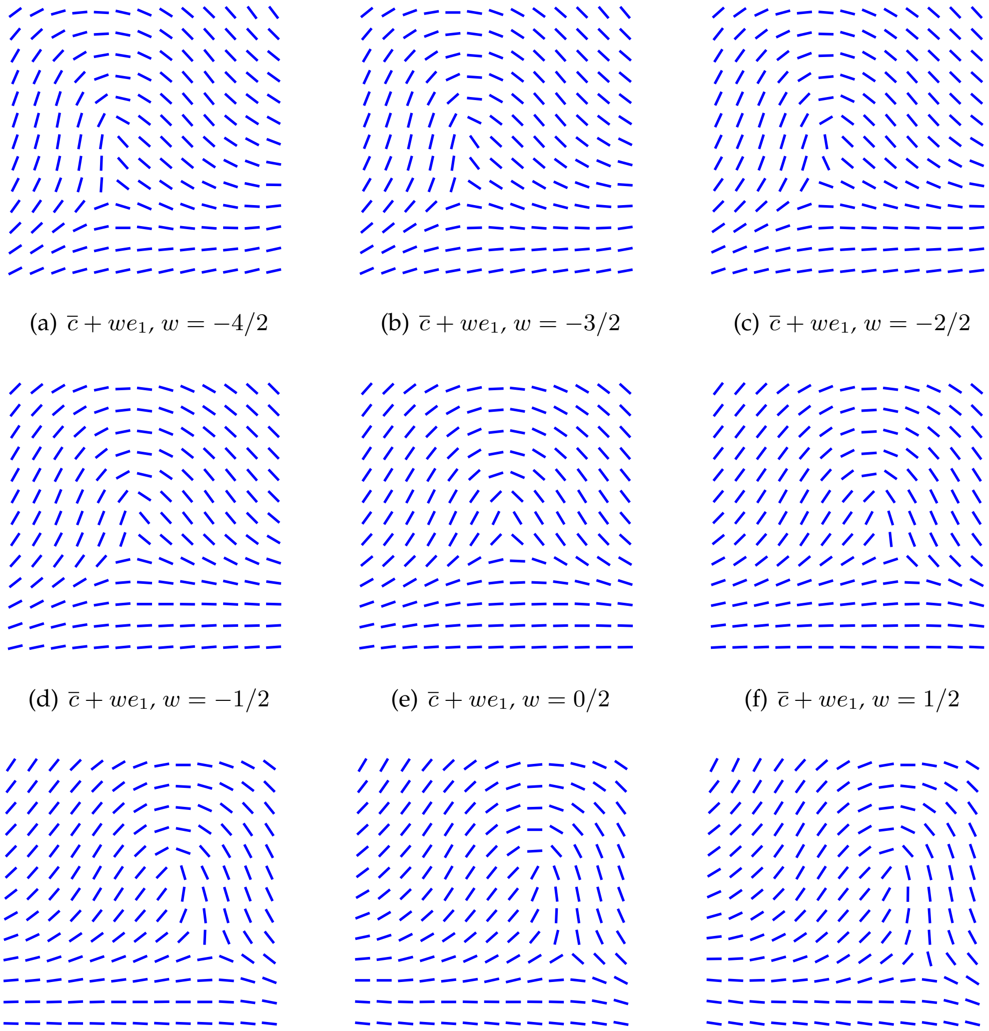
(m) $\overline{c} + e_{12}$

(n) $\overline{c} + e_{13}$

(o) $\overline{c} + e_{14}$

(p) $\overline{c} + e_{15}$

Figure 4.5: **Eigen-orientations.** This illustrates the Eigenmodes ('Eigen-orientations') computed using PCA. Note that it does not make much sense to visualize a Eigen-orientation e_i directly, because of the non-linear correlation between orientation and coefficients (due to the arctan function and the division for back conversion to the angle space). Due to this reason we show how the mean can be altered by adding different Eigen-orientations. Interestingly almost all of the Eigen-orientations result in smooth OFs of fingerprints.



(a) $\bar{c} + we_1$, $w = -4/2$

(b) $\bar{c} + we_1$, $w = -3/2$

(c) $\bar{c} + we_1$, $w = -2/2$

(d) $\bar{c} + we_1$, $w = -1/2$

(e) $\bar{c} + we_1$, $w = 0/2$

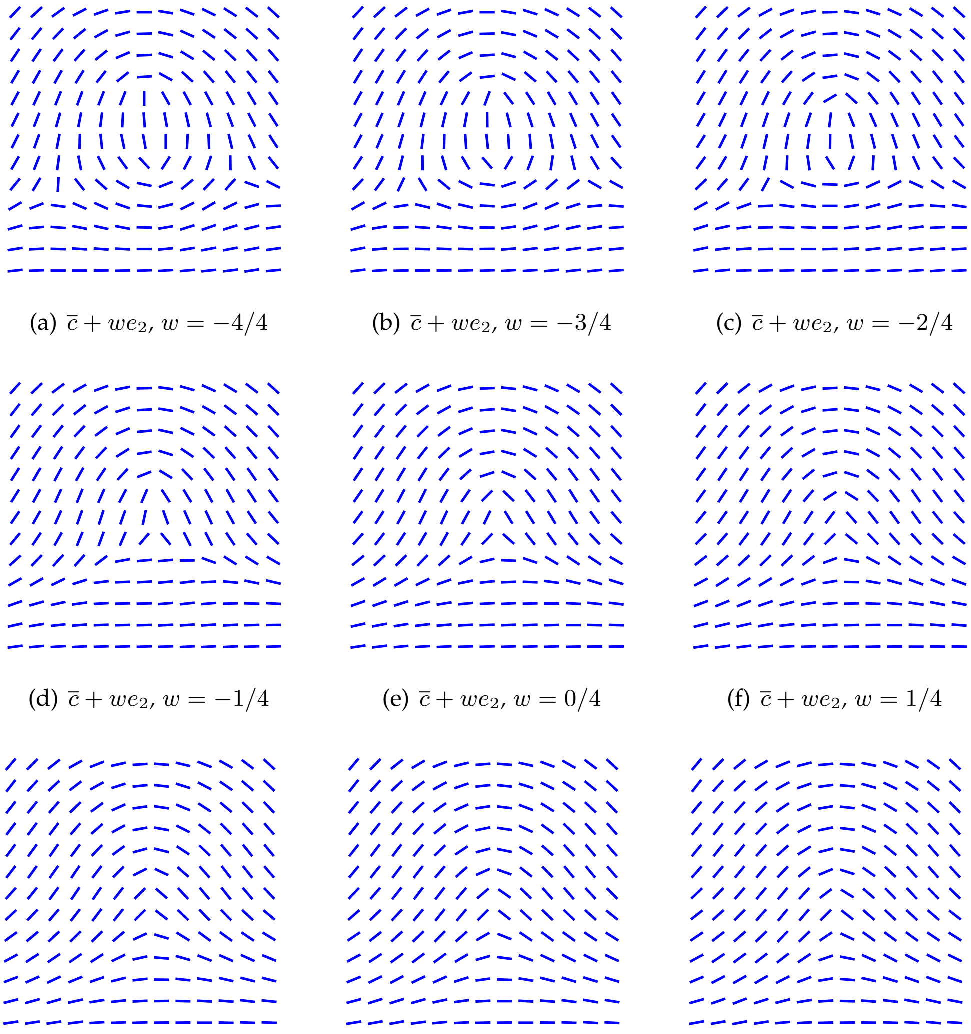
(f) $\bar{c} + we_1$, $w = 1/2$

(g) $\bar{c} + we_1$, $w = 2/2$

(h) $\bar{c} + we_1$, $w = 3/2$

(i) $\bar{c} + we_1$, $w = 4/2$

Figure 4.6: **Movement along the principal axis.** The illustration shows how the flow pattern changes along the principal axis. Moving in negative and positive direction (on the principal axis) varies the flow from left loop to right loop.



(a) $\bar{c} + we_2$, $w = -4/4$

(b) $\bar{c} + we_2$, $w = -3/4$

(c) $\bar{c} + we_2$, $w = -2/4$

(d) $\bar{c} + we_2$, $w = -1/4$

(e) $\bar{c} + we_2$, $w = 0/4$

(f) $\bar{c} + we_2$, $w = 1/4$

(g) $\bar{c} + we_2$, $w = 2/4$

(h) $\bar{c} + we_2$, $w = 3/4$

(i) $\bar{c} + we_2$, $w = 4/4$

Figure 4.7: **Movement along the second axis.** In this illustration we move along the second axis. As can be seen, the fingerprint flow pattern changes from whorl type to arch type. Note that the usage of the first two Eigen-orientations allows to model five different fingerprint types.

4.3 Fitting the Model to the Nearest Plausible Fingerprint Flow Pattern

In order to only generate examples similar to the training set, we have to choose a parameter $\mathbf{d} \in \Psi$. Therefore, we propose to minimize the following cost function:

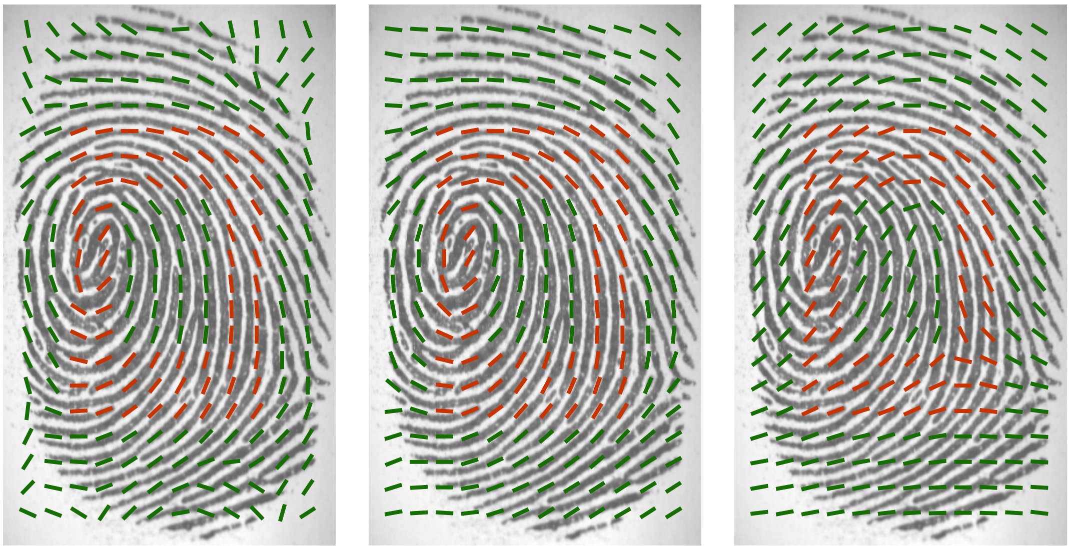
$$\underset{\mathbf{d}}{min} \quad \sum_{j=1}^{i} \omega_i \left[\sin\Big(M(\mathbf{x}_j) - O(\mathbf{x}_j) \Big) \right]^2 + \mu \left[\frac{1}{P(\mathbf{x}_j)} - P(\mathbf{x}_j) \right]^2 \tag{4.11}$$

The cost function compares the model's orientation estimation $M(\mathbf{x}_j)$ with the observed function value $O(\mathbf{x}_j)$ (obtained from local image gradients). We use the sine-function in order to resolve the discontinuity problem at zero and π. Then, one can compute $M(\mathbf{x}_j)$ as described in Equation 4.12:

$$M(\mathbf{x}_j) = \frac{1}{2} \arctan \frac{\Phi(\mathbf{x}_j)\mathbf{a}^T}{\Phi(\mathbf{x}_j)\mathbf{b}^T} \tag{4.12}$$

$$P(\mathbf{x}_j) = (\Phi(\mathbf{x}_j)\mathbf{a}^T)^2 + (\Phi(\mathbf{x}_j)\mathbf{b}^T)^2 \tag{4.13}$$

Note that the vectors $\mathbf{a}$ and $\mathbf{b}$ can be computed by the inverse mapping $\mathbf{c} = [\mathbf{a}, \mathbf{b}] = \varphi^{-1}(\mathbf{d})$ The second term of Equation 4.11 is a penalty function which regularizes the orientation vector to unit length ($\sin^2 + \cos^2 = 1$). The regularization prevents solutions which represent 'illegal' linear combinations of the Eigen-orientations. Such 'illegal' linear combinations create orientation vectors much larger or much smaller than unit size and hence create flow patterns completely different from the learning dataset. Enforcing the orientation vector to unit length creates hard to optimize problems and does not generalize well on a slightly different image set (intra class variability). Practical observations show that allowing a minor variation from the unit size (eg. in the range $[0.3 - 3]$) enables the model to generalize much better, at the same time disabling the model to create 'illegal' flow patterns. A typical scenario is shown in Figure 4.8 where a different OF (compared to the training set) is fitted to the model. We want to point out, that this regularization scheme is the main mechanism to enable the model to fit slightly rotated and shifted fingerprints.



(a) *no treatment* (b) *regularisation as proposed* (c) *constraining to unit length*

Figure 4.8: **Optimization options.** In this Figure three types of optimization preferences are shown for illustration. We use a fingerprint from the fvc2006db2a fingerprint database. The input data is given in red colour, the output of the optimization is shown in green. In Figure 4.8(a) the optimization is done without any treatments. In this case, the optimization finds linear combinations of Eigen-orientations which result in an extremely low error in the input data (red), unfortunately the solution is not optimal regarding interpolation and extrapolation (green) quality. Basically, the problem can be accounted to invalid linear combinations of Eigen-orientations. Such linear combinations can result in extremely large or extremely small numerical values (length of orientation vectors). One way to enforce the optimization to employ linear combinations which were similar to the 'learned' one is to constrain the orientation vectors to unit length. Unfortunately such a optimization (shown in Figure 4.8(c)) it is not easy to carry out (many local minima). This is especially an issue for Fingerprints which are different from the learning set (e.g. contain off-centric cores where the learned one has centred cores only). Regularisation allows a variation of the orientation vector lengths but still favouring solutions with unit length. Using this procedure, it is possible to model fingerprint flows which are slightly different from the training dataset.

Optimization

The minimization of the cost function in Equation 4.11 is done by using the Levenberg-Marquard (LM) algorithm. A LM iteration can be given as:

$$\mathbf{d_{iter+1}} = \mathbf{d_{iter}} + (\mathbf{H} - \lambda_{iter}\mathrm{diag}[\mathbf{H}]^{-1})\mathbf{D} \tag{4.14}$$

$\mathbf{H}$ is the Hessian and $\mathbf{D}$ is the derivative of the cost function. λ_{iter} is called the damping parameter and is an integral part of the LM. For further details we refer the reader to [67]. The parameter vector $\mathbf{d}$ can be computed as described in Equation 4.9 using the transformation $\mathbf{d}_i \approx \varphi(\mathbf{c}_i)$. Where $\mathbf{c}$ consists of the concatenated parameter vectors $\mathbf{c} = [\mathbf{a}, \mathbf{b}]$. Note that, as described above, each iteration of the LM uses the inverse mapping $\mathbf{c}_i = \varphi^{-1}(\mathbf{d}_i)$ in order to evaluate the cost function as given in Equation 4.11. The factor μ is set to $3 * 10^{-4}$ in all our experiments. The initial value $\mathbf{d_0}$ for the LM is set to the null vector. This corresponds to the mean OF ($\bar{\mathbf{c}}$). The LM algorithm stops when a minima is reached or when the number of iterations exceeds 40.

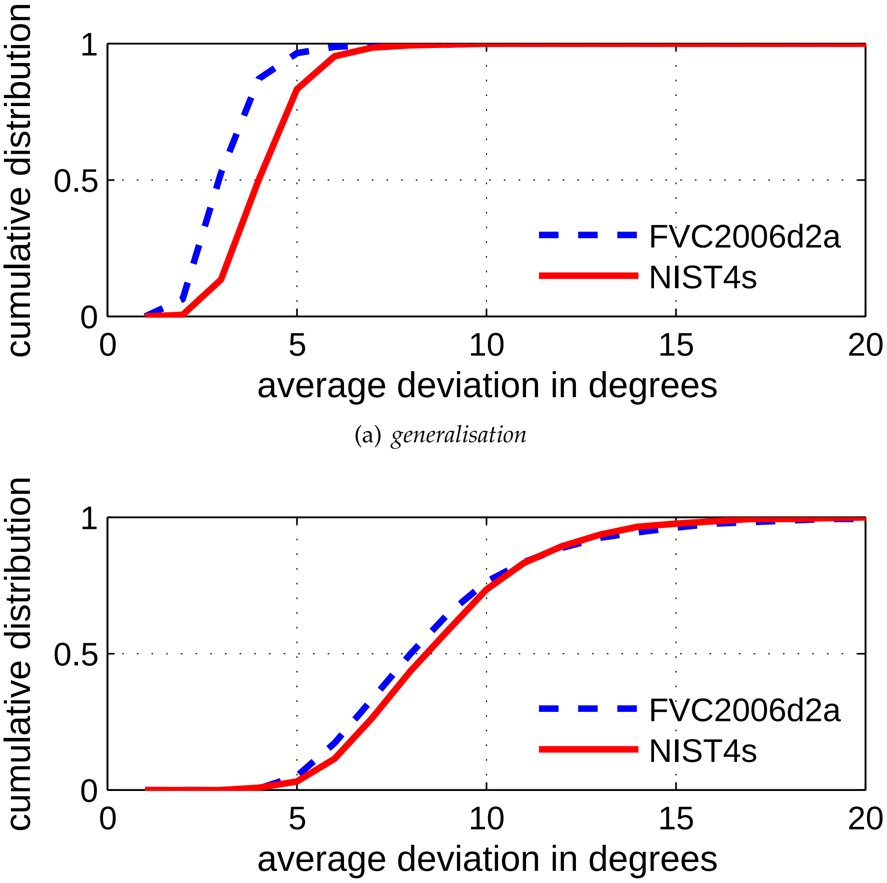
4.4 Evaluation

This section presents the experimental results. For training the model, we used the NIST4 special database [91]. This database contains 2000 fingerprints evenly distributed among the five Henry classes. The number of Eigen-orientations is limited to 80. For evaluation of the proposed method, we used the NIST4 s-prints (all 2000 images) and the FVC2006 2a [20] (all 1680 images) database.

4.4.1 Generalisation Test

In this subsection we test how well the proposed model generalises to a given test database. Therefore the model is fitted to the raw (unsmoothed) OF of the given fingerprint. To measure the quality of the fit, we compute the absolute mean deviation between the ground truth OF and the fitted orientation field in degrees, where an error is only computed for foreground pixels. The ground truth OF is computed using the mentioned fingerprint software. The figures depicted in Figure 4.9(a) show cumulative distribution functions of the absolute mean deviation in degrees summarized over all images of the database. Most of the images show a mean deviation of smaller than five degrees. A large fraction of this error can be adhered to the block-wise processing of the commercial fingerprint matcher. Furthermore, we want to point out that the ground truth OF contains errors and thus a possible improvement using the proposed method is impossible to measure.

The reader should note that this evaluation procedure is exactly the same as described in [41]. The experiments and datasets are identical, but the authors of [41] deleted 20% of the images (due to missing SPs). In direct comparison our model generalises significantly better to fingerprints than the one proposed by Huckemann et al. [41]. We can report almost all of the images to have smaller than 10 degrees absolute mean deviation, in comparison [41] can only report 50%.



(a) *generalisation*

(b) *prediction*

Figure 4.9: Generalisation and Prediction Evaluation.

4.4.2 Prediction Test

The orientation interpolation capability of the proposed algorithm is tested in a simulated scenario where we remove 70% of the OF. The OF remains in a rectangle with 40% area size, except a smaller rectangle with 10% of the total image size in pixels. Both rectangles are centred in the middle of the image and exhibit the same aspect ratio as the image. See Figure 4.4.2 for a visual explanation of the proportions.

Using this scenario, we tested the extrapolation as well as interpolation ability of the proposed algorithm. The figures are computed for the predicted OF only. Additionally, the background is removed from the input OF. This prediction evaluation is done using

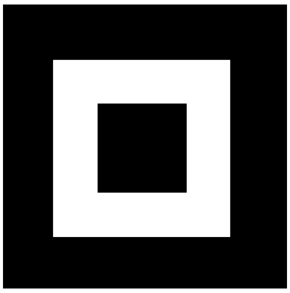


Figure 4.10: **Image proportions for prediction scenario.** The black area constitutes the fingerprint image area which is removed for the prediction evaluation. The white space displays the area from which the orientation information is used for the interpolation and extrapolation of the fingerprints OF.

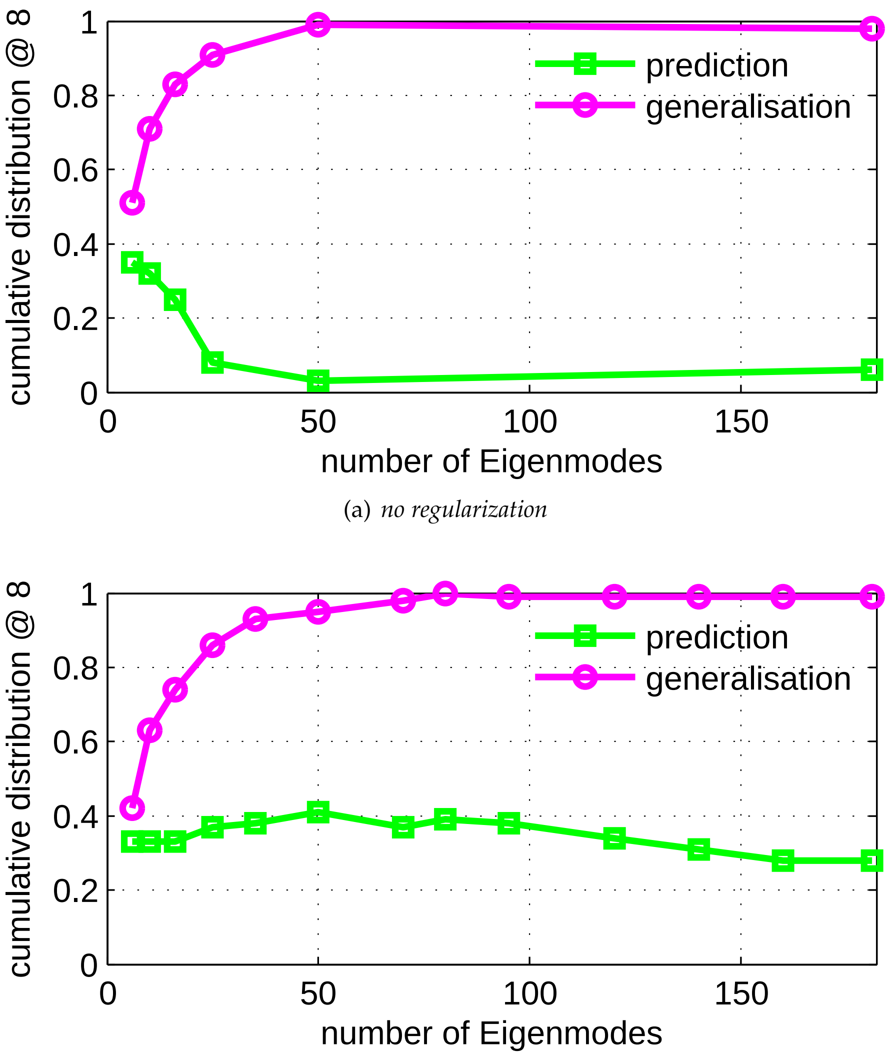
the same database configuration as mentioned above. It should be noted that not all predictions with a large absolute mean deviation are wrong in terms of plausibility.

The only comparable work with a significant large evaluated database is available from Hotz [39] (co-author of [41]). In his evaluation scenario the prediction was performed for only 5% occlusion (compared to 70% of our testing scenario). Unfortunately, this makes a possible comparison meaningless.

4.4.3 Estimating the Number of Eigen-orientations

In order to estimate the best number of Eigen-orientations we performed the above mentioned prediction and generalisation experiments for a varying number. The evaluation criteria for the prediction and generalisation figures was the relative number of fingerprints with less than eight degrees average error. Due to the computational burden, only the first 100 images of the NIST4 database (s-prints) were used. We performed this evaluation for two scenarios. In the first scenario (shown in Subfigure 4.11(a)) one can see how the number of coefficients affects the model without regularisation. Furthermore, it shows the trade off between generalisation and prediction capability of the method. In general, a lower number of Eigen-orientations results in good prediction figures but bad generalisation capability of the model - and vice versa. The second scenario (see Subfigure 4.11(b)) shows the model as proposed with the regularisation. It is clearly visible that the regularisation leads to a significant improvement.

In Figure 4.12 we give a example of the model fitted to a noisy fingerprint OF. The number of Eigen-orientations is varied from low to high. It can be seen that the application of a very higher number of Eigen-orientations is prone to noise (Subfigure 4.12(b)). Using a number in between these two extrema represents the best trade off. As



(a) *no regularization*

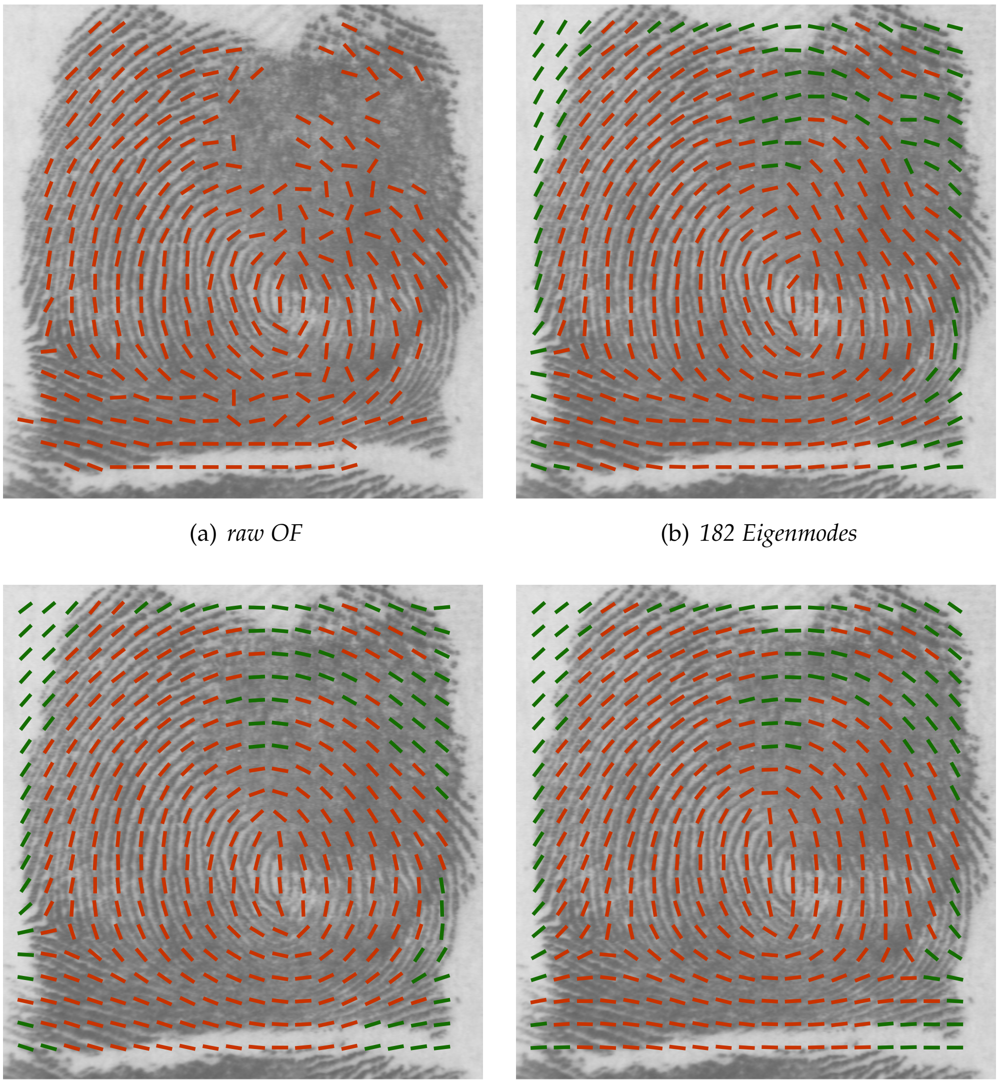
(b) *with regularization*

Figure 4.11: **Regularisation.** Applying a regularisation on the cost function, where we force the orientation vectors to unit length, we can significantly improve the results.

can be seen in Figure 4.12(c), the model correctly estimated the OF, while at the same time preserving fine details of the OF.

4.4.4 Illustrative Examples

In this subsection we want to display some illustrative examples showing the abilities of the proposed method. Figure 4.13 shows the model fitted to difficult fingerprint sam-



(a) *raw OF*

(b) *182 Eigenmodes*

(c) *80 Eigenmodes*

(d) *5 Eigenmodes*

Figure 4.12: **Effects of the number of Eigen-orientations.** In case of to few Eigen-orientations (Subfigure (d)) the model fails to generalize. Typically the application of a very higher number of Eigen-orientations is prone to noise (Subfigure (b)). Using a number in between these two extrema represents the best trade off. As can be seen in Figure 4.12(c), the model correctly estimated the OF, while at the same time preserves fine details.

ples, by using the above mentioned prediction scenario for illustration. The prediction is shown in green color. Subfigure 4.13(a) shows the model fitted to a rotated fingerprint.

Although artefacts started to raise, the model's prediction is still acceptable. Subfigure 4.13(b) shows the model fitted to a loop type fingerprint, here the SP is shifted to the far right. The model is able to perfectly fit and predict the OF of the shown fingerprint. Subfigures 4.13(c) and 4.13(d) show the model fitted to fingerprints from a completely different database (FVC2006db2a) in comparison to the learning dataset. Subfigure 4.13(c) shows a completely uncentred whorl type fingerprint fitted. In Subfigure 4.13(d) we fitted the model to a fingerprint image which has been wrongly segmented. Although the prediction is far from perfect, the resulting flow pattern looks naturally possible.

A prediction capability comparison to other methods available in literature is given in Figure 4.14. We used a noise image of a whorl type fingerprint (shown in 4.14(a)) for this illustration.

The proposed method (see Subfigure 4.14(b)) is generating a smooth and plausible OF, exhibiting a very high prediction quality. The FOMFE approach of [90] shown in Subfigure 4.14(c) has very poor prediction capabilities, especially visible in the outer regions. In direct comparison, the proposed method from chapter 3 generates a smoother flow. Despite this smoothness the method is not able to correctly recover the flow pattern of the fingerprint.

Figure 4.15 shows fingerprint examples where the model failed to fit the OF. In Subfigure 4.15(a) the algorithm failed to correctly model the left loop of the twin loop. In Subfigure 4.15(b) one can see the model fitted to a noisy OF, apparently the application of the algorithm was not successful.

Illustration 4.15 shows the 180 degree rotation of a loop type fingerprint. While in the unrotated case (Subfigure 4.16(a)) the model perfectly fitted the fingerprint, the model corrects the OF of the rotated version to a whorl type fingerprint.

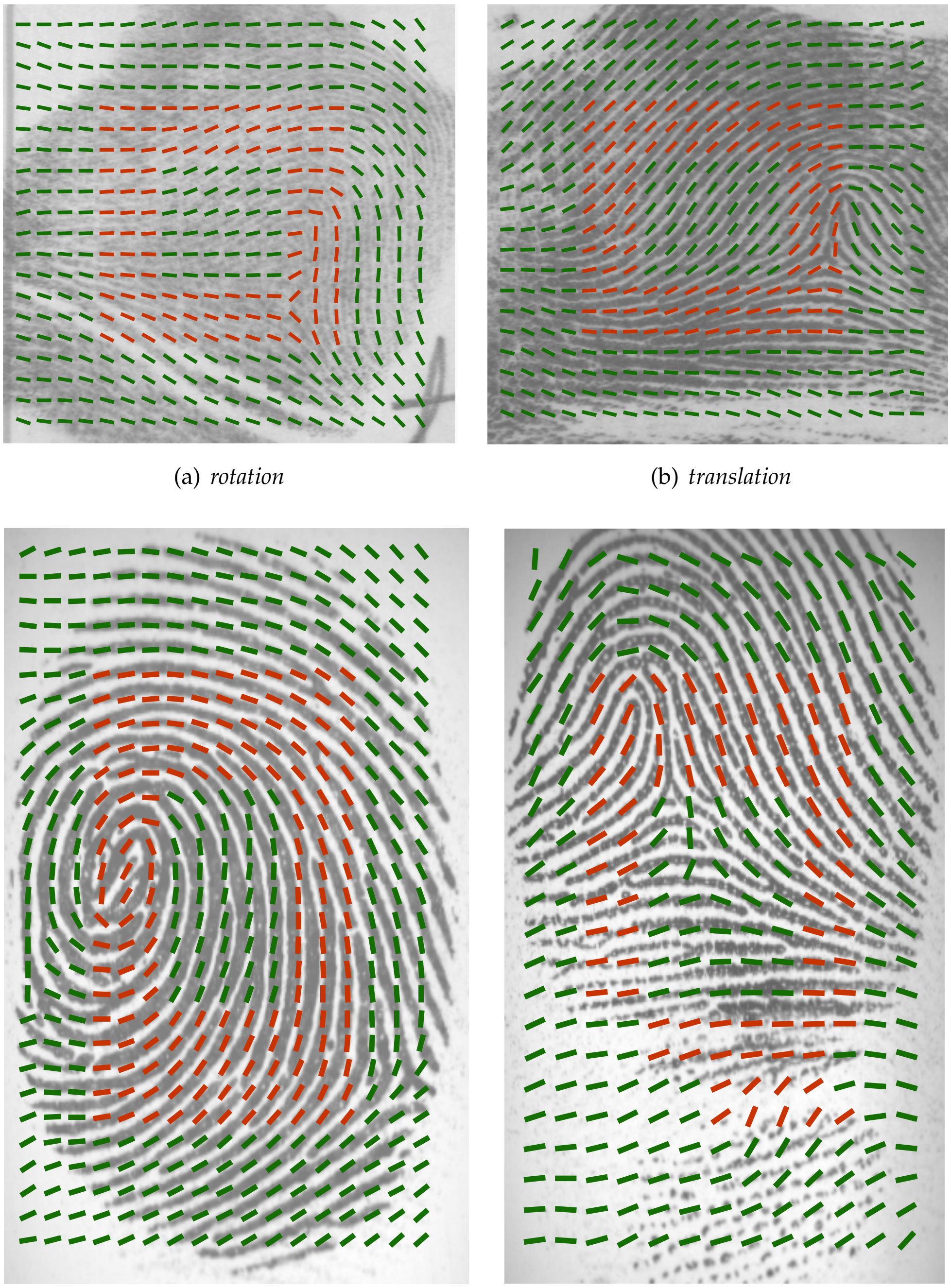
Figure 4.17 shows the OF estimation in comparison to a commercial fingerprint matcher.

4.4.5 Conclusion

In this chapter we presented a deformable template method for fingerprint ridge flows. The fingerprint orientation field (OF) can be constrained by the Active Fingerprint Ridge Orientation Model (AFROM) to vary only in ways seen in a training set. The OF of fingerprints is represented by a vectorially linear regression using Legendre polynomials. Fitting parameters to a given fingerprint is done using the Levenberg-Marquard (LM) algorithm. During the optimization procedure the parameters are limited to a previously learned linear subspace, where only 'legal' fingerprints reside. Using the proposed method, the AFROM iteratively deforms to fit an OF of a fingerprint. Our method does not depend on any pre-alignment or registration of the considered images. The training can be done fully automatic without any user interaction. Furthermore, our method does not depend on any other computed data, except a segmentation.

In the evaluation section of this chapter, we perform generalisation and prediction tests of the proposed method. A generalisation test is done in order to evaluate how well the model 'fits' to a large number of OFs. Using the presented prediction test, we asses how specific the model is. This is the ability to constrain unknown or noisy regions of the OF to valid fingerprint flow patterns. All experiments are performed on public databases, one of the databases is is fairly different to the learning dataset) These experiments, comparable with a very recent paper [41], assess our method a very good performance. Furthermore, it should be noted that our method is the first fingerprint OF model making use of prior knowledge for OF estimation. The major conception behind existing methods (e.g. [41]) is a hand crafted model which fits only to valid fingerprint OFs, without the possibility for machine based training. Our approach can also be seen as a method to find those elements (Eigenmodes comply to 'Eigen-Orientations') which, when (linear) combined, give biological valid patterns of fingerprints. We want to point out, that we used the full NIST4 f-prints database for training, including many noisy fingerprint images.

Future work includes the experimentation with other subspace methods than PCA (e.g. ICA, K-PCA, etc.). Moreover, the regularization term of the cost functional accommodates a large potential for future improvements. Another topic is the inclusion of an image quality estimation algorithm, where the model adjusts the amount of prior knowledge depending on the local image quality.



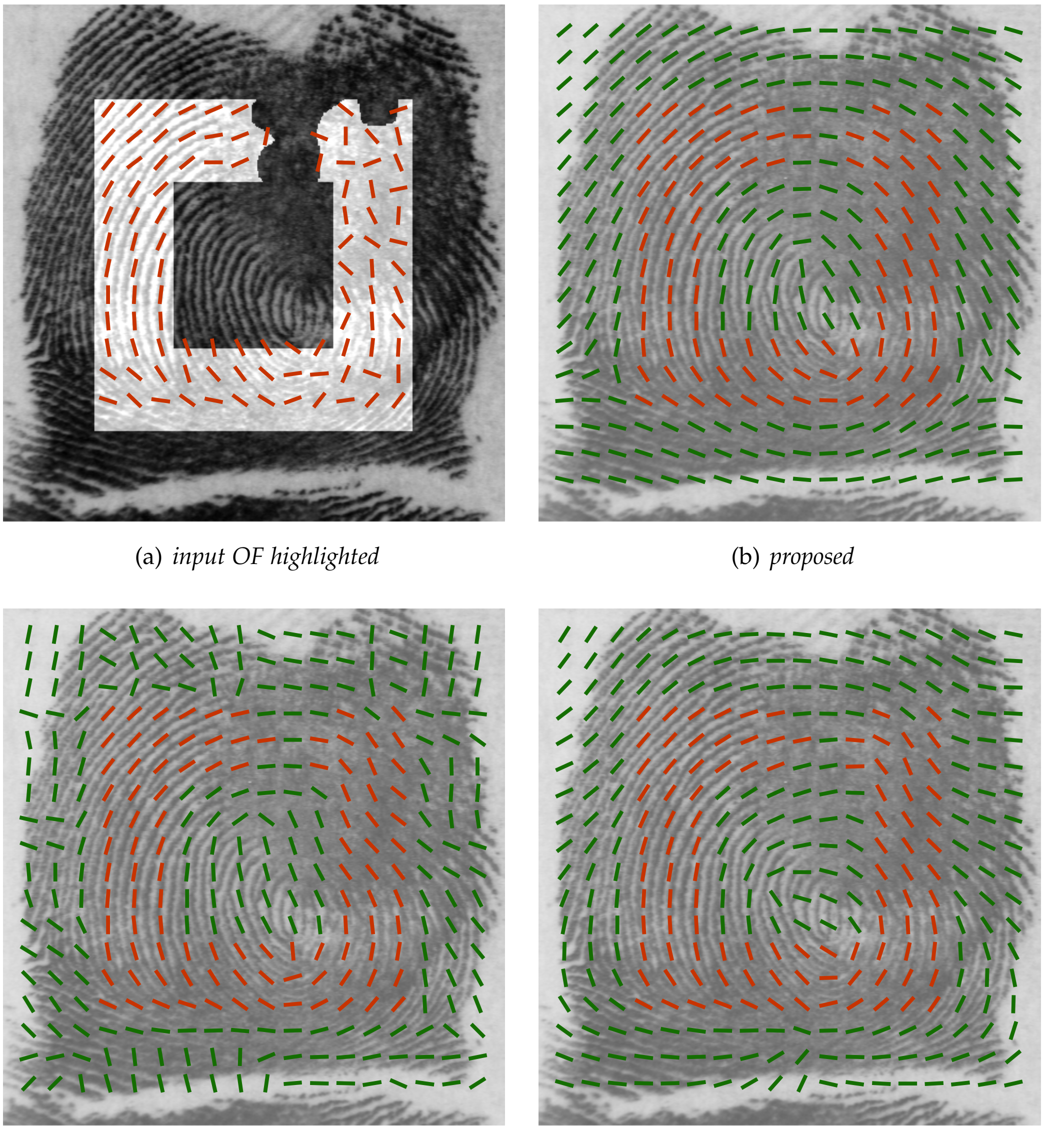
(a) *rotation*

(b) *translation*

(c) *different database, uncentred core*

(d) *different database, segmentation error*

Figure 4.13: **Difficult prediction samples.** Subfigure 4.13(a) shows a 45 degree rotated fingerprint image. In Subfigure 4.13(b) an uncentered loop type fingerprint is shown. Subfigure 4.13(c) shows a slightly rotated and uncentered fingerprint from the FVC2006db2a [20] database. The predicted OF is shown in green.

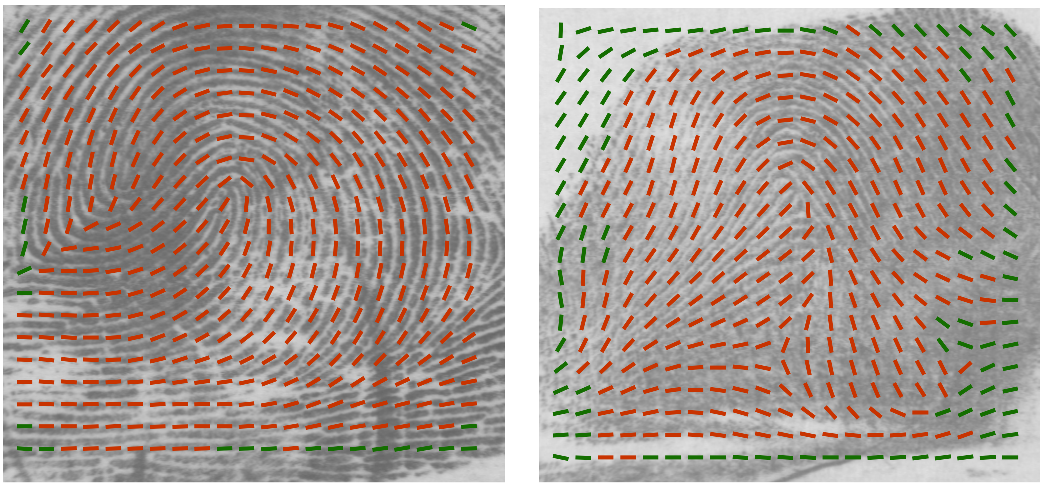

(a) *input OF highlighted*

(b) *proposed*

(c) *FOMFE, 162 parameters*

(d) *Legendre Polynomials*

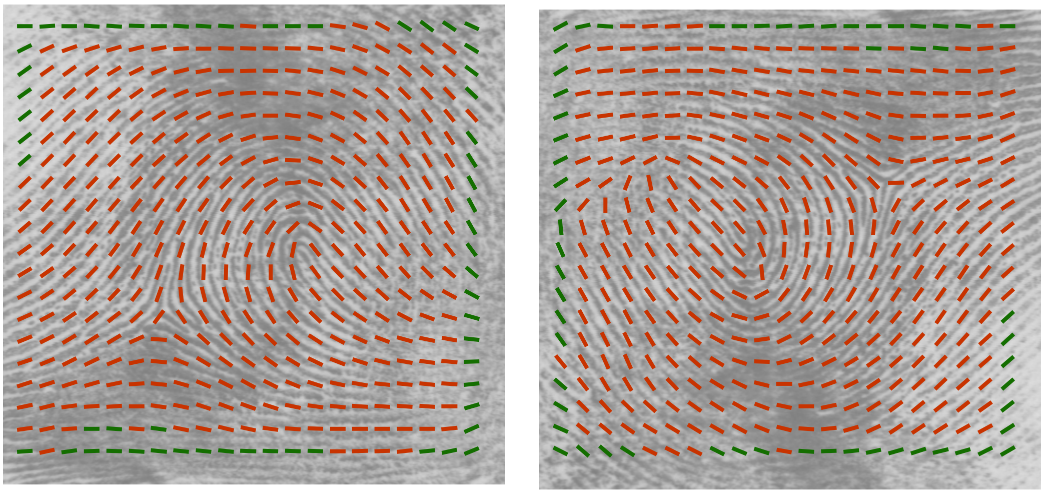
Figure 4.14: **Prediction ability of various methods proposed in literature**. Input OF as shown in 4.14(a), including noise. Green color is used to display the interpolated/extrapolated OF. Subfigure 4.14(b) shows the results of the proposed method. The prediction cabability of the FOMFE approach (exactly implemented as described in [90]) is shown in image 4.14(c). Subfigure 4.14(d) shows results of the approach described in Chapter 3. Note how the proposed method (Figure 4.14(b)) generates the most plausible OF.



(a) *example 1*

(b) *example 2*

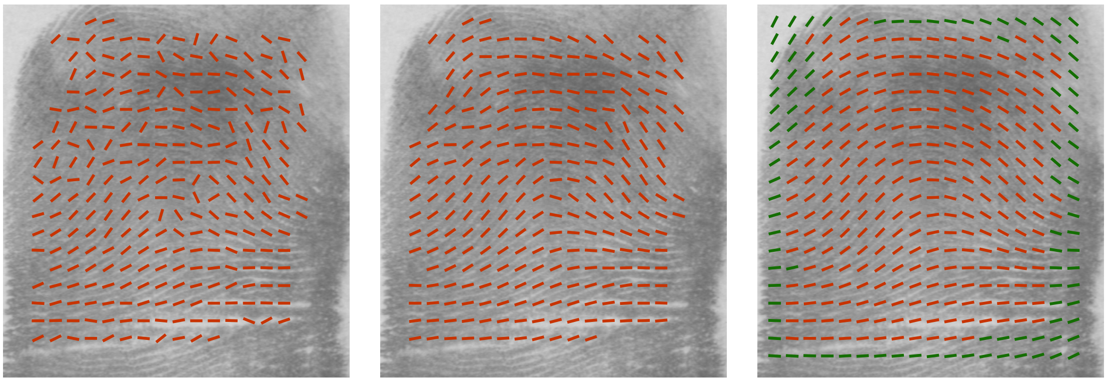
Figure 4.15: **Estimation errors**. In Subfigure 4.15(a) the method failed to generalise the original OF. The twin loop is not correctly modelled. Subfigure 4.15(b) shows the model fitted to a noisy OF. The algorithm could not achieve a plausible fit.



(a) loop type fingerprint

(b) upside down of 4.16(a)

Figure 4.16: **Illegal example test**. In this example we create a 'illegal' fingerprint example by rotating a given fingerprint by 180°. Note how the model corrected the flow pattern to the most plausible type - a whorl type fingerprint.



(a) *input OF* (b) *block based smoothing* (c) *model*

Figure 4.17: **Fitting a noisy OF.** The input image in Subfigure 4.17(a) is processed with commercial fingerprint software (Subfigure 4.17(b)). In Subfigure 4.17(c) the same data has been processed with the proposed method.

Chapter 5

Fingerprint Image Enhancement

Contents

Abstract

Automatic fingerprint identification systems apply enhancement algorithms in order to suppress noise in the input data. As described in the chapters before, one of the most important parameter for enhancement algorithms is orientation. This chapter demonstrates the effectiveness of the fingerprint ridge orientation estimation methods proposed in the previous chapters. For image enhancement we choose a suitable method available in literature and show how it impacts one the error rates of a commercially available fingerprint matcher. The evaluations and comparison made in this chapter emphasize the importance of image enhancement methods and highlight the saliency of fingerprint ridge orientation.

5.1 Introduction

The extractability of both features, SPs and minutiae is heavily depending on successful determination of ridge orientation. In general, there are several types of degradation associated [44] with fingerprint images:

1. the ridges are not strictly continuous, containing gaps and small breaks.
2. parallel ridges are not well separated. This is due to the presence of noise which links parallel ridges, and thus results in their poor separation.
3. cuts, creases, and bruises.

The above mentioned degradations make ridge extraction extremely difficult in highly corrupted regions, particularly resulting in problems during minutiae detection. In noisy images typically a significant number of spurious minutiae are extracted. On the other hand a large number of genuine minutiae are missed due to noise in the input image. Another issue introduced by noise is the precise determination of the minutiae position. To ensure good performance of the ridge and minutiae extraction algorithm in poor quality fingerprint images, an enhancement algorithm is necessary. Generally [44], the fingerprint areas can be divided into three categories (see Figure 5.1 for a visual explanation of the mentioned regions):

1. well-defined region: ridges are clearly differentiated from each other.
2. recoverable region: here ridges are corrupted by a small amount of gaps, creases, smudges links and the like, but they are still visible and the neighbouring regions provide sufficient information about their true structure.

3. unrecoverable regions: ridges are corrupted by such a severe amount of noise and distortion that no ridges are visible and the neighbouring regions do not allow them to be reconstructed.

It should be noted that the above mentioned classification is depending on the enhancement strategy used. Therefore a ’better’ enhancement algorithm may be able to recover regions which otherwise are unrecoverable.

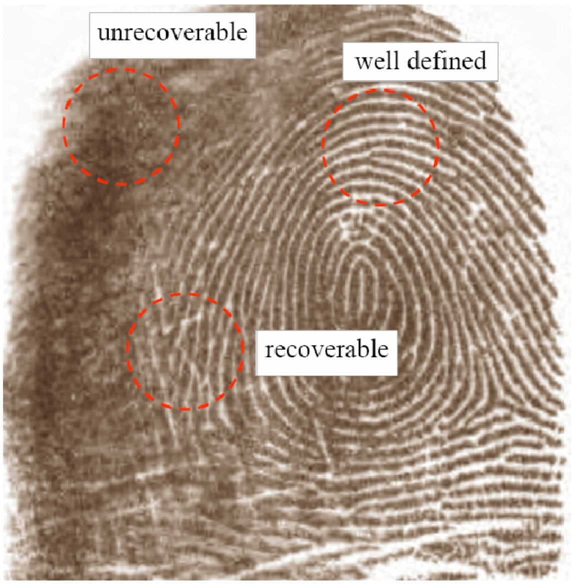


Figure 5.1: **Fingerprint quality categories :** well defined, recoverable (using image enhancement strategies) and unrecoverable regions. Image courtesy of [44].

The above mentioned reasons represented a high motivation for many authors [70, 38, 82, 11, 1, 95, 8, 26] who have demonstrated the successful application of their image enhancement schemas. The principal idea of such image enhancing methods can be summarised [44] as in the following:

- low pass filtering with the aim of linking small gaps and filling impurities due to pores or noise

- band pass filtering in the direction orthogonal to the ridges in order to increase the discrimination between ridges and valleys.

5.2 Related Work: Image Enhancement Techniques

Image enhancement can be seen as a basic step in fingerprint recognition algorithms and is absolutely essential for reconstructing the actual fingerprint pattern as true to the original as possible. The number of publications on this topic is literally endless, which emphasizes its importance. In the following, we will sum up some of the available methods in literature. Basically, the approaches can be classified into **Spatial domain** based and **Fourier domain** based.

Without exception, all described methods ([70,38,82,11,1,95,8,26]) highly depend on the correctly extracted orientation of the considered fingerprint. As already mentioned the extraction of the latter is difficult in areas with low image quality. Allusively, all these papers underline the necessity for ridge orientation modelling.

5.2.1 Fourier domain filtering

In [82] the authors propose filtering in the Fourier domain. The advantage in contrast to spatial domain techniques, which use large convolution masks is mainly execution time. However, the algorithm assumes that the ridge frequency is constant throughout the image in order to prevent a large amount of precomputed filters.

In [11] Sharat et al. introduce a method for fingerprint enhancement based on short time Fourier Transform Analysis. This approach proceeds by analysing the image in local windows and simultaneously computing the orientation and frequency map. In the last step the authors propose the application of so calls raised cosine filters.

5.2.2 Spatial domain filtering

A classic in terms of fingerprint enhancement is the paper of Hong et al. [38]. The authors propose a set of Gabor band-pass filters tuned to the corresponding ridge frequency and ridge orientation. All operations are performed in the spatial domain.

Almansa and Lindeberg [1] propose fingerprint enhancement by application of automatic scale selection mechanisms. The discrete scale space prevents the occurance of artefacts which often plagues blockwise methods. Furthermore the level of detail is adapted to the local image quality.

Weikert [95] proposes the application of coherence-enhancing diffusion filtering for image enhancement. The underlying concept enables true anisotropic behaviour by adapting the diffusion process not only to the location, but also allowing variable smoothing depending on the directions. This smoothing schema has been adopted by Cheng et al. [8] and extensively evaluated using an AFIS.

A method based on a so called Reaction-Diffusion System is described in [42]. In this paper the authors exploit the fact that non-linear reaction diffusion dynamical systems can be used to generate biological textures, including fingerprint like formations. The proposed method combines ridge orientation estimation and an adaptive digital reaction diffusion system in order to enhance a given image of fingerprints.

A very recent approach towards image enhancement is given by Fronthaler et. al [26]. The authors propose bandpass filtering using a multi level pyramid. The typical ridge-valley flow is coherence enhanced by using directional averaging at the structure tensor direction at each level of the pyramid.

5.3 Proposed Method

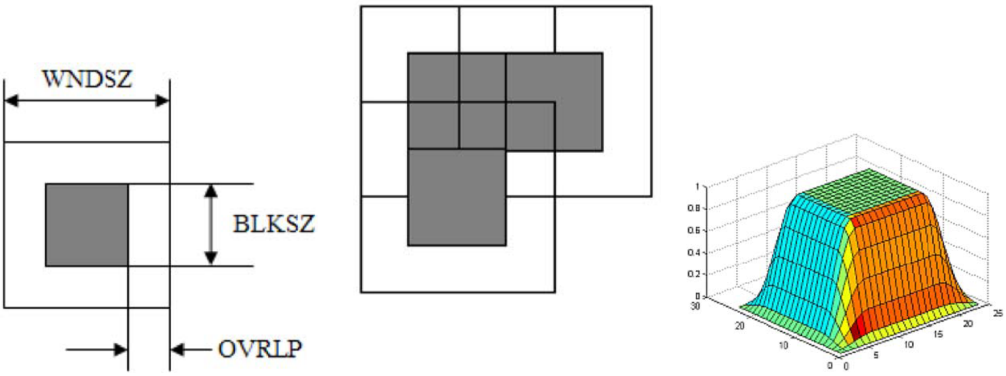
For enhancing fingerprint images we use the above mentioned subsequent combination of two methods, one for band pass filtering and another one for directional averaging. Enhancement algorithm outline:

1. **Fourier domain based band pass filtering:** In order to suppress noise (of very high or very low frequency in comparison to the ridge frequency and/or different orientation than the computed one) we use a block based band pass filtering. The Fast Fourier Transform (FFT) is applied for each block. A butterworth filter in the temporal domain is used for filtering. Each overlapping block is filtered using the given orientation.

2. **PDE based directional smoothing:** In order to prevent block artefacts the directional smoothing is applied using a partial differential equation (PDE). Such a method is more suitable for linking small gaps than block based methods. Especially important, is the possibility to adapt the amount of 'blurring' depending on the curvature of the fingerprint. Usually, low curvature areas allow more smoothing while high curvature areas need less smoothing.

5.3.1 Fourier domain based band pass filtering

The chosen approach for band pass filtering is similar to Chikkerur et al. [11], where during a short time Fourier transform (STFT) analysis, the image is divided into overlapping windows. The overlapping preserves the ridge continuity and reduces block effects common with block processing image operations. It should be noted, that unlike regular Fourier transform, the result is depending on the choice of the spectral window. Since we are interested in enhancing and reconstructing the fingerprint image directly



(a) window parameters (b) overlap (c) spectral window

Figure 5.2: **Window parameters used in the STFT analysis [11].**

from this Fourier analysis, the choice of windows is restricted. In order to provide suitable reconstruction during enhancement, we use a raised cosine window that tapers smoothly near the border and is unity at the center of the window. The raised cosine window is obtained using:

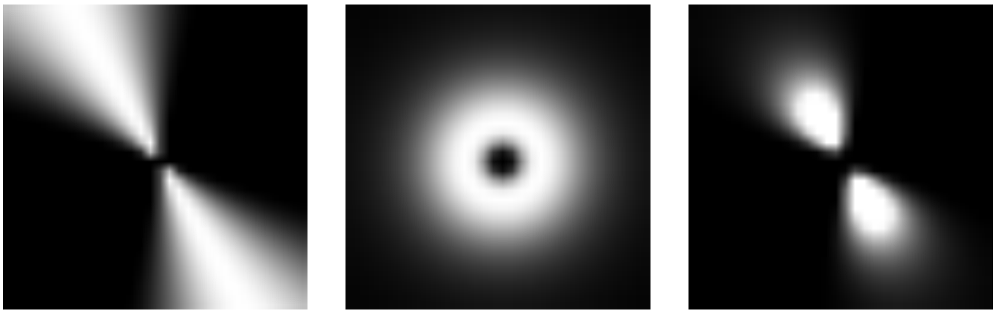
$$W(x,y) = \begin{cases} 1 & \text{if } (|x|,|y|) < \text{BLKSZ}/2, \\ \frac{1}{2}(1 + cos\frac{\pi(x - BLKSZ/2)}{OVRLP}) & \text{otherwise.} \end{cases} \tag{5.1}$$

where BLKSZ is the blocksize and OVRLP the overlapping of the window in pixels. Figure 5.2 shows a thumbnail sketch of the overlapping windows and the raised cosine window. Our values for WNDSZ are 32, for BLKSZ = 10 and for OVRLP = 11.

The directional bandpass filter can be expressed as a separable function [82]:

$$H(\rho,\phi) = H_{radial}(\rho)H_{angle}(\phi) \tag{5.2}$$

in order to allow independent manipulation of its directional and frequency responses $H_{angle}(\phi)$ depends on the local ridge orientation, while $H_{radial}(\rho)$ depends on the ridge frequency. Any good classical bandpass filter would be adequate for $H_{radial}(\rho)$. The Butterworth was suggested by Sherlock and Monroe in [82] because its implementation is simpler than any alternatives. The expression for this filter is:



(a) angular filter $H_{angle}(\phi)$ (b) radial frequency filter $H_{radial}(\rho)$ (c) combined $H(\rho, \phi)$

Figure 5.3: **Bandpass filtering.** Each block of the fingerprint image is filtered by a frequency selective bandpass filter. Note that the bandpass is orientation selective.

$$H_{radial}(\rho) = \sqrt{\frac{(\rho \cdot \rho_{BW})^{2n}}{(\rho \cdot \rho_{BW})^{2n} + (\rho^2 - \rho_0^2)^{2n}}} \tag{5.3}$$

where ρ_{BW} and ρ_0 are the desired bandwidth and centre frequency. A value of n=2 worked well and used throughout all experiments. For ρ_{BW} we use a value of 20 and for ρ_0 a value of 14. Note that these values should be adjusted for the spatial resolution of the given fingerprint sensor. We used this rather generic values for all our experiments.

For $H_{angle}(\phi)$ the following function was used [82]:

$$H_{angle}(\phi) = \begin{cases} \cos^2 \frac{\pi(\phi - \phi_c)}{2\phi_{BW}} & \text{if } |\phi| < \phi_{BW} \\ 0 & \text{otherwise} \end{cases} \tag{5.4}$$

where ϕ_{BW} is the 'angular bandwidth' of the filter, i.e. the range of angles for which $|H_{angle}| \geq 0.5$ and ϕ_c is the model based computed orientation. The directional bandpass filter is shown in Figure 5.3. Throughout all the experiments we used $\phi_{BW} = 0.5$.

5.3.2 PDE based directional smoothing

We adapt an anisotropic diffusion filtering method which uses a diffusion tensor to evolves the initial image under an evolution equation of type:

$$\begin{aligned} \partial_t u &= \operatorname{div}(D\nabla u) \\ u(x,y,0) &= u_0 \end{aligned} \tag{5.5}$$

where $u(x,y,t)$ is the evolving image, t is the diffusion time, and D is the diffusion tensor, a positive definite symmetric matrix that is adapted on the local ridge curvature.

D can be given as:

$$D = \begin{pmatrix} w_1 & w_2 \end{pmatrix} \begin{pmatrix} c & 0 \\ 0 & c \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \tag{5.6}$$

where c represents the diffusion strength and w_1 and w_2 describe the orientation of the diffusion. Note that the orientation O is assumed to be given for this pre-processing step. Then w_1 and w_2 can be computed directly by:

$$w_1 = cos(O) \tag{5.7}$$

$$w_2 = sin(O) \tag{5.8}$$

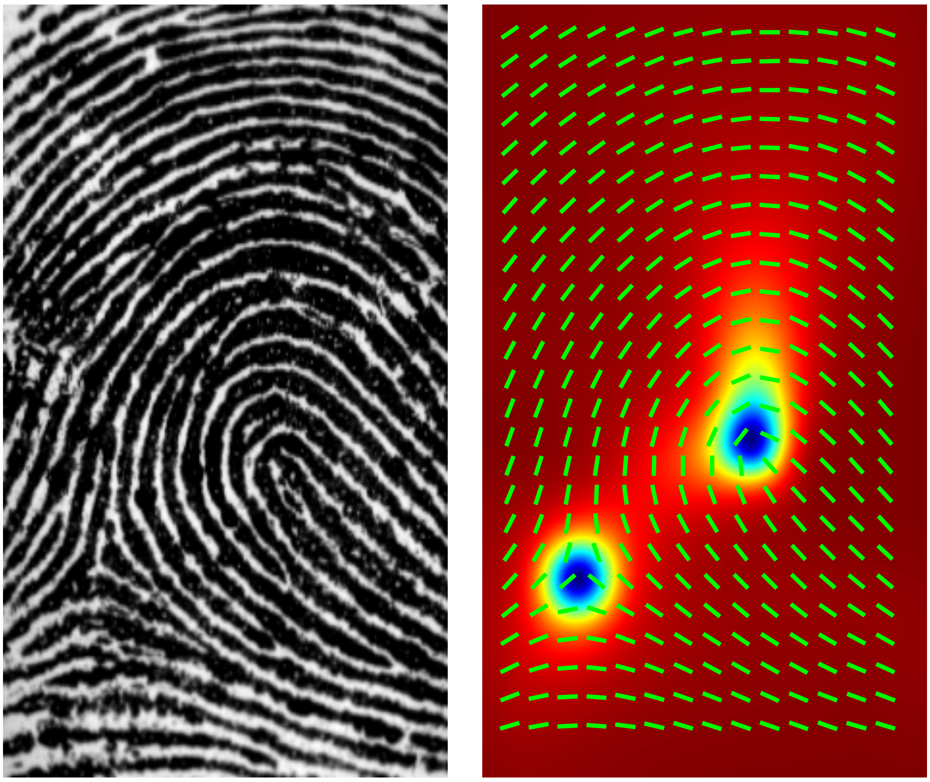
The diffusion strength c is necessary in order to adapt the amount of smoothing depending on the curvature. Typically, the directional smoothing should be lower at high curvature areas (singular points) and higher at low curvature areas. For computing the diffusion strength we use the coherence (computed from the given orientation) as described in [48]. This measure quantifies how well all orientation vectors share the same orientation and is defined in the range $[0,1]$:

$$c = \sqrt{J_1^2 + J_2^2} \tag{5.9}$$

where J_1 and J_2 can be computed by convolving the vectorial orientation by using an Gaussian G_ρ.

$$\begin{aligned} J_1 &= cos(2O) * G_\rho \\ J_2 &= sin(2O) * G_\rho \end{aligned} \tag{5.10}$$

Note that the coherence computed from orientation vectors needs no normalization as opposed to the coherence computed from image gradients as in [48]. We used a standard



(a) fingerprint (b) diffusion strength and orientation

Figure 5.4: **Diffusion strength** c**.** Lower values (blue) for the diffusion strength are used in regions of high curvature in order to preserve the image details. In low curvature areas (red) the diffusion strength is increased. The orientation of the diffusion is computed using the proposed method as described in chapter 3.

deviation of $\rho = 12$ for the Gaussian convolution mask in all our experiments.

Note that the proposed method for directional smoothing resembles Weikert's [95] 'coherence enhancing diffusion'. Weikert embeds the structure tensor approach into a non-linear diffusion process, where the diffusion process is steered by the diffusion tensor. In our method the diffusion tensor is not directly computed from the image but instead from the given orientation.

Numerical Issues

For the numerical approximation of the described partial differential equation (PDE), one needs to replace the derivatives by finite differences. The PDE has the structure:

$$\partial_t u = \sum_{i,j=1}^{m} \partial_{x_i}(d_{ij}\partial_{x_j} u), \tag{5.11}$$

the simplest discretization is given by the finite difference scheme:

$$\partial_t u = \sum_{i,j=1}^{m} L_{ij}^k U^k. \tag{5.12}$$

In this notation, U describes a vector containing the values at each pixel. The upper index denotes the time level and L_{ij} is a central difference approximation to the operator $\partial_{x_i}(d_{ij}\partial_{x_j})$. Equation 5.12 can be rewritten as

$$U^{k+1} = (I + \Delta t \sum_{i,j=1}^{m} L_{ij}^k) U^k \tag{5.13}$$

U^{k+1} can be calculated explicitly from U^k without any matrix inversions. For this reason this scheme is called explicit scheme (for more details see [2]).

The problem adhered to such an explicit scheme is that it requires very small time steps (high runtime!) in order to be stable. Therefore, it is suggested in literature [94,95] to use an implicit scheme having the same first-order Taylor expansion in Δt, but better stability properties. One possibility is the so called 'additive operator splitting stabilized scheme' (AOS scheme), which has been introduced in [94] as an efficient and reliable method for isotropic non-linear diffusion filtering and which performs well on parallel computer architectures. Equation 5.14 shows one way how to extend the AOS-scheme to anisotropic processes with a diffusion tensor:

$$U^{k+1} = \frac{1}{m} \sum_{l=1}^{m} (I - m\Delta t L_{ll}^k)^{-1} \cdot (I + \Delta t \sum_{i=1}^{m} \sum_{j \neq i} L_{ij}^k) U^k \tag{5.14}$$

This method achieves a stabilization through the non-negative matrices $(I - m\Delta t L_{ll}^k)^{-1}$. They describe a semi-implicit discretization of the diffusion caused by the l-th diagonal entry of the diffusion tensor. The typically necassary step size $\Delta t = 2$ is about an order of a magnitude larger than the ones for of the classic explicit scheme. For the standard approximations with central derivatives within a (3x3)-stencil, the matrix inversions in Equation 5.14 comes down to solving diagonally dominant tridiagonal systems of linear equations. This can be performed in linear complexity with a modified Gaussian algorithm (Thomas algorithm, see [93,2]).

The final algorithm is as follows:

1. Calculation of the diffusion tensor in each pixel.
2. Calculation of:
$$V^k = (I + \Delta t \sum_{i=1}^{m} \sum_{j \neq i} L_{ij}^k) U^k$$
3. For l=1..m: calculation of
$$W_l^{k+1} = (I - m\Delta t L_{ll} k)^{-1} V^k \tag{5.15}$$
by means of the Thomas algorithm ([93]).
4. Calculation of
$$U^{k+1} = \frac{1}{m} \sum_{l=1}^{m} W_l^{k+1}. \tag{5.16}$$

This algorithm has proven to be of low computational complexity. We apply 30 iterations of this algorithm in our Matlab implementation. Using a good C/C++ implementation and exploiting multiple cores on a state of the art computer run times are reported to be less than 70 ms for a 512x512 image.

Figure 5.5 shows an example fingerprint enhanced using the proposed method.

5.4 Evaluation

The performance of the enhancement algorithm is assessed on the FVC2004 db3 and FVC2006 db2 dataset. We demonstrate that incorporating the enhancement algorithm in the a commercial fingerprint verification system improves the system performance. Figure 5.6 5.7 show the ROC curves which demonstrate the improvements made using the proposed method.

For the FVC2004db3a the original matcher's EER is 2.38%. Using Legendre polynomials as described in Chapter 3 we can decrease the EER to 1.81 (24% improvement). Additionally, using the mentioned enhancement method we can decrease the EER to 1.61% which represents a relative improvement of 33%.

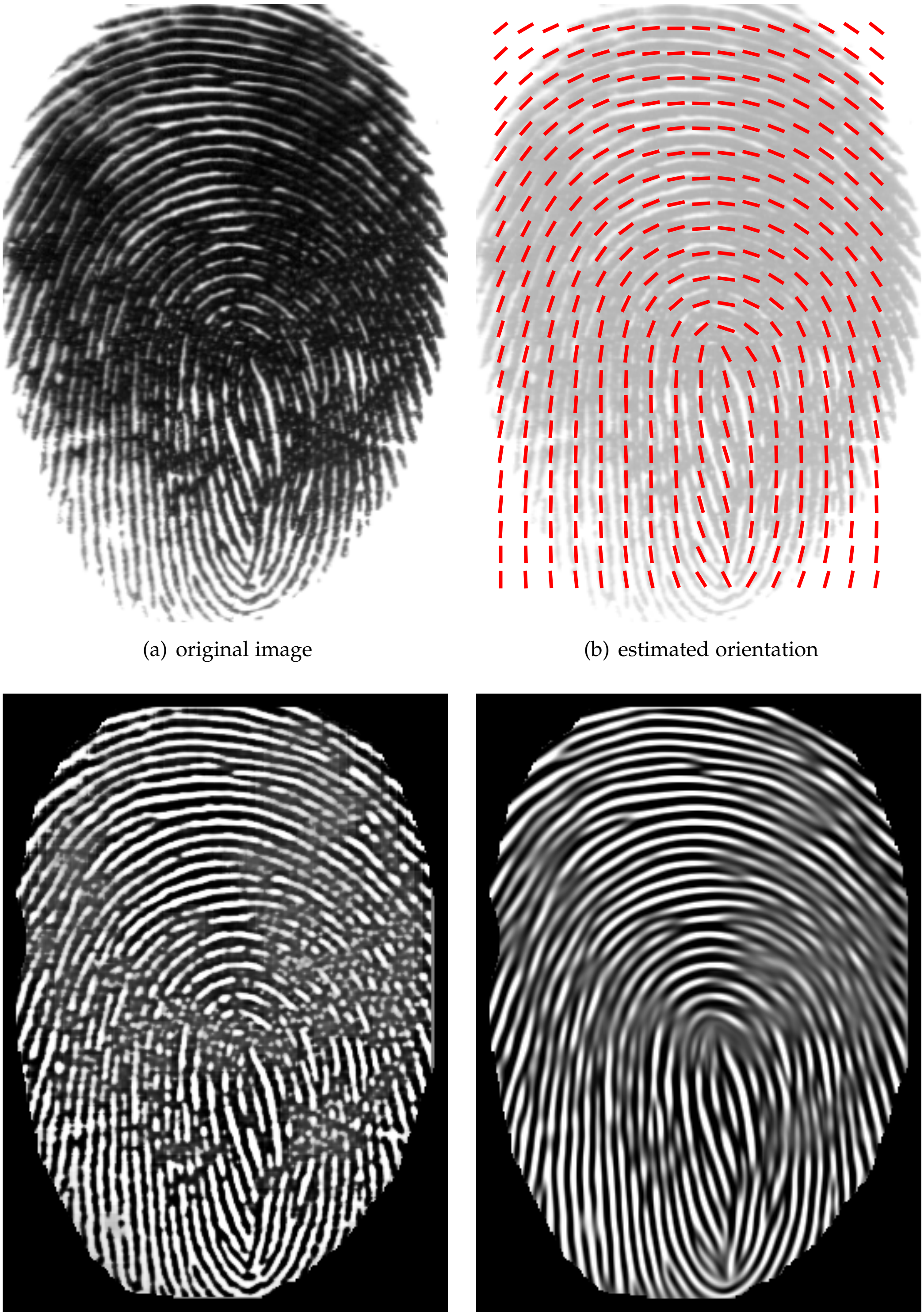
When applying the proposed enhancement strategy to the FVC2006db2a dataset, the EER can be reduced to 0.136%. While the original EER was 0.284% we could reduce it 0.18% using the ridge orientation model only. The relative improvement using the model was 37%, using the model and the proposed enhancement strategy we can more than half (52%) the EER or the original matcher.

5.5 Conclusion

In this chapter we have applied a pre-enhancement step to a commercial fingerprint verification system. The proposed enhancement algorithm was chosen to the following reasons. First, it should be noted that ridge frequency estimation as often proposed in literature, is very difficult (if not impossible) to realize in low quality regions. Therefore, the ridge frequency filtering is done for a given possible ridge frequency range. In our implementation this is achieved by bandpass filtering of the image in the Fourier space. Note that this orientation sensitive filtering adds very little additional computational costs to the final matching algorithm as the Fourier transformation is already part of most commercial AFIS (eg. for image quality estimation). Second, we apply directional smoothing using a partial differential equation (PDE). This method is chosen in order to prevent blocking effects and to reduce spurious minutiae detection, a common problem with many other filtering methods. Furthermore, the PDE allows an adoption of the smoothing strength depending on the curvature and scale of the fingerprint.

Both mentioned filtering steps, necessarily make use of the fingerprint's ridge orientation. We have performed several evaluation on publicly available databases. These experimental results show that the enhancement algorithm is capable of improving the recognition rates.

Note that the algorithm can be further improved by excluding unrecoverable regions, which can be detected during the enhancement step.



(a) original image

(b) estimated orientation

(c) bandpass filtered

(d) directionally smoothed

Figure 5.5: **Enhancement example.** The original image (a) is taken from the fvc2004 database 1a and contains ridges which can not be well discriminated from the valleys. The orientation in Subfigure (b) is estimated using the model based procedure as described in chapter 3. The bandpass filtering results in the image (c). The final enhancement including the directional smoothing is shown in (d).

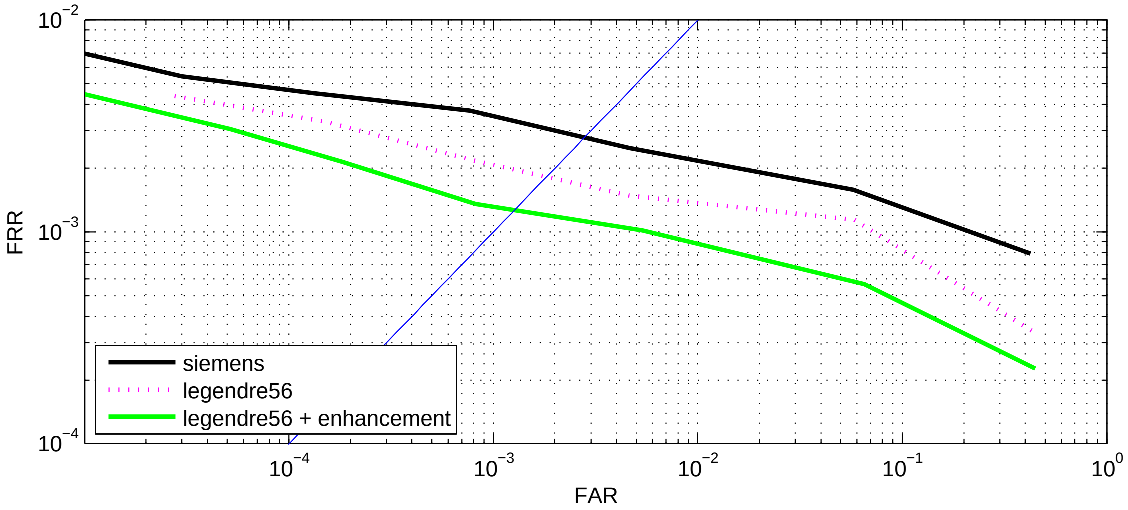


Figure 5.6: **ROC for fvc2006db2a.** This figure shows the improvements made using the proposed enhancing method. The improvements using the orientation model from chapter 3 are 37%. Using the proposed enhancement we can increase this rate to 52%.

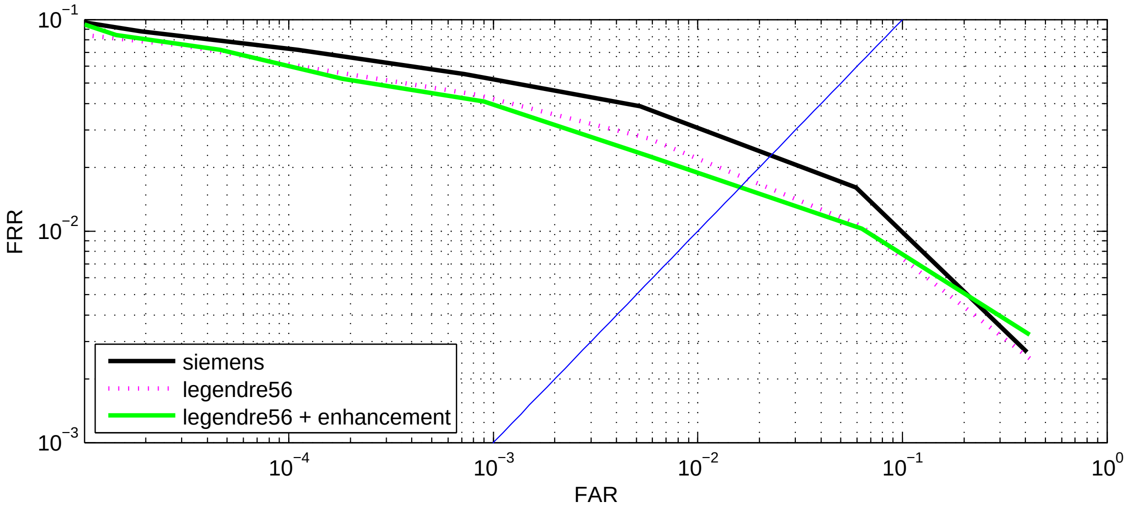


Figure 5.7: **ROC for fvc2004db3.** This figure shows the improvements made using the proposed enhancing method. The improvements using the orientation model from chapter 3 are 24%. Using the proposed enhancement we can increase this rate to 33%.

Chapter 6

Thesis Conlusion

6.1 Summary

In this PhD thesis we have studied fingerprint ridge orientation modelling. This section gives a summary of the individual chapters.

Chapter 1

The goal of the first chapter is to give a general introduction on biometrics with focus on fingerprint verification. It is explained how a fingerprint verification system extracts distinctive features and uses these features for 'matching' two fingerprints. One of the main modules of a fingerprint verification system, namely fingerprint ridge orientation estimation, is described in detail. We will describe the role of this module and discuss the motivations behind fingerprint ridge orientation modelling. These motivations are mainly concerned with lower error rates, higher compression ratios and lower processing times. In the last part of this chapter an outline of the thesis is given.

Chapter 2

This chapter discusses prior work related to this thesis. First, we will give an overview of the orientation estimation process. We will discuss the different representations of orientation, which are necessary because orientation is a π cyclic quantity. In the second section of this chapter we give a list of available fingerprint ridge orientation models in literature. This literature will be classified based on different aspects. The third part of this chapter shows a deep analysis of the standard orientation averaging method which is based on vectorial orientation smoothing. Furthermore, we will discuss the Bias-Variance trade-off applicable to the orientation smoothing.

Chapter 3

Based on the findings from the previous chapter, Chapter 3 proposes a novel method for fingerprint ridge orientation modelling. One of the main problems the presented method addresses is smoothing orientation data while preserving details in high curvature areas, especially singular points. We show that singular points, which result in a discontinuous orientation field, can be modelled by the zero-poles of Legendre Polynomials. The method proceeds in two steps. In the first step we roughly estimate the parameters using a closed form solution. In the second optimization step, we propose to use a non-linear optimization technique for more precise parameter estimation. We showed, that only five iterations of the Levenberg-Marquardt algorithm lead to a significantly improved orientation field. Another advantage of the proposed method is a extremely compact representation of the orientation field, using only 56 coefficients. We have carried out extensive experiments using a commercial fingerprint matcher and a singular point detector. Moreover, we compared the proposed method with other state-of-the-art fingerprint orientation estimation algorithms. We can report significant improvements in both - singular point detection as well as matching rates.

Chapter 4

In this chapter we apply a priori knowledge to the fingerprint ridge orientation models as described in the previous chapter. In the first part of this chapter, we will describe the idea of flexible templates models. This models, usually used for shape analysis in literature, can be used to apply prior knowledge within pattern recognition algorithms. Furthermore, we will give an overview of these methods and describe some of the available techniques. Using a priori knowledge, the OFs can be constrained by the model to vary only in ways as they occur in nature. We will describe how such a method can be used to estimate OFs in noisy fingerprints as well as to interpolate larger OF parts. The proposed method uses Active Fingerprint Ridge Orientation Models (AFROMs) which iteratively deform to fit an OF of a fingerprint. Our method does not depend on any pre-alignment or registration of the considered images. The training can be done fully automatic without any user interaction. Furthermore, our method does not depend on any other computed data, except a segmentation. We give a exhaustive evaluation - the generalisation as well as the prediction capability of the proposed method is analysed.

Chapter 5

Automatic fingerprint identification systems apply enhancement algorithms in order to suppress noise in the input data. As described in the chapters before, one of the most important parameter for enhancement algorithms is orientation. This chapter demonstrates the effectiveness of the fingerprint ridge orientation estimation methods proposed in the previous chapters. For enhancing fingerprint images, we choose a suitable method from literature. In order to suppress artefacts we use block based band pass filtering. This is done by applying the Fast Fourier Transform for each block. A Butterworth filter in the temporal domain is used for the final filtering. Each overlapping block is filtered by using the computed orientation. In a second step, directional smoothing is applied using a partial differential equation. The proposed 'continuous' method is more suitable for linking small gaps than block based only methods. Especially worth mentioning is the possibility to adapt the amount of 'blurring' depending on the curvature of the fingerprint. The evaluations and comparison made in this chapter clearly underline the importance of image enhancement methods and highlight the saliency of fingerprint ridge orientation. Finally, it should be noted that all experiments carried out have been performed using state of the are commercial fingerprint software.

6.2 Main Contributions

This section emphasizes on the research contributions of this PhD Thesis.

- **Literature review:** Fingerprint ridge orientation modelling methods are studied for over 30 years, this thesis gives a deep analysis of related work. We are using profound issues for the partitioning of this literature into several classes. The underlying concepts, as well as the advantages and disadvantages of these methods are discussed. This literature review is given in Chapter 2.

- **Novel methods:**

  - A novel method for higher (compared to state of the art methods) fidelity in fingerprint ridge orientation estimation is presented. One of the main problems the proposed method addresses is smoothing orientation data while preserving details in high curvature areas, especially around singular points. We show that singular points, which result in a discontinuous orientation field, can be modelled by the zero-poles of orthogonal polynomials. The models parameters are obtained in a fast two staged optimization procedure. This part is described in Chapter 3.

  - Another contribution of this thesis is the application of **a priori knowledge** in fingerprint orientation models. Available methods construct a model by applying a suitable analytic expressions (e.g. differential equations with trajectories similar to fingerprint flow patterns). These 'hand crafted' analytic expressions together with the suggested heuristics are very complex while still being of limited usefulness. This is due to the fact that the mechanisms which give rise to variability are insufficiently well understood to allow a theoretical model of deformability to be proposed. Starting from the theoretical framework of flexible templates models, we developed a method which constraints the fingerprint orientations to vary only in ways as they occur in nature. The proposed procedure statistically 'learns' specific patterns of variability from a representative training set of the OFs. Furthermore, we want to note that the proposed model is one of the rare models which can be used to model all classes of fingerprints, including arch type fingerprints. Arch type fingerprints are difficult to model because they do not contain singular points. Another advantage of the method is that the user does not need to label any samples. Instead, the statistically relevant 'ingredients' are extracted from large enough sample database (a few hundred images). This contribution is given in Chapter 4.

- **Evaluation:**
  - For evaluating the proposed method we perform feature extraction on publicly available databases. For testing the improvement in orientation extraction, we replace the orientation field of a state of the art fingerprint matcher with a orientation field computed using the proposed method. The matching results show statistically significant improvements.
  - Current literature does not have an established 'standard' for evaluating SP detections. This thesis presents a systematic method for the evaluation of SP detections, therefore creating a common basis for comparing two different singularity detectors without depending on specific thresholds.

In conclusion, the fingerprint orientation model proposed in this PhD thesis provides a clear improvement over prior models. This improvements are backed up by the extensive studies conducted in this thesis. Further research might however be needed to allow for their wide-spread applicability.

Appendix

6.3 Weighted Pseudoinverse Technique for Least Squares Approximation

For the approximation of a discrete two dimensional function $f(x, y)$ we propose to use a linear combination of n basis functions. Then for every point $\mathbf{x}_i = (x_i, y_i)$ the following equation can be evaluated:

$$f(x, y) \approx \sum_{j=0}^{n} a_j \phi_j(x, y) \tag{6.1}$$

Let

$$\Phi(\mathbf{x_i}) = [\phi_0(\mathbf{x}) \quad \phi_1(\mathbf{x}) \quad \dots \quad \phi_n(\mathbf{x})] \tag{6.2}$$

be the row vector containing the set of basis functions $[\phi_0(\mathbf{x}) \quad \phi_1(\mathbf{x}) \quad \dots \quad \phi_n(\mathbf{x})]$ evaluated for a given coordinate $\mathbf{x} = (x, y)$, the system matrix $\mathbf{V}$ can be defined as:

$$\mathbf{V} = \begin{pmatrix} \Phi(\mathrm{x}_1) \\ \Phi(\mathrm{x}_2) \\ \vdots \\ \Phi(\mathrm{x}_i) \end{pmatrix}$$

Using this expression, we can complete the system matrix to:

$$\mathbf{V} = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \dots & \phi_n(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \dots & \phi_n(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_i) & \phi_1(\mathbf{x}_i) & \dots & \phi_n(\mathbf{x}_i) \end{pmatrix} \tag{6.3}$$

Where the size of the system matrix $\mathbf{V}$ is determined by the number of coordinate points i and the number of basis functions n. Further, we can write the parameter vector as:

$$\mathbf{a} = [a_1, a_2, \ldots, a_n]^T \tag{6.4}$$

and the vector of observed function values $\mathbf{f}$ as:

$$\mathbf{f} = [f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_i)]^T \tag{6.5}$$

where $f(\mathbf{x}_k)$ is the observed function value at the coordinate $\mathbf{x_k}$. We use the method of least squares to model the numerical data $\mathbf{f}$. The best fit is characterized by the least value of the sum of squared residuals $\mathcal{F}$. Furthermore, a weight ω to every pixel $\mathbf{x} = (x, y)$ is assigned because not all points are of equal value in determining a solution. Using this convention, we can write:

$$\mathcal{F} = \sum_{j=1}^{i} \omega_j [\Phi(\mathbf{x}_j)\mathbf{a}^T - f(\mathbf{x}_j)]^2. \tag{6.6}$$

Since the number of data points (and thus equations) is much larger than the number of basis functions, we use the pseudoinverse technique to estimate a solution [28, 73]. The solution vector $\mathbf{a}$ can be obtained as following:

$$\mathbf{a} = (\mathbf{V}^T\mathbf{W}\mathbf{V})^{-1}\mathbf{V}^T\mathbf{W}\mathbf{f} \tag{6.7}$$

Where $\mathbf{W} = diag(\omega_1, .., \omega_i)$ is the diagonal weighting matrix containing the weights for every coordinate.

6.4 Averaging Orientation Data

Rao and Schunk

This section is based on [77]. Rao and Schunk [77] mention that it is not possible to sum up directions vectorially in order find the average of two directions. This will not work for two reasons. Firstly, any given line segment does not have a unique direction, since it could be taken to point either in the direction ϕ or $\phi + \pi$. Secondly, even if the line segments (ridges and valleys in fingerprint images) were assigned directions, there is the danger that segments pointing in opposite directions will cancel each other out, instead of influencing the choice of dominant orientation as they should. One way to tackle this problem is as follows. Assume that the pixels are indexed by the subscripts i, where i ranges from 1 to N, the number of pixels. Consider a line oriented at an angle ϕ as shown. Let j^{th} segment subtend an angle ϕ_j. The next few steps will show that it does not matter what sense this angle is taken in, i.e. it is immaterial as to what vector direction one chooses for the line segment. Let R_j be the length of the j^{th} segment. The projection of this segment onto the line is $R_j \cos(\phi_j - \phi)$. Consider the sum of the absolute value of all such projections,

$$S_1 = \sum_{j=1}^{j=N} ||R_j \cos(\phi_j - \phi)|| \tag{6.8}$$

S_1 varies as the orientation of the line ϕ is varied. That value of ϕ which maximizes S_1 is the dominant orientation of the given set of line segments. Thus, one can evaluate the dominant orientation by maximizing S_1 with respect to ϕ. Since the absolute value function is not differentiable everywhere, one can equivalently maximize the sum S_2, where

$$S_2 = \sum_{j=1}^{j=N} R_j^2 \cos^2(\phi_j - \phi) \tag{6.9}$$

Here we have taken the sum of the square of the projections. Differentiating Equation 6.9 with respect to ϕ we get

$$\frac{\partial S_2}{\partial \phi} = \sum_{j=1}^{j=N} R_j^2 \cos(\phi_j - \phi) \sin(\phi_j - \phi) \tag{6.10}$$

Setting Equation $\partial S_2 / \partial \phi$ to zero in order to obtain an extremum, we get from the above equation

$$\sum_{j=1}^{j=N} R_j^2 2 \sin(\phi_j - \phi) = 0 \tag{6.11}$$

$$\sum_{j=1}^{j=N} R_j^2 2 \sin 2\phi_j \cos 2\phi = \sum_{j=1}^{j=N} R_j^2 2 \cos 2\phi_j \sin 2\phi \tag{6.12}$$

Hence

$$\tan 2\phi = \frac{\sum_{j=1}^{j=N} R_j^2 \sin 2\phi_j}{\sum_{j=1}^{j=N} R_j^2 \cos 2\phi_j} \tag{6.13}$$

Let $\hat{\phi}$ be the value of ϕ which satisfies Equation 6.13. Instead of finding the sum S_2 at different orientations, Equation 6.13 tells us in a single computation the orientation $\hat{\phi}$ that maximizes S_2 and hence the dominant direction of the pattern of line segments. That $\hat{\phi}$ indeed maximizes S_2 will be proven shortly.

Equation 6.13 has an interesting interpretation. Consider the line segments to lie in the complex plane, each segment being represented by $R_j e^{i\phi_j}$, where R_j is the length of the segment and ϕ_j is its direction. Now square all the segments, which have been represented as complex numbers. Thus each segment will give rise to a term or the form $R_j^2 e^{2i\phi_j}$. If summing up these numbers, the resulting complex number has an orientation α, with respect to the x-axis, given by

$$\tan\phi = \frac{\sum_{j=1}^{j=N} R_j^2 \sin 2\phi_j}{\sum_{j=1}^{j=N} R_j^2 \cos 2\phi_j} \tag{6.14}$$

This equation is the same as Equation 6.13. Interestingly, Witkin and Kass [48] arrived at a similar result. Their method is described in the next subsection. The orientation smoothing method described in Equation 6.13 is one of the most cited method for orientation smoothing.

Witkin and Kass

This section describes the orientation smoothing approach as described by Witkin and Kass in [48]. In their paper they discuss squaring the gradient vectors a necessary step for deriving a smooth orientation vector field. This section is based on [77].

Consider the vector in the complex plane formed by combining the gradients Gx and Gy as $(Gx + iGy)$. Let this vector have the polar representation $Re^{i\phi}$. The square of this vector is $R^2e^{2i\phi}$. Consider the vector $Re^{2i\phi+2\pi} = Re^{2i\phi}$. Hence, squaring gradient vectors that point in opposite directions makes them reinforce each other. This is the basis of the first scheme for combining gradient orientations, and has been proposed by Witkin and Kass [48].

Let $J(i,j)$ denote the squared gradient vector at (i,j). The x component of J is $Jx = Gx(i,j)^2 - Gy(i,j)^2$ and the y component of J is $Jy(i,j) = Gx(i,j)Gy(i,j)$. Jx and Jy are computed from the gradient of the image in this manner. The next step is to smooth Jx and Jy in order to average the orientation estimates over a neighbourhood. This is done using Gaussian filters. Let $Jx^*(i,j)$ and $Jy^*(i,j)$ represent the smoothed squared gradient vector at (i, j). Let ϕ_{ij} be defined by the equation

$$\phi_{ij} = \arctan \frac{Jy^*(i,j)}{Jx^*(i,j)} \frac{1}{2} \tag{6.15}$$

where the arctangent is computed using two arguments and lies in the range $[0, 2\pi)$. The division by 2 occurs because the original gradient vector was squared. The estimated orientation angle at (i, j) is then $\phi_{ij} + \frac{\pi}{2}$, since the gradient vector is perpendicular to the direction of anisotropy.

It should be mentioned, that this method is equivariant to the above mentioned method of Rao and Schunk [77]. This fact can be easily verified by converting the Cartesian coordinates into Polar coordinates.

6.5 The Bias-Variance Tradeoff

This section describes the bias-variance tradeoff and is represents a quote from [87]. The bias-variance tradeoff (also known as bias-variance dilemma) is a very important issue in data modelling. Ignoring it is a frequent cause of model failure and although it has a deep theoretical rooting, it can be explained in simple terms. The phenomenon is not specific to a specific model. In fact, it shows-up under various guises in any kind of model. So, quite generally, the bias-variance tradeoff principle can be stated as follows :

- Models with too few parameters are inaccurate because of a large bias (not enough flexibility).
- Models with too many parameters are inaccurate because of a large variance (too much sensitivity to the sample).
- Identifying the best model requires identifying the proper 'model complexity' (number of parameters)

6.5.1 A model is a set of estimators

We'll use regression as an example. The data is supposed to have been generated by a process:

$$y = f(x_1, x_2, ..., x_p) + \epsilon \tag{6.16}$$

where f is deterministic, and ϵ is random with 0 mean. A regression model $y^* = f^*(x_1, x_2, ..., x_p)$ is built from the sample. Let x_0 be a point of the feature space. Then

$f^*(x_0)$ is hoped to be close to $y_0 = f(x_0)$, the true value of the regression function. Because of the randomness of ϵ, that is, the randomness of the sample, the model depends on the actual sample used to build it. Another sample would have led to a different model, and therefore a different response at x_0. So the response of a model at any point is a random variable. In the terminology of Statistics, such a regression model therefore puts at each and every point x_0 of the feature space a random variable that is an estimator of y_0, the true value of the regression function at this point. This estimator is denoted by $f^*(y, x = x_0)$, or f_0^* for short.

6.5.2 Bias-Variance decomposition

We here focus on the response error at x_0 (although a more general study can be conducted on the entire space, taking into account the unconditional probability distribution $p(x)$. The estimator f_0^* is good if its realizations are close to the true value y_0 in a probabilistic sense, that is, for instance, if its Mean Square Error (MSE):

$$\text{MSE} = E[(f_0^* - y_0)^2] \tag{6.17}$$

is small. It is easily shown that:

$$\text{MSE} = \text{Bias}^2 + \text{Variance} \tag{6.18}$$

where 'Bias' and 'Variance' are that of the response of the model, considered as an estimator of y_0. So the errors made by a model have two origins:

- The Bias, that measures how far the model response f *(x0) is from the value f(x0) of the true regression function on the average (that is, over all possible samples),
- The Variance, that measures how sensitive f *(x0) is to the particular sample that was used for building the model.

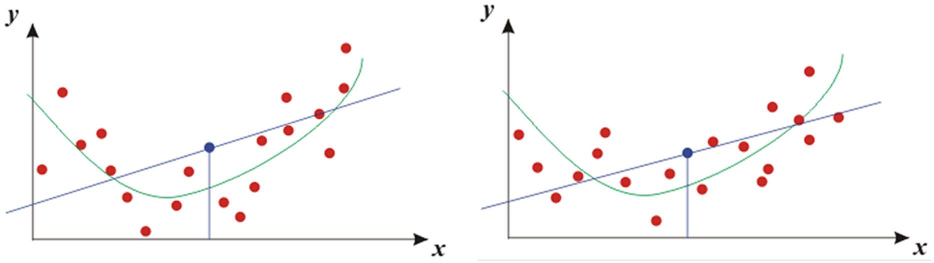
It is never the case that a data set makes obvious the choice of a particular model architecture. The analyst will always consider several candidate models, and his goal is of course to select the model with the most accurate response (on new data). For example, in the case of regression, it is common to have many candidate independent variables (the regressors). A large part of the effort of model building will consist in identifying an adequate subset of regressors to be incorporated into the model. But each subset of regressors will yield a model, so a family of models is to be considered. At point x_0, each of these models will have its own bias, its own variance, and therefore its own error level (MSE).

6.5.3 The Bias-Variance Tradeoff

The bias-variance tradeoff principle states that within a given family of models :

- A model with a low bias has a large variance.
- A model with a low variance has a large bias.
- The best model (lowest MSE) in the family will have neither a very low bias, nor a very low variance.

Identifying this best model with certainty is of course impossible, as this would require knowing the true regression function $f(x)$. But attempts can be made to identify models which are probably good. This is the object of 'model selection' (see below).



(a) Sample 1

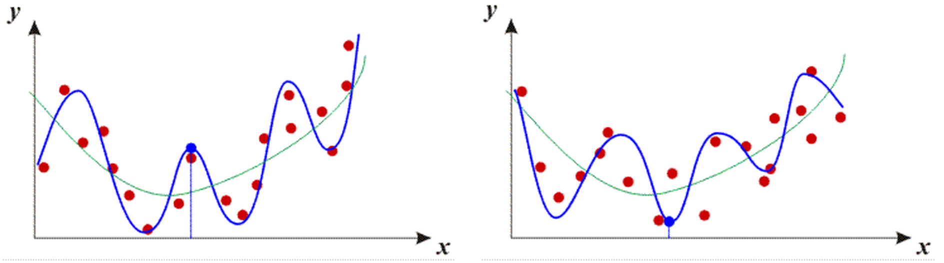
(b) Sample 2

Figure 6.1: **Bias.** If the data is highly non-linear, a low degree polynomial (order 1, shown in blue in this illustration) will not have the flexibility needed to capture the global shape of the distribution. The polynomial line will be most of the time far from the data points, leading to large errors. The model is then said to have a large bias because the bias of its predictions for a given x (blue dot) is high. On the other hand, because of this very rigidity, the predictions of the model will depend only little on the particular sample that was used for building the model, and will therefore have a low variance (lower image of the illustration below). Images courtesy of [87].

6.5.4 Model complexity

It is convenient to consider the number of parameters (the complexity of the model) as a way to sort models in a family. The bias-variance tradeoff then states that, in the family of models:

- Low complexity models have a low variance but a large bias.
- High complexity models have a low bias but a large variance.



(a) Sample 1

(b) Sample 2

Figure 6.2: **Variance.** But too large a degree will make this polynomial line very sensitive to the details of the sample. Another sample would have lead to a completely different model, with completely different predictions (lower image of the illustration below). The model is then said to have a large variance because the variance of its predictions (for a given x) is large. In good models, points that are far from the true regression line (green) have a large contribution to the quadratic error. But here, because of the flexibility conferred by its high degree, the polynomial line can now get close to these points (low bias), and the quadratic error measured on the design sample is low. So the model appears to be performing well, but will in fact perform poorly on new data. Images courtesy of [87].

Consequently, the 'best' model will always have a number of parameters that is neither too small nor too large. The analyst will have to find the proper tradeoff between bias and variance within this family of models, largely (but not only) by tuning the number of parameters.

6.5.5 Overparametrization and Overfitting

The true performance of a model is that observed on new data that did not take part to the construction of the model, not the observed performance on the design data.

- For a model that is higly biased because its complexity is too low, these two performances are similar, and both poor.
- But for a model with too large a variance because its complexity is too high (overparametrization), one will observe :
  - Excellent (but meaningless) performance on the design data, and
  - Poor performance on new data because of the high variance of the local estimators. This is called overfitting.

For example, if f^* is chosen in the family of polynomials, then higher degree polynomials (large number of parameters) can get closer to the data points than lower degree polynomials, thus leading to a lower quadratic error. In fact, if the design set contains n data points, it is well known that a n-degree polynom will go exactly through the points, thus reducing the error on the design set to 0. But this polynom undergoes oscillations that are both very large and whose features strongly depend on the exact positions of the points, thus conducing to a model with a huge variance and very large response errors.

Let's insist again: even a moderate overparametrization can cause the variance of the model to grow in an explosive way. Because this phenomenon is masked by excellent performances on the design set, and becomes visible only when it is too late (that is, when the model is put to work on new data), it tends to be overlooked by the newcomer to data modelling.

6.5.6 Sample size

Quite generally, larger samples make for smaller variances. Unfortunately, practical considerations prohibit resorting to arbitrarily large samples to bypass the bias-variance problem. Conversely, small samples make the bias-variance tradeoff even more acute. For a given bias, the variance of the model response is larger than for a model built from a larger sample. The sample size issue is both important and complex as a new concept now steps in: that of the dimension of the data space. Is a 1000-observation sample large or small ?

- Suppose that one wants to make a 10-bin histogram of this 1-dimensional data : then 1000 observations is plentiful, for there will be an average of 100 observations per bin, and the variance of the histogram will be small, and the bias reasonable.

- Suppose now that the data is 3-dimensional (there are 3 variables x_i). One wants to make a 3-dimensional histogram for the purpose of probability density estimation, and wants to maintain the 10-bracket resolution per dimension. The '3D-histogram' will then have 10^3 = 1.000 bins (cubic boxes), and there will be an average of only 1 observation per bin. Clearly, such a histogram is useless because of its huge variance.

So sample size by itself means nothing. What really matters is not the number of observations, but the density of the observations in the feature space. This density collapses as more dimensions are added for a given sample size, and therefore as more parameters added to the model. Conversely, if one wants to maintain the same density (and therefore maintain the accuracy of the model) when more dimensions are added, then the sample size should increase enormously (usually exponentially with the number of dimensions). This is known as the **'curse of dimensionality'**.

6.5.7 Model selection

In the family of models that is being considered, how is the 'best' model going to be identified? First, we will never be certain that we have identified the best model in the family, because of the random nature of the sample. But it is possible (and necessary) to identify models that are probably fairly good. This can be done in two ways:

- The error level measured on the design set is always lower than the error level observed on new data (an extreme case is that of the polynom of degree n that goes through every point of the design set, the 'in-sample' error level being then 0). It is said to be 'optimistic'. In some cases, additional assumptions about the mechanism that generated the data allow this optimism to be quantified analytically. An estimation of the true error level can then be calculated. This is for example the case in Linear Regression. When the assumptions of the Linear Model are met:
  - The Coefficient of determination R^2 is a measure of the performance of the model on the design set.
  - The 'Adjusted R^2' is a (calculated) estimation of the performance of the model on new data.

  The analyst then builds several models, and retains the model with the lowest predicted error level.

- If the direct estimation of the error level is not feasible, one can apply various computer-intensive simulation techniques like:
  - Cross-validation, that repetitively puts aside some of the data to simulate 'new' data.
  - Bootstrap, that uses the empirical distribution function as an estimate of the true distribution function.

Bibliography

[1] A. Almansa and T. Lindeberg, *Fingerprint enhancement by shape adaptation of scale-space operators with automatic scale-selection*, 2000

[2] G. Aubert and P. Kornprobst, *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations (second edition)*, volume 147 of *Applied Mathematical Sciences*, Springer-Verlag, 2006

[3] D.H. Ballard, *Generalizing the Hough Transform to Detect Arbitrary Shapes*, Pattern Recognition **13**(2), pages 111–122, 1981

[4] Asker M. Bazen and Sabih H. Gerez, *Systematic Methods for the Computation of the Directional Fields and Singular Points of Fingerprints*, IEEE Transactions on Pattern Analysis and Machine Intelligence **24**(7), pages 905–919, 2002

[5] J. Bigun and G. H. Granlund, *Optimal Orientation Detection of Linear Symmetry*, In IEEE First International Conference on Computer Vision, pages 433–438, London, Great Britain, June 1987

[6] Christopher M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer, August 2006

[7] R. Cappelli, D. Maio, A. Lumini, and D. Maltoni, *Fingerprint Image Reconstruction from Standard Templates*, IEEE Transactions on Pattern Analysis and Machine Intelligence **29**(9), pages 1489–1503, 2007

[8] Jiangang Cheng, Jie Tian, Hong Chen, Qun Ren, and Xin Yang, *Fingerprint Enhancement Using Oriented Diffusion Filter*, Audio- and Video-Based Biometric Person Authentication pages 1054–1054, 2003

[9] A. Chessel, F. Cao, and R. Fablet, *Interpolating orientation fields: an axiomatic approach*, In Proc. European Conf. Comp. Vision (ECCV'06), volume 3951 (LNCS), pages 241–254, Graz, Austria, May 2006

[10] S. Chikkerur, C. Wu, and V. Govindaraju, *A Systematic Approach for Feature Extraction in Fingerprint Images*, In International Conference on Biometric Authentication, 2004

[11] Sharat Chikkerur, Alexander N. Cartwright, and Venu Govindaraju, *Fingerprint enhancement using STFT analysis*, Pattern Recognition **40**(1), pages 198–211, 2007

[12] Roland T. Chin and Charles R. Dyer, *Model-based recognition in robot vision*, ACM Comput. Surv. **18**(1), pages 67–108, 1986

[13] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, *Active Shape Models-Their Training and Application*, Computer Vision and Image Understanding **61**(1), pages 38–59, January 1995

[14] Timothy F. Cootes, Gareth J. Edwards, and Christopher J. Taylor, *Active Appearance Models*, IEEE Transactions on Pattern Analysis and Machine Intelligence **23**(6), pages 681–685, 2001

[15] Maio D., Maltoni D., Cappelli R., Wayman J. L., and Jain A. K., *FVC2002: Second Fingerprint Verification Competition*, In ICPR 2002, volume 3, pages 811–814, 2002

[16] Richard O. Duda and Peter E. Hart, *Use of the Hough transformation to detect lines and curves in pictures*, Commun. ACM **15**(1), pages 11–15, 1972

[17] Gunnar Farnebäck, *Polynomial Expansion for Orientation and Motion Estimation*, PhD thesis, Linköping University, Sweden, SE-581 83 Linköping, Sweden, 2002 Dissertation No 790, ISBN 91-7373-475-6

[18] Jianjiang Feng and Anil K. Jain, *FM Model Based Fingerprint Reconstruction from Minutiae Template.*, In Massimo Tistarelli and Mark S. Nixon, editors, ICB, volume 5558 of *Lecture Notes in Computer Science*, pages 544–553. Springer, 2009

[19] Jianjiang Feng, Soweon Yoon, and Anil K. Jain, *Latent Fingerprint Matching: Fusion of Rolled and Plain Fingerprints.*, In Massimo Tistarelli and Mark S. Nixon, editors, ICB, volume 5558 of *Lecture Notes in Computer Science*, pages 695–704. Springer, 2009

[20] Julian Fierrez, Javier Ortega-Garcia, Torre, and Joaquin Gonzalez-Rodriguez, *Biosec baseline corpus: A multimodal biometric database*, Pattern Recognition **40**(4), pages 1389–1392, April 2007

[21] J. Fierrez-Aguilar, L. M. Muñoz-Serrano, F. Alonso-Fernandez, and J. Ortega-Garcia, *On the effects of image quality degradation on minutiae- and ridge-based automatic fingerprint recognition*, In Proc. IEEE Intl. Carnahan Conf. on Security Technology, ICCST, pages 79–82, October 2005

[22] M. A. Fischler and R. A. Elschlager, *The Representation and Matching of Pictorial Structures*, Computers, IEEE Transactions on **C-22**(1), pages 67–92, 1973

[23] Martin A. Fischler and Robert C. Bolles, *Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography*, Communications of the ACM **24**(6), pages 381–395, 1981

[24] R. Fisker, *Making Deformable Template Models Operational*, PhD thesis, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, 2000

[25] Ralph M. Ford and Robin N. Strickland, *Nonlinear models for representation, compression, and visualizationof fluid flow images and velocimetry data*, In Visualization and Machine Vision, 1994. Proceedings., IEEE Workshop on, pages 1–12, Seattle, WA, USA, 1994. IEEE Computer Society

[26] H. Fronthaler, K. Kollreider, and J. Bigun, *Local Features for Enhancement and Minutiae Extraction in Fingerprints*, Image Processing, IEEE Transactions on **17**(3), pages 354–363, 2008

[27] Sir Galton, Francis, *Finger prints*, Macmillan, 1892

[28] Phiilips George, *Interpolation and Approximation by Polynomials*, Canadian Mathematical Society, 2003

[29] Cardillo Giuseppe, *uROCcomp: compare two unpaired ROC curves*, 2009 Mathworks Matlab-Central http://www.mathworks.com/matlabcentral/fileexchange/23020

[30] Colin Goodall, *Procrustes Methods in the Statistical Analysis of Shape*, Journal of the Royal Statistical Society. Series B (Methodological) **53**(2), pages 285–339, 1991

[31] Goesta H. Granlund, *In search of a general picture processing operator*, In Computer Graphics and Image Processing, pages 155–173, 1978

[32] Jinwei Gu and J. Zhou, *A novel model for orientation field of fingerprints*, In CVPR03, pages II: 493–498, 2003

[33] Isabelle Guyon, John Makhoul, Richard Schwartz, and Vladimir Vapnik, *What Size Test Set Gives Good Error Rate Estimates?*, IEEE Transactions on Pattern Analysis and Machine Intelligence **20**(1), pages 52–64, 1998

[34] James .A. Hanley and Barbara J. McNeil, *A method of comparing the area under two ROC curves derived from the same cases*, Radiology (147), pages 839–843, 1983

[35] Janne Heikkilä, *Robust Regression*, Graduate course on Advanced statistical signal processing.

[36] E.R. Henry, *Classification and Uses of Finger Prints*, Routledge, 1900

[37] Rhodes Henry, *Alphonse Bertillon - father of scientific detection*, Harrap, 1956

[38] Lin Hong, Yifei Wan, and Anil Jain, *Fingerprint Image Enhancement: Algorithm and Performance Evaluation*, IEEE Trans. Pattern Anal. Mach. Intell. **20**(8), pages 777–789, 1998

[39] Thomas Hotz, *Modelling and Analysing Orientation Field of Fingerprints*, phd thesis, 2007

[40] Peter Huber, *Robust Statistics*, John Wiley and Sons, 1981

[41] S. Huckemann, T. Hotz, and A. Munk, *Global Models for the Orientation Field of Fingerprints: An Approach Based on Quadratic Differentials*, PAMI **30**(9), pages 1507–1519, 2008

[42] Koichi Ito, Takafumi Aoki, and Tatsuo Higuchi, *Fingerprint Restoration Using Digital Reaction-Diffusion System and Its Evaluation (Special Section on Digital Signal Processing)*, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences **86**(8), pages 1916–1924, 2003

[43] Anil K. Jain, Yi Chen, and Meltem Demirkus, *Pores and Ridges: High-Resolution Fingerprint Matching Using Level 3 Features*, IEEE Transactions on Pattern Analysis and Machine Intelligence **29**(1), pages 15–27, 2007

[44] Anil K. Jain and David Maltoni, *Handbook of Fingerprint Recognition*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2003

[45] Anil K. Jain, Sharath Pankanti, Salil Prabhakar, Lin Hong, and Arun Ross, *Biometrics: A Grand Challenge*, icpr **02**, pages 935–942, 2004

[46] Anil K. Jain, Yu Zhong, and Marie-Pierre Dubuisson-Jolly, *Deformable template models: a review*, Signal Process. **71**(2), pages 109–129, 1998

[47] Bolle Ruud Jain Anil K. and Pankanti Sharath, *Biometrics, Personal Identification in Networked Society: Personal Identification in Networked Society*, Kluwer Academic Publishers, Norwell, MA, USA, 1998

[48] Michael Kass and Andrew Witkin, *Analyzing oriented patterns*, Comput. Vision Graph. Image Process. **37**(3), pages 362–385, 1987

[49] Michael Kass, Andrew Witkin, and Demetri Terzopoulos, *Snakes: Active contour models*, International Journal of Computer Vision **V1**(4), pages 321–331, January 1988

[50] Masahiro Kawagoe and Akio Tojo, *Fingerprint pattern classification*, Pattern Recogn. **17**(3), pages 295–303, 1984

[51] Carl T. Kelley, *Iterative Methods for Optimization*, Number 18 in Frontiers in Applied Mathematics. SIAM, Philadelphia, 1999

[52] H. Knutsson, *Representing Local Structure Using Tensors*, In The 6th Scandinavian Conference on Image Analysis, pages 244–251, Oulu, Finland, June 1989 Report LiTH–ISY–I–1019, Computer Vision Laboratory, Linköping University, Sweden, 1989

[53] Erwin Kreyszig, *Advanced Engineering Mathematics*, Wiley, 6th edition, 1988

[54] M. Kucken and A. Newell, *Fingerprint formation*, Journal of Theoretical Biology **235**(1), pages 71–83, July 2005

[55] Kok F. Lai and Roland T. Chin, *Deformable Contours: Modeling and Extraction*, IEEE Transactions on Pattern Analysis and Machine Intelligence **17**(11), pages 1084–1090, 1995

[56] C.H. Lee, S.H. Lee, and J.H. Kim, *A Study of Touchless Fingerprint Recognition System*, pages 358–365, 2006

[57] J. Li, W.Y. Yau, and H. Wang, *Constrained nonlinear models of fingerprint orientations with prediction*, PR **39**(1), pages 102–114, January 2006

[58] Jun Li, Wei-Yun Yau, Jian-Gang Wang, and Wee Ser, *Stability Analysis of Constrained Nonlinear Phase Portrait Models of Fingerprint Orientation Images*, In ICB, pages 493–502, 2007

[59] Tony Lindeberg, *Scale-Space Theory in Computer Vision*, Kluwer Academic Publishers, 1994

[60] Maio Dario Luminia Alessandra and Maltoni Davide, *Continuous versus exclusive classification for fingerprint retrieval*, Pattern Recognition Letters **18**(10), pages 1027–1034, October 1997

[61] D. Maio and D. Maltoni, *A structural approach to fingerprint classification*, Pattern Recognition, 1996., Proceedings of the 13th International Conference on **3**, pages 578–585 vol.3, 1996

[62] Dario Maio, Davide Maltoni, Raffaele Cappelli, James L. Wayman, and Anil K. Jain, *FVC2004: Third Fingerprint Verification Competition.*, In David Zhang and Anil K. Jain, editors, ICBA, volume 3072 of *Lecture Notes in Computer Science*, pages 1–7. Springer, 2004

[63] Li Q. Mardia K. V. and Hainsworth, *On the Penrose hypothesis on fingerprint patterns*, IMA Journal of Mathematics Applied in Medicine & Biology (9), pages 289–294, 1992

[64] R. Mester and M. Muhlich, *Improving Motion and Orientation Estimation Using an Equilibrated Total Least Squares Approach*, In ICIP01, pages II: 929–932, 2001

[65] Rudolf Mester, *Orientation estimation: conventional techniques and a new approach*, In InProc. European Signal Processing Conference (EUSIPCO2000), Tampere, 2000

[66] Matthias Mühlich and Rudolf Mester, *A Fast Algorithm for Statistically Optimized Orientation Estimation*, In DAGM-Symposium, pages 238–245, 2005

[67] Jorge Nocedal and Stephen J. Wright, *Numerical Optimization*, Springer, second edition, 2006

[68] K. Nordberg, H. Knutsson, and G. Granlund, *On the Equivariance of the Orientation and the Tensor Field Representation*, In Proceedings of the 8th Scandinavian Conference on Image Analysis, pages 57–63, Tromso, May 1993. SCIA, NOBIM, Norwegian Society for Image Processing and Pattern Recognition ISBN 82-992872-0-0

[69] Federal Bureau of Investigation, *WSQ gray-scale fingerprint image compression specification*, 1993

[70] L. O'Gorman and J. V. Nickerson, *Matched filter design for fingerprint image enhancement*, In Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on, pages 916–919 vol.2, 1988

[71] A. Ouanane and A. Serir, *Fingerprint Compression by Ridgelet Transform*, In Signal Processing and Information Technology, 2008. ISSPIT 2008. IEEE International Symposium on, pages 46–51, 2008

[72] Sharath Pankanti, Salil Prabhakar, and Anil K. Jain, *On the Individuality of Fingerprints*, IEEE Trans. Pattern Anal. Mach. Intell. **24**(8), pages 1010–1025, 2002

[73] Roger Penrose, *On best approximate solution of linear matrix equations*, Proceedings of the Cambridge Philosophical Society **52**(52), pages 17–19, 1956

[74] P. Perona, *Orientation Diffusions*, IEEE Trans. Image Processing **7**, pages 457–467, 1999

[75] J. Preece, Y. Rogers, H. Sharp, D. Benyon, S. Holland, and T. Carey, *Human-Computer Interaction*, Addison-Wesley, 1994

[76] A. R. Rao and R. Jain, *Analyzing oriented textures through phase portraits*, Pattern Recognition, 1990. Proceedings., 10th International Conference on **i**, pages 336–340 vol.1, 1990

[77] A. R. Rao and R. C. Jain, *Computerized flow field analysis: oriented texture fields*, Pattern Analysis and Machine Intelligence, IEEE Transactions on **14**(7), pages 693–709, 1992

[78] A. R. Rao and B. G. Schunck, *Computing oriented texture fields*, Computer Vision and Pattern Recognition, 1989. Proceedings CVPR '89., IEEE Computer Society Conference on pages 61–68, 1989

[79] Peter J. Rousseeuw and Annick M. Leroy, *Robust regression and outlier detection*, 1987

[80] Jidnya Shah, *From Template to Image: Reconstructing Fingerprints from Minutiae Points*, IEEE Trans. Pattern Anal. Mach. Intell. **29**(4), pages 544–560, 2007 Member-Ross, Arun and Fellow-Jain, Anil K.

[81] B. G. Sherlock and D. M. Monro, *Optimized wavelets for fingerprint compression*, In ICASSP '96: Proceedings of the Acoustics, Speech, and Signal Processing, 1996. on Conference Proceedings., 1996 IEEE International Conference, pages 1447–1450, Washington, DC, USA, 1996. IEEE Computer Society

[82] B. G. Sherlock, D. M. Monro, and K. Millard, *Fingerprint enhancement by directional Fourier filtering*, Vision, Image and Signal Processing, IEE Proceedings - **141**(2), pages 87–94, 1994

[83] B.G. Sherlock and D.M. Monro, *A model for interpreting fingerprint topology*, PR **26**(7), pages 1047–1055, July 1993

[84] C. A. B. Smith, *Note on the forms of dermatoglyphic patterns*, Birth Defects: Original Article Series **15**, pages 43–52, 1978

[85] Milan Sonka, Vaclav Hlavac, and Roger Boyle, *Image Processing, Analysis, and Machine Vision*, Thomson-Engineering, 2007

[86] P. K. Suetin, *Orthogonal Polynomials in Two Variables*, Gordon and Breach Science Publishers, 1999

[87] AI Access Training and Consulting in Data Mining, *Probabilities, Statistics and Data Modeling*, AI Access

[88] Matthew Turk and Alex Pentland, *Eigenfaces for Recognition*, Journal of Cognitive Neuroscience **3**(1), pages 71–86, 1991

[89] P.R. Vizcaya and L.A. Gerhardt, *A Nonlinear Orientation Model for Global Description of Fingerprints*, PR **29**(7), pages 1221–1231, July 1996

[90] Y. Wang, J. Hu, and D. Phillips, *A Fingerprint Orientation Model Based on 2D Fourier Expansion (FOMFE) and Its Application to Singular-Point Detection and Fingerprint Indexing*, PAMI **29**(4), pages 573–585, April 2007

[91] C.I. Watson and C.L. Wilson, *Nist Special Database 4: Fingerprint Database*, National Institute of Standards and Technology, 1992

[92] Shen Wei, Chen Xia, and Jun Shen, *Robust detection of singular points for fingerprint recognition*, Signal Processing and Its Applications, 2003. Proceedings. Seventh International Symposium on **2**, pages 439–442 vol.2, 2003

[93] J. Weickert, B. Romeny, and M. Viergever, *Efficient and Reliable Schemes for Nonlinear Diffusion Filtering*, IEEE Transactions on Image Processing **7**(3), pages 398–410, March 1998

[94] Joachim Weickert, *Anisotropic Diffusion in Image Processing*, volume 1 of *ECMI Serie*, B.G. Teubner Stuttgart, 1998

[95] Joachim Weickert, *Coherence-Enhancing Diffusion Filtering*, Int. J. Comput. Vision **31**(2-3), pages 111–127, 1999

[96] B. Widrow, *The rubber-mask technique, I: Pattern measurement and analysis*, Pattern Recognition **5**(3), pages 175–176, September 1973

[97] B. Widrow, *The rubber-mask technique, II: Pattern Storage and Recognition*, Pattern Recognition **5**(3), pages 199–211, September 1973

[98] Jin Chu Wu and Charles L. Wilson, *Nonparametric analysis of fingerprint data on large data sets*, Pattern Recogn. **40**(9), pages 2574–2584, 2007

[99] G. Y. Yang, H. Z. Shu, C. Toumoulin, G. N. Han, and L. M. Luo, *Efficient Legendre moment computation for grey level images*, Pattern Recogn. **39**(1), pages 74–80, 2006

[100] Zhengyou Zhang, *Parameter Estimation Techniques: A Tutorial with Application to Conic Fitting*, Technical Report RR-2676

[101] Qijun Zhao, Lei Zhang, David Zhang, and Nan Luo, *Direct Pore Matching for Fingerprint Recognition*, In ICB '09: Proceedings of the Third International Conference on Advances in Biometrics, pages 597–606, Berlin, Heidelberg, 2009. Springer-Verlag

[102] J. Zhou and J. Gu, *A model-based method for the computation of fingerprints' orientation field*, IP **13**(6), pages 821–835, June 2004

[103] J. Zhou and J. Gu, *Modeling orientation fields of fingerprints with rational complex functions*, PR **37**(2), pages 389–391, February 2004