The study of similarity score calculation methods for minutia-based fingerprint matching algorithms

by

Anton J. de Kock

Thesis presented in partial fulfilment of the requirements for the degree of Master of Applied Mathematics in the Faculty of Science at Stellenbosch University

Supervisor: Dr. J. Coetzer

Co-supervisor: Mr. M. E. Mathekga

December 2016

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: December 2016 ....

Abstract

The study of similarity score calculation methods for minutia-based fingerprint matching algorithms

A.J. de Kock

Department of Mathematical Sciences, Division of Applied Mathematics,
Stellenbosch University,
Private Bag X1, Matieland 7602, South Africa.

Thesis: MSc

December 2016

This study aims to establish guidelines for calculating the similarity score between two minutia point representations of fingerprints for minutia-based fingerprint matching. Existing research does not provide clear guidelines on how to calculate the similarity score between two minutia point representations and the reported performance of most existing algorithms include those comparisons for which the point matching algorithm failed. This study therefore compares the performance of existing similarity score calculation methods *after* the erroneous comparisons from the point matching algorithm have been *removed*. It furthermore investigates in which way and to what extent these methods are affected by intra-class variations and inter-class similarities. The results indicate that *none* of the *existing* similarity score calculation methods is superior to *all* the others when implemented on the FVC2002 and FVC2004 fingerprint databases. This study also proposes an *improved local descriptor* for local similarity score calculation and investigates whether the *combination* of different types of similarity score calculation methods better addresses intra-class variations and inter-class similarities and therefore improves proficiency. The results indicate that similarity score calculation methods that address both global and local inter-class similarities, and are robust to intra-class variations, perform better across multiple databases. Even though this study concludes that the combination of different types of similarity score calculation methods generally improves proficiency, high levels of noise and nonlinear distortion still adversely affect performance. Future work should therefore focus on improving the stages *preceding* the similarity score calculation stage, i.e. minutia extraction and point matching.

Uittreksel

’n Study vergelyking waarde berekeings vir minutia gebaseerde vingerafdruk vergelyking

(“The study of matching score calculation methods for minutia-based fingerprint matching algorithms”)

A.J. de Kock

Departement Wiskundige Wetenskappe, Afdeling Toegepaste Wiskunde,
Universiteit Stellenbosch,
Privaatsak X1, Matieland 7602, Suid Afrika.

Tesis: MSc

Desember 2016

Hierdie studie poog om riglyne vir die berekening van die eendersheid-telling tussen twee minutia-puntvoorstellings van vingerafdrukke vir minutia-gebaseerde vingerafdrukpassing daar te stel. Bestaande navorsing verskaf nie duidelike riglyne vir hoe om die eendersheid-telling tussen twee minutia punt-voorstellings te bereken nie en die gerapporteerde prestasie vir die meeste bestaande algoritmes sluit daardie vergelykings waarvoor die puntpassings-algoritme misluk in. Hierdie studie vergelyk dus die prestasie van bestaande eendersheid-telling berekeningsmetodes *nadat* die foutiewe vergelykings van die puntpassingsalgoritme *verwyder* is. Dit ondersoek ook op watter manier en in watter mate hierdie metodes deur intra-klas variasies en inter-klas ooreenstemmings beïnvloed word. Die resultate dui daarop dat *geen* van die *bestaande* eendersheid-telling berekeningsmetodes better as *al* die ander vaar wanneer dit op die FVC2002 en FVC2004 vingerafdruk databasisse geïmplementeer word nie. Hierdie studie stel ook ’n *verbeterde lokale beskrywer* vir lokale eendersheid-telling berekening voor en ondersoek of die *kombinasie* van verskillende eendersheid-telling berekeingsmetodes intra-klas variasies and inter-klas ooreenstemmings beter aanspreek en dus die prestasie verhoog. Die resultate dui daarop dat eendersheid-telling berekeningsmetodes wat beide globale en lokale inter-klas ooreenstemmings aanspreek, en onsensitief ten opsigte van intra-klas variasies is, beter oor veelvuldige databasisse vaar. Nieteenstaande die feit dat hierdie studie die gevolgtrekking maak dat die kombinasie van verskillende tipes van eendersheid-telling berekeningsmetodes die prestasie in die

algemeen verhoog, word die prestasie steeds deur hoë ruisvlakke en nie-lineêre vervorming verswak. Toekomstige werk moet dus op die verbetering van die stadia wat die eerdersheid-telling berekeningstadium voorafgaan fokus, m.a.w. minutia-onttrekking en puntpassing.

Acknowledgements

I would like to express my sincere gratitude to the following persons and/or institutions for their role in the successful completion of this study:

- My supervisor at Stellenbosch University, Dr. Hanno Coetzer for his support, patience and guidance throughout the completion of this thesis.
- My supervisor at the Council for Scientific and Industrial Research (CSIR), Mr. Dick Mathekga for his insightful comments and continuous support.
- The CSIR for financing this study.
- My family for their support through writing this thesis and my life in general.

Contents

List of Figures

List of Tables

List of Abbreviations

AUC	Area Under the (ROC) Curve
DB	Database
EER	Equal Error Rate
FMR	False Match Rate
FMR_{zero}	The lowest FNMR for which the FMR is equal to zero.
FMR_{100}	The lowest FNMR for which the FMR is smaller than or equal to 1%.
FMR_{1000}	The lowest FNMR for which the FMR is smaller than or equal to 0.1%.
FMR_{10000}	The lowest FNMR for which the FMR is smaller than or equal to 0.001%.
FNMR	False Non-Match Rate
$FNMR_{zero}$	The lowest FMR for which the FNMR is equal to zero
FVC	Fingerprint Verification Competition
LGS_DTR	Local Greedy Similarity with Distortion Tolerant Relaxation
MCC	Minutia Cylinder Code
ROC	Receiver Operating Characteristic

Nomenclature

Constants

$w_{\mathrm{R}} =$ 0.3

$n_{\mathrm{rel}} =$ 3.0

$\Lambda =$ 90.0 pixels

$\omega =$ 1.2 radians

Variables

x	Minutia point column coordinate
y	Minutia point row coordinate
θ	Minutia point rotation angle
S	Matching score value
P	Penalization factor
M	Minutia point
d	Distance between two minutia points
β	Orientation difference
ϕ	Angle difference between the orientation of a reference minutia point and the distance vector to a neighbouring minutia point
δ	Angle difference between the orientation of a neighbouring minutia point and the distance vector to a reference minutia point
S_{local}	Local similarity score value
S_{pair}	Similarity score value between two matched minutia points
S_{nm}	Similarity score value between two neighbouring minutia points
T	Template minutia point fingerprint representation or template set
Q	Query minutia point fingerprint representation or query set
t	Minutia point type
η	Ridge count between minutia points
N_{nm}	Number of paired neighbouring minutia points
N_{nn}	Number of unpaired neighbouring minutia points
N_{m}	Number of paired minutia points

N_{T}	Number of minutia points in the template set
N_{Q}	Number of minutia points in the query set
N_{TO}	Number of minutia points in the template set that falls in the overlapping convex hull
N_{QO}	Number of minutia points in the query set that falls in the overlapping convex hull
γ	Distance between two paired minutia points
ψ	Orientation difference between two paired minutia points
υ	The curve's maximum value in the sigmoid normalization function
τ	The steepness of the curve in sigmoid normalization function
ε	The efficiency of a minutia point
R_{m}	Noise measure between two aligned minutia sets
S_{GS}	Global inter-class similarity measure
S_{LS}	Local inter-class similarity measure
K	The symmetric sigmoid function
Θ	The angle difference function
s_d	Standard deviation
μ	Mean

Vectors

$\mathbf{Z}$	Feature vector containing the features extracted between two aligned sets of minutia points
$\mathbf{C}$	Vector representing the minutia cylinder code
$\mathbf{F}$	Vector containing the descriptor features and therefore representing the descriptor
$\boldsymbol{\varrho}$	Vector of descriptor feature thresholds

Chapter 1

Introduction

1.1 Background

The ease with which the bar-coded green South African national I.D. book can be forged has led to many instances of identity theft which are costing South African businesses, especially within the retail sector, millions of Rands in lost revenue. The response from the South African government to the threat posed by identity theft to businesses within the country, and therefore the economy as a whole, has been to replace the bar-coded green I.D. book with a smart-card based I.D. The smart-card based I.D. has in addition to physical security features, like holograms, laser-engraved biographic details and a photo of the I.D. holder for tamper proofing, also a representation of the holder's fingerprint stored in electronic form on the card for automated identity verification through fingerprint recognition.

Automated identity verification using fingerprints has become accepted as a more reliable means of confirming identity due to the uniqueness of fingerprints (Jain *et al.*, 2002) and the maturity of the technology. This is performed by comparing the fingerprint of one of the fingers of a person claiming an identity to the fingerprint of a similar finger that is linked to said identity, e.g. the fingerprint stored on the smart-card based I.D. This comparison is performed using a fingerprint matching algorithm, which calculates a similarity score which is then compared to a similarity score threshold[1] in order to decide whether the fingerprint was captured from the same finger. For example, a person will be considered to be the true owner of an identity if the comparison of his/her fingerprint with that stored against the claimed identity results in a similarity score value of 75, given that the similarity score threshold is 70.

There are different approaches to comparing fingerprints, that is fingerprint matching algorithms, which can broadly be grouped into (1) *image-based* and

[1] The similarity score threshold is the minimum similarity score value that is expected from a comparison of fingerprints of the same finger, assuming that low similarity score values are assigned to fingerprints captured from different fingers.

(2) *feature-based* matching algorithms. Feature-based matching algorithms can be further sub-categorised into *minutia-based*, *non-minutia-based*, and *hybrid* matching algorithms, with minutia-based matching algorithms being the most popular, mainly because they are both efficient and robust (Wen *et al.*, 2013).

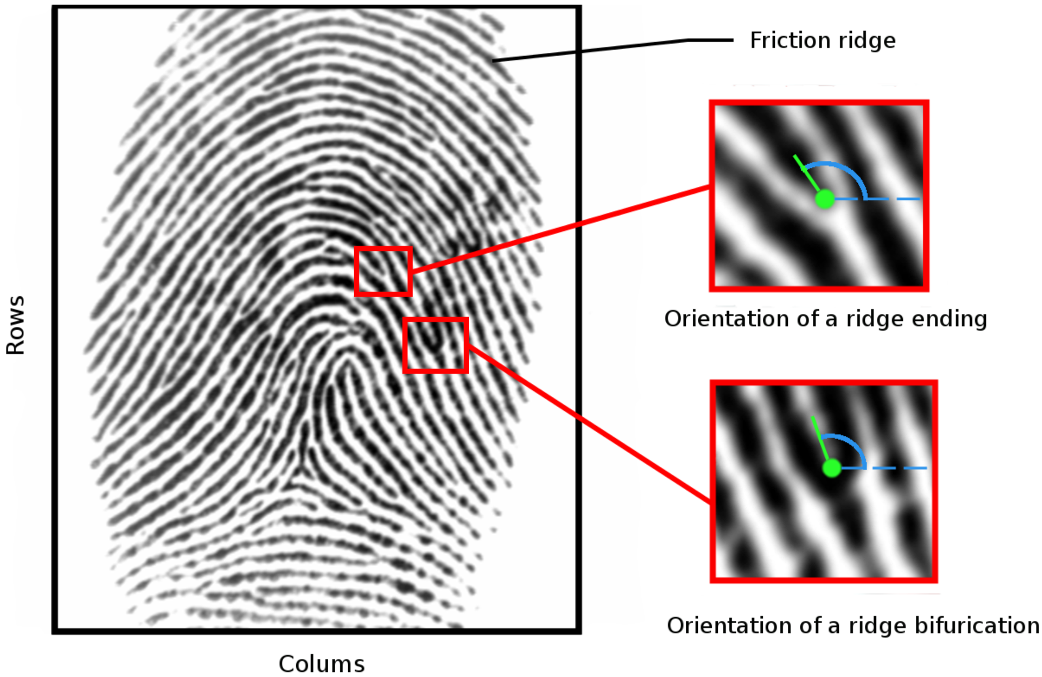


Figure 1.1: Examples of the location and orientation of a bifurcation and ridge ending minutia point on a fingerprint image.

Minutia-based matching algorithms compare fingerprints in an indirect way by comparing the locations and other characterising information of the minutia points on the fingerprints. Minutia points are the locations on a fingerprint where a ridge, which is the raised section of the skin on the inner surface of the hand (represented by dark lines on the fingerprint image in Figure 1.1), either ends or divides into two separate ridges. There are therefore two types of minutia points, a ridge ending and a ridge bifurcation, at the location where a ridge either ends or separates. Minutia points can therefore be characterised by their type and the orientation of the ridge(s) that they are linked to, as illustrated in Figure 1.1.

Two fingerprints are compared through a minutia-based matching algorithm (see Figure 1.2) by implementing the following three steps.

- Step 1 involves the *extraction* of minutia points from the fingerprints that are being compared.

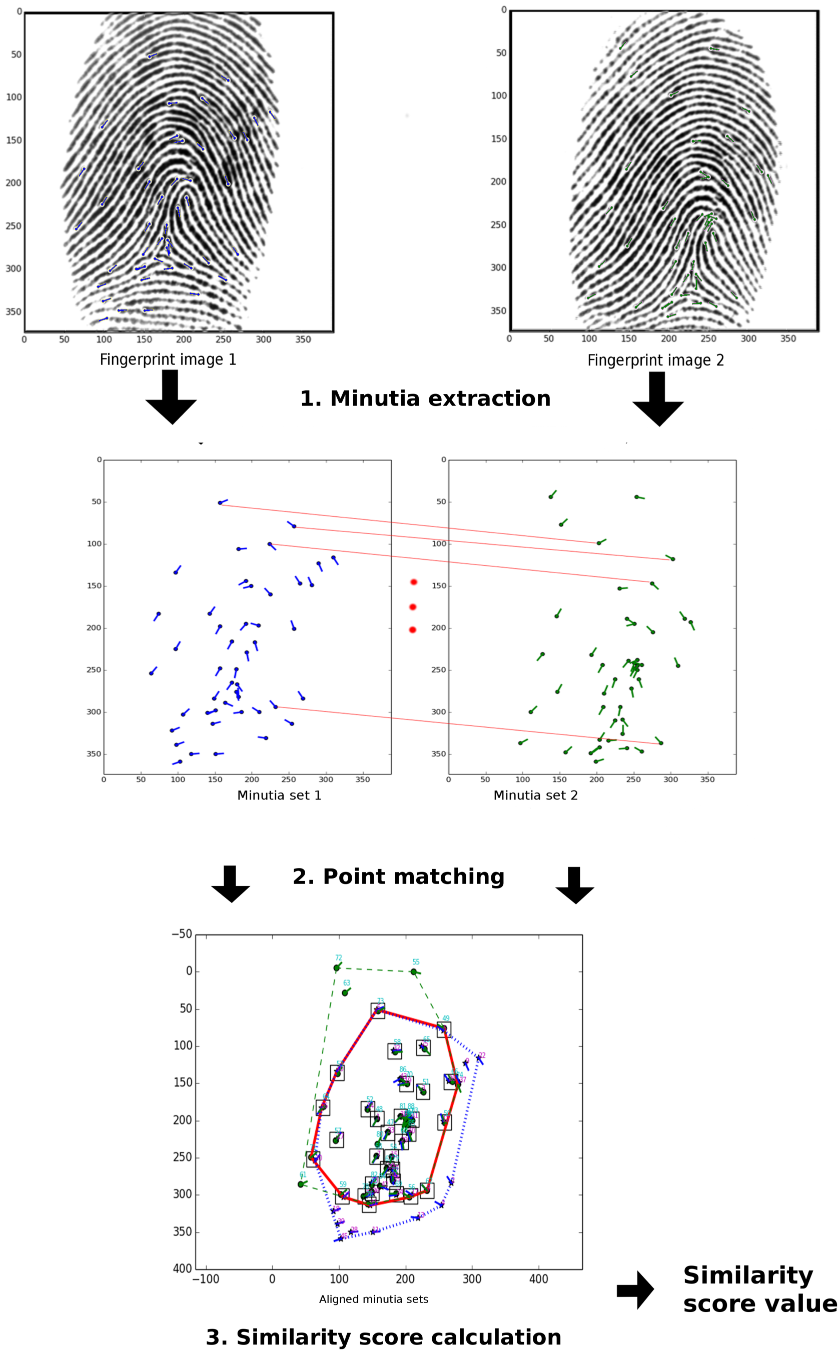


Figure 1.2: The three steps in minutia-based fingerprint matching: (1) minutia extraction, (2) point matching, and (3) similarity score calculation.

- Step 2 involves the *mapping* of the minutia points from one finger that constitute a similar pattern than those minutia points on another finger; a process that is known as minutia point matching (or simply point matching), the outcome of which is a list of mapped or paired minutia points.

- Step 3 involves the calculation of a *similarity score* based on the paired minutia points.

The term minutia-based matching algorithm will however be used to refer to *both* steps 2 and 3 listed above for the remainder of this thesis, as is the case in most of the current literature.

1.2 Problem statement

The goal of a matching algorithm is to calculate a similarity score that can be used to accurately tell impostor and genuine comparisons apart by assigning high similarity score values to genuine comparisons and low values to impostor comparisons. Genuine and impostor comparisons constitute comparisons of fingerprints belonging to the same and different fingers, respectively.

An obvious similarity score, which is mostly used by fingerprint experts, is the *number* of minutia points that are matched or paired. This serves as a good similarity score whenever the minutia points are matched or paired by a fingerprint expert using not only their relative position, but also the fingerprint images to guide the matching process. However, in the absence of fingerprint images to guide the process, as happens to be the case for minutia-based matching algorithms (the subject of the present work), minutia points that form *similar* patterns on fingerprints of *different* fingers may be erroneously paired, so that the number of paired minutia points (for this case) is comparable to the number of minutia points that form similar patterns on fingerprints of the *same finger*. This is due to *intra-class variations* and *inter-class similarities*, which constitute the differences and similarities between fingerprints, or minutia points from fingerprints, of the same finger and different fingers, respectively.

Intra-class variations that are typically encountered when comparing minutia points extracted from fingerprints of the *same* finger are shown in Figure 1.3 where four plots, each of minutia points extracted from fingerprints of the same finger are overlaid in such a manner that matching minutia points (inside square boxes) are aligned. When feature extraction errors are ignored, the different types of intra-class variations as illustrated in the plots of Figure 1.3 are mainly due to the fact that a finger is generally impressed *differently* on a fingerprint scanner each time a fingerprint is captured. In addition to this, the skin condition varies. A finger is generally not placed at exactly the same location, with the same orientation, and impressed with the same pressure on a fingerprint scanner each time a fingerprint is captured. The amount of pressure with which a finger is impressed on a fingerprint scanner has an effect

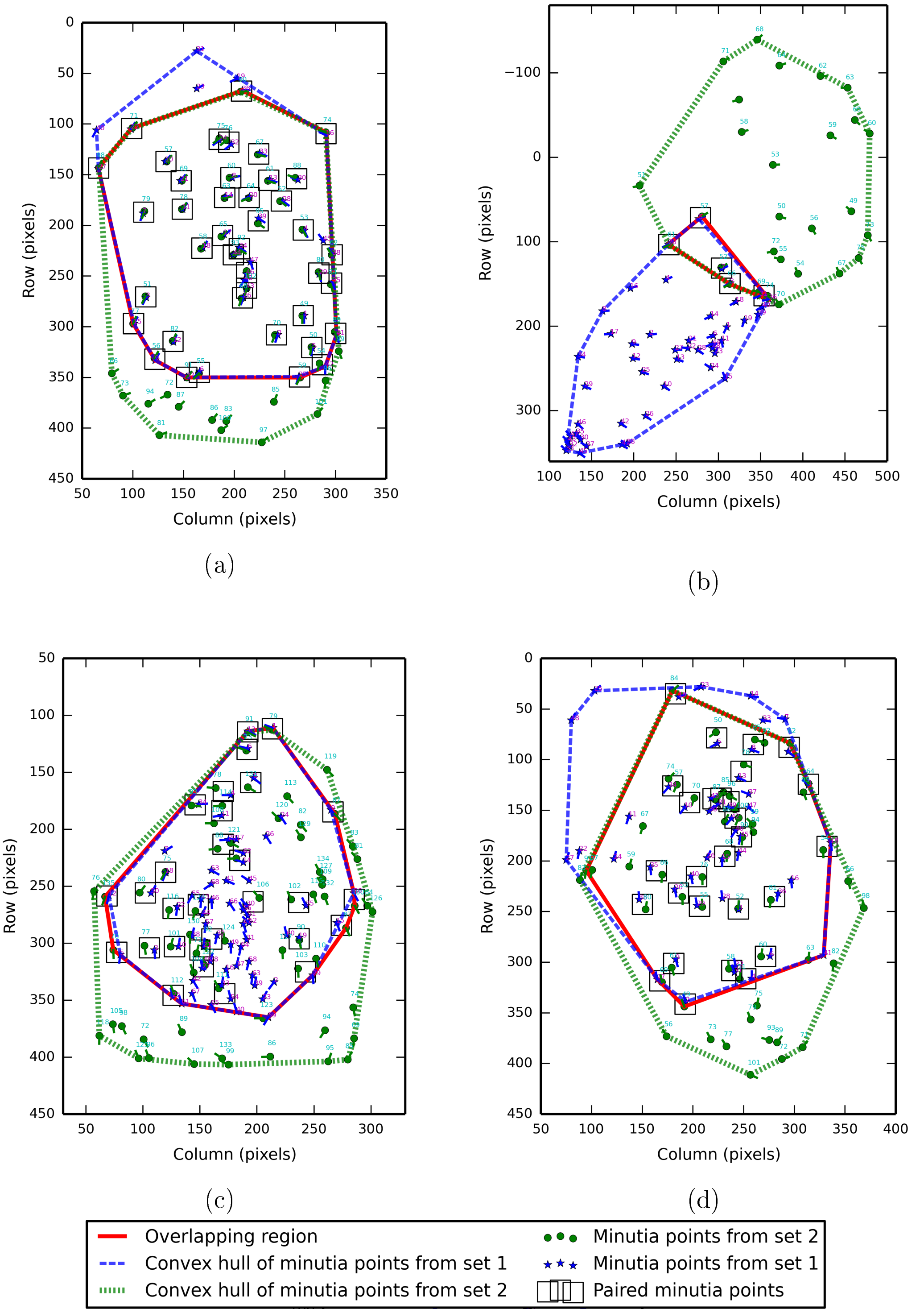


Figure 1.3: Different types of intra-class variations: (a) A scenario with ideal genuine comparisons with little variation between the two sets being compared. (b) A scenario where the variation is within the captured region, which leads to a small common region. (c) A scenario where the one set (indicated in blue) has many false minutia points which causes a large difference in the number of minutia points in the overlap. (d) A scenario where the one set (indicated in blue) has high levels of distortion in the top left corner which causes correctly paired minutia points to have a large variation in location.

on the thickness of the fingerprint ridges, how the skin of the finger stretches, and the amount of skin that makes contact with the scanning surface.

- A fingerprint image captured in such as way that the pressure with which the finger is impressed on a fingerprint scanner is relatively high, will be characterised by thick ridges, while the opposite is also true. Thick ridges may touch one another, therefore creating ridge bifurcations on a fingerprint image that do not exist on the physical finger.

- When fingers are impressed on a fingerprint scanner at different pressures the ridges will not align perfectly. This implies that the location of corresponding minutia points may differ.

- There is generally more skin detail on a fingerprint when the pressure with which a finger is impressed on a fingerprint scanner is high, while the opposite is also true.

Skin details on fingerprints that are captured by placing a finger at different locations and with different orientations on a fingerprint scanner (coupled with the factors listed above) will generally also be different, resulting in some fingerprints having details that are not present in others. Lastly, the skin condition, e.g. moisture levels, may be significantly different each time that a fingerprint is captured which has an effect on the clarity of the ridges. For example, a fingerprint captured from a dry finger will be different from that captured from a wet finger; a dry finger generally has discontinuous ridges, which means that the fingerprint may have ridge endings that are not present on the actual finger.

Examples of *inter-class similarities* are shown in Figure 1.4 where two plots, each showing minutia points extracted from fingerprints of *different* fingers, are overlaid in such a manner that matching points extracted from the fingerprints (inside the square boxes) are aligned. Inter-class similarities such as those depicted in Figure 1.4 (a) occur between fingerprints with the same ridge pattern – there are five different ridge patterns in total. Examples of fingerprints showing each of these five ridge patterns (with labels) are shown in Figure 1.5. Furthermore, in this work a distinction is made between global inter-class similarity and local inter-class similarity, where:

1. *global inter-class similarity* is defined as a form of inter-class similarity where many minutia points from fingerprints of different fingers are paired and a substantial overlap between the convex hulls of the minutia points from each fingerprint is observed such as those shown in Figure 1.4 (a); and

2. *local inter-class similarity* is defined as a form of inter-class similarity where the minutia points from fingerprints of different fingers are found to form a similar pattern in such a way that a small overlap between

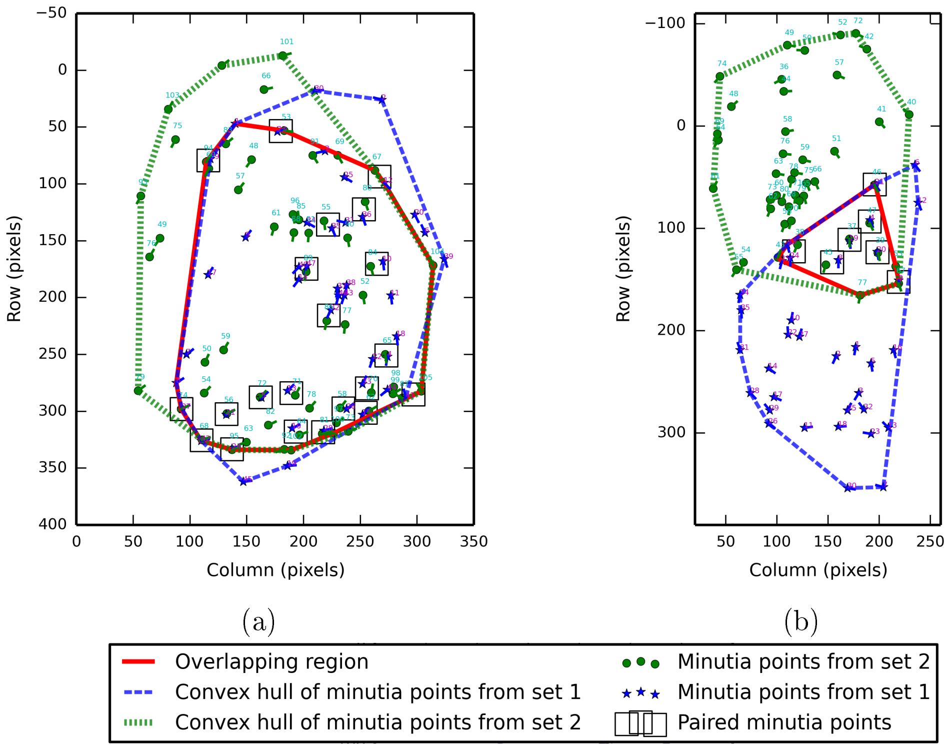


Figure 1.4: Illustration of global and local inter-class similarity between two minutia sets from fingerprints of different fingers. (a) Global inter-class similarity results from impostor comparisons with a globally similar minutia point structure, which causes many minutia points to fall within the tolerance box for impostor comparisons. (b) Local inter-class similarity results from impostor comparisons with a high similarity in the local minutia point structure.

> the convex hulls of the minutia points from each fingerprint is observed, with most of the minutia points in the overlap paired, as shown in Figure 1.4 (b).

Therefore, using the *number* of paired (or matched) minutia points as a similarity score will not meet the goal of being able to tell impostor and genuine comparisons apart, especially when intra-class variations and inter-class similarities are taken into account. However, the problem of calculating a similarity score or quantifying the similarity between representations of minutia points in fingerprints has received inadequate attention in the literature, despite the large number of minutia-based matching algorithms that have been published to date; to such an extent that only half a page is dedicated to this topic in the Handbook of Fingerprint Recognition by Maltoni *et al.* (2009). There is therefore a need for an approach that can be used to quantify the similarity between minutia points extracted from two fingerprints in such a manner that the goal of being able to tell impostor and genuine comparisons apart, are met,

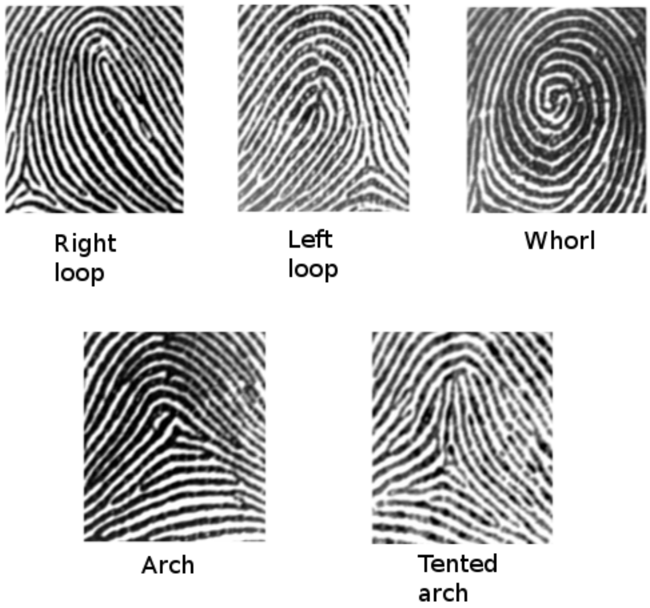


Figure 1.5: The five different fingerprint patterns.

while the above-mentioned inter-class similarities and intra-class variations are still taken into account.

1.3 Objectives

The goal of this research is to develop guidelines that will contribute towards the development of fingerprint matching algorithms which are not based on machine learning techniques. The emphasis is on the development of a similarity score calculation method that may be used to tell impostor and genuine comparisons apart.

The objectives of this research are therefore to:

- identify those methods employed in published minutia-based matching algorithms that specifically quantify the similarity between minutia point representations of fingerprints;
- compare the proficiency of different similarity score calculation methods as proposed in the above-mentioned (identified) publications in distinguishing impostor comparisons from genuine comparisons, taking both inter-class similarities and intra-class variations into account; and
- interpret the results of the above-mentioned comparisons, therefore identifying the strengths and weaknesses of each method, and also formulat-

ing guidelines that may be followed in calculating a similarity score that is able to proficiently tell impostor and genuine comparisons apart, again taking both inter-class similarities and intra-class variations into account.

The above-mentioned guidelines are then used to investigate two case studies, that aim to:

- determine whether *improvements* can be made to one or more of the existing similarity score calculation methods when implemented by some of the minutia-based matching algorithms being investigated; and
- investigate whether a *combined* similarity score as obtained by *fusing* several different similarity scores, each one using a different approach, will better meet the goal of proficiently telling impostor and genuine comparisons apart.

1.4 Delineations and limitations

Only minutia-based matching algorithms that calculate a similarity score between minutia point representations of fingerprints, using only the location and ridge orientation as a descriptor for a minutia point, are considered as part of the work presented in this thesis. This is due to the fact that minutia quality, which indicates amongst other things the reliability of the minutia point location and its ridge orientation, is usually added as a descriptor for the minutia point in question. However, there is no consensus on how minutia quality should be calculated and not all software programs for extracting minutia points include minutia quality as a feature within its descriptor.

Only non-learning-based approaches for quantifying the similarity between minutia point representations of fingerprints are considered throughout this study. This is mainly because of the fact that learning-based methods require training, which may not be feasible for many applications, and the performance of similarity scores calculated using these approaches will depend on the representativeness of the training data.

This investigation only focuses on minutia-based matching algorithms that the author is aware of and may not include all existing minutia-based matching algorithms. The guidelines reported in this thesis are therefore by no means comprehensive and only include those that are informed by the knowledge gained from evaluating the performances of the similarity scores calculated, using the matching algorithms that had been identified.

This study only focuses on the similarity between minutia points extracted from *live* scanned fingerprints. The guidelines presented here may therefore not generally apply to the similarity between *latent* fingerprints. Latent fingerprints are fingerprints that have been lifted from different surfaces at crime scenes; the comparison of latent fingerprints is a semi-automated process.

1.5 Assumptions

The work presented in this thesis is based upon the following assumptions:

- no two people have fingers with matching fingerprints, including identical twins (Tao and Veldhuis, 2013);
- the databases used in this study are sufficiently representative of typical levels of intra-class variations and inter-class similarities for live scanned fingerprints submitted to a typical fingerprint recognition application; and
- there will always be at least *three* paired minutia points for all genuine comparisons.

1.6 Summary of results

In the process of developing guidelines for calculating a similarity score for a matching method, we considered the different variations and similarities that may occur in genuine and impostor comparisons. Additionally, we compared different existing similarity score calculation methods in order to establish how the best methods are constructed and proposed *improvements* to some of these methods. We also *removed* the comparisons for which the point matching algorithm completely failed in order to determine whether existing error rates are a result of said point matching algorithm or the similarity score calculation method or both. We also analysed the *problematic* comparisons for existing similarity score calculation methods in order to identify whether any of them are sensitive to specific intra-class variations or inter-class similarities. Finally, we *combined* different similarity score calculation methods in order to establish whether this may improve the performance of the individual methods. A summary of the principal results from the above-mentioned investigation is presented below.

- Not one of the existing similarity score calculation methods being compared was shown to be superior to *all* the others when implemented on the databases considered, since each method was sensitive to at least one type of intra-class variation or affected by one type of inter-class similarity.
- However, the implementation of similarity scores that combine the *local* similarity with either the *structural* similarity or a *percentage of paired minutia points* in order to address global inter-class similarity, and which also use a *penalization factor* in order to address local inter-class similarity, lead to a higher average accuracy across *all* the databases.

- The assignment and calculation of the descriptor similarity between individual neighbouring minutia points, while considering the maximum number of paired neighbouring minutia points, lead to a *significantly* higher accuracy when implemented on the FVC2004 fingerprint databases than was the case for the *MCC-based local similarity*. This improvement is however not significant when implemented on the FVC2002 fingerprint databases, but a better average accuracy is still reported when compared to the MCC-based descriptor.

- The combination of the different types of similarity score calculation methods through a simple rule-based fusion method *significantly* increased the accuracy when implemented on the FVC2004 fingerprint databases. The improvement was however not *significant* for the FVC2002 fingerprint databases. The fusion-based method however preformed *significantly* better when implemented on the FVC2002 fingerprint databases when compared to the *best* existing similarity score calculation method as proposed by Cappelli *et al.* (2010*b*). A *significant* improvement in proficiency was however not observed for the FVC2004 fingerprint databases.

1.7 Significance

The work presented in this thesis constitutes one of the very few works that directly focuses on the calculation of the similarity score for the comparison of minutia point representations of fingerprints, as indicated by the fact that only half a page is dedicated to this topic in the Handbook of Fingerprint Recognition (Maltoni *et al.*, 2009). This work results in the following contributions:

- it *categorises* the different approaches to calculating similarity scores into *three* groups;

- it defines the concepts of global inter-class similarity and local inter-class similarity, in order to aid with the task of understanding how the performance of similarity score calculation methods are affected by inter-class similarities and intra-class variations;

- it explains why certain approaches to quantifying the similarity between minutia point representations of fingerprints perform better than others; and

- it proposes an improved local descriptor for calculating the similarity between two minutia points based on their local neighbourhoods in scenarios where high noise levels are present.

1.8 Brief thesis overview

The remainder of this thesis consists of six chapters with Chapters 2, 3, and 4 focusing on the performance comparison of existing similarity score calculation methods. Chapter 2 provides an overview of existing similarity score calculation methods. Chapter 3 explains the methodology followed, while Chapter 4 presents the results. Chapter 5 focuses on *improving* existing local similarity score calculation methods and Chapter 6 verifies whether the *combination* of the three different similarity score calculation methods identified in Chapter 2 improves the performance of the individual methods. Chapter 7 presents some final conclusions and discusses possible future work.

Chapter 2

Literature review

2.1 Introduction

The first proposed similarity scores simply constituted the *number* of paired minutia points between two fingerprints. The Rule of Thumb by Locard (Neumann *et al.*, 2012) suggested that 12 paired minutia points between two fingerprints, without any differences, establish identity beyond doubt. In 2009 the International Association for Identification however stated that “no valid basis exists for requiring that a predetermined minimum number of friction ridge characteristics must be present in two impressions in order to establish identity” (Neumann *et al.*, 2012).

This chapter reviews methods utilized to calculate similarity scores and is partitioned into two sections. The first section focuses on existing similarity score calculation methods and discusses how these are affected by intra-class variations and inter-class similarities. The second section demonstrates the lack of existing research that focuses on the calculation of the similarity score and discusses how this affects automated minutia-based matching. This chapter therefore provides an overview on automated minutia-based similarity score calculation methods and also highlights the problems associated with these methods.

2.2 Similarity score calculation methods

Although different similarity score calculation methods have been proposed, no study was found that *compares* existing methods or *categorises* similar methods. This literature review therefore provides an overview of existing similarity score calculation methods as depicted in Figure 2.1. The similarity score calculation methods are *categorised* based on the features that they incorporate. The first group of methods is limited to the *percentage of paired minutia points* in relation to either the maximum number of minutia points within the overlapping region that can be paired or the total number of minutia points overall that can be paired. The second group considers the similarity between paired minutia points and the similarity between different minutia pairs based on the spatial and orientation relations of the neighbouring minutia points or neighbouring minutia point pairs.

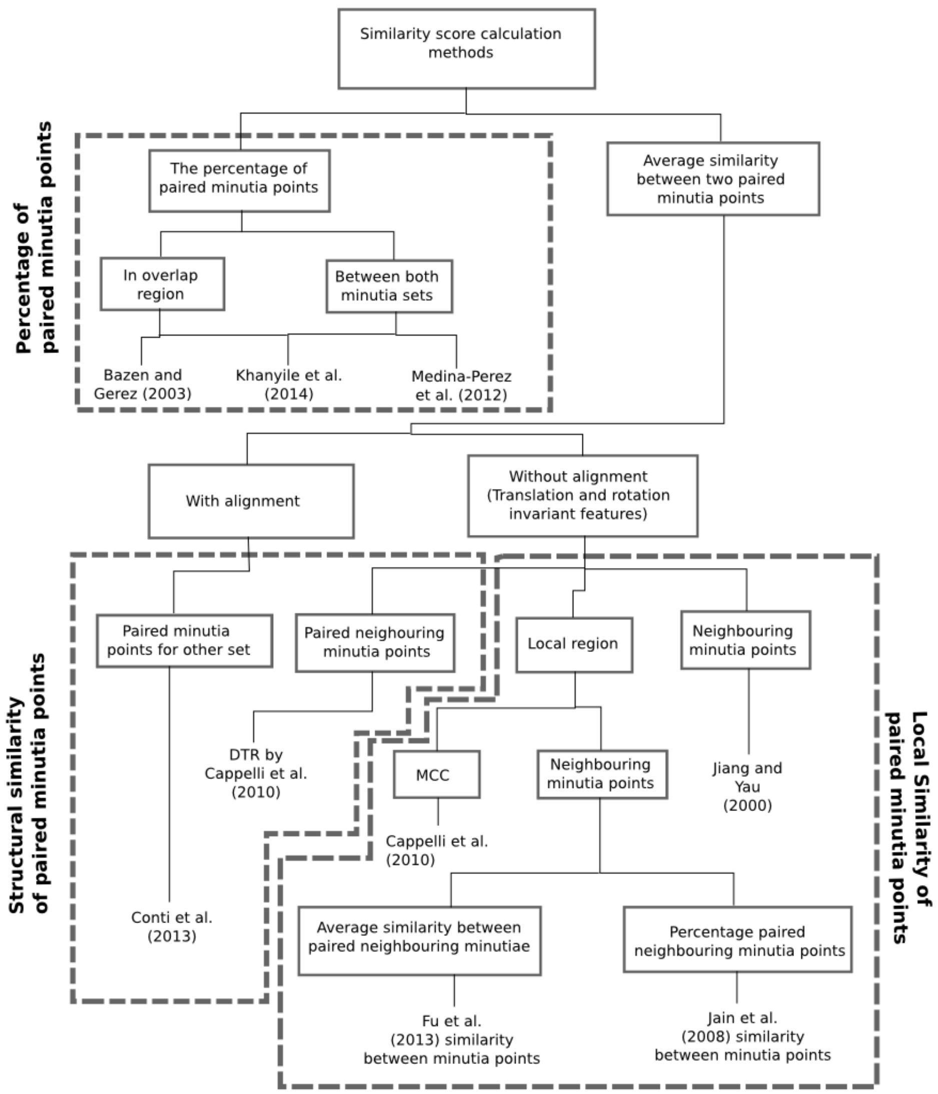


Figure 2.1: Overview of existing similarity score calculation methods.

In this section these different similarity score calculation methods are grouped into three categories. Even though many different ways for categorising these methods are possible (see Figure 2.1), it is clear that clusters of these methods capture similar features and are therefore robust/sensitive to the same type of intra-class variations and inter-class similarities. To simplify the discussion, this study categorises similarity scores into the following three groups based on how they are calculated, as well as their sensitivity and robustness to intra-class variations and inter-class similarities: (1) *percentage of paired minutia points*, (2) *local similarity score calculation methods*, and (3) *structural similarity score calculation methods*. The remainder of this section elaborates on existing methods belonging to these three groups and outlines their strengths and weaknesses.

2.2.1 Percentage of paired minutia points

The similarity score value for methods belonging to this group is calculated as a percentage of the paired number of minutia points (with some variations). Jea and Govindaraju (2005), as well as Medina-Pérez *et al.* (2012), calculate the similarity between minutia points from two fingerprints as a percentage of the paired minutia points using the following formula,

$$S = \frac{2.0 N_{\mathrm{m}}}{N_{\mathrm{T}} + N_{\mathrm{Q}}}, \tag{2.1}$$

where N_{m} denotes the number of paired minutia points, while N_{T} and N_{Q} denote the number of minutia points in sets T and Q respectively. The similarity score value calculated through this method very effectively distinguishes between genuine and impostor comparisons in scenarios with low levels of intra-class variations, but penalizes genuine comparisons with partial overlap and/or noise. Furthermore, it fails to adequately capture the dissimilarity of impostor comparisons with high levels of global inter-class similarity.

Both Bazen and Gerez (2003) and Fu *et al.* (2013) consider the number of paired minutia points relative to the total number of minutia points in the overlap as follows,

$$S = \frac{2.0 N_{\mathrm{m}}}{N_{\mathrm{TO}} + N_{\mathrm{QO}}}, \tag{2.2}$$

where N_{m} again denotes the number of paired minutia points, while N_{TO} and N_{QO} are the number of minutia points in the *overlapping region* of minutia sets T and Q respectively. This method more accurately captures the dissimilarity in cases of high levels of global inter-class similarity, but its performance is adversely affected by noise. The main problem associated with this similarity score calculation method lies in the fact that it generates high score values for impostor comparisons with high levels of local inter-class similarity. In these cases this similarity score calculation method fails to decrease the level of certainty reflected in the similarity score when the comparisons have a small overlap.

Khanyile *et al.* (2014) and Jain *et al.* (2008) address the above-mentioned problem by adding a *penalization factor*, P, to the similarity score in Equation 2.2. The penalization factor proposed by Khanyile *et al.* (2014) is based on the ratio of minutia points within the overlapping region for the two sets,

$$P = \left(\frac{N_{\mathrm{TO}}}{N_{\mathrm{T}}} + \frac{N_{\mathrm{QO}}}{N_{\mathrm{Q}}} \right), \tag{2.3}$$

while the penalization factor proposed by Jain *et al.* (2008) is based on the number of paired minutia points,

$$P = \left(\frac{N_{\mathrm{m}}}{N_{\mathrm{m}} + 8.0} \right). \tag{2.4}$$

Both Equations 2.3 and 2.4 decrease the similarity score associated with impostor comparisons with high levels of local inter-class similarity, since they are *multiplied* with the percentage of paired minutia points calculated through Equation 2.2. The penalization factor proposed by Khanyile *et al.* (2014) may however be adversely affected by two partial sets that completely overlap, while the factor proposed by Jain *et al.* (2008) may be adversely affected by significant global inter-class similarity when there are relatively many paired minutia points.

The similarity scores calculated using the methods in this group can be employed to accurately distinguish between genuine and impostor comparisons with relatively low levels of inter-class similarity and intra-class variation. Furthermore, the extra penalization factors proposed by Khanyile *et al.* (2014) and Jain *et al.* (2008) may be able to deal with partial overlap and local inter-class similarity to a certain degree. However, the similarity score values obtained through methods in this group are highly sensitive to noise, while a limited number of minutia points do not provide a sufficiently high resolution for these methods, and may complicate the process of distinguishing between genuine and impostor comparisons.

2.2.2 Local similarity score calculation methods

Local similarity within the context of this thesis is defined as a measure of the similarity between two minutia points from different fingers based on the configuration and features of *neighbouring* minutia points within a local region; where neighbouring minutia points are either the n closest minutia points or those minutia points falling within a fixed radius from the minutia point in question. The similarity is calculated by comparing rotation and translation invariant features of different minutia points as derived from spatial and orientation features of their neighbouring minutia points, which are known as *local descriptors*. The *local similarity score* is the *average* local similarity value associated with the paired minutia points.

The local similarity score has several advantages that result in local similarity score calculation methods being useful for minutia-based matching. Firstly, the local similarity measure is less affected by nonlinear distortion due to the

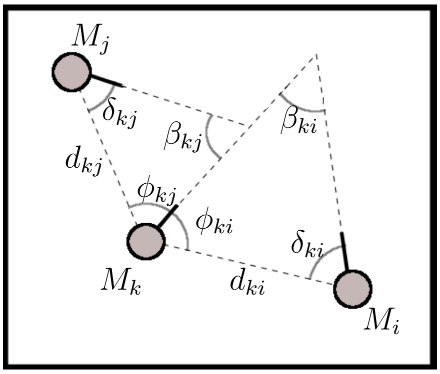


Figure 2.2: With M_j and M_i neighbouring minutia points of M_k, most local similarity calculation methods use the features depicted in this figure to create local descriptors and calculate the local similarity. The symbol d denotes the Euclidean distance while β, δ and ϕ are angle differences within the range $[0, \pi]$.

fact that only a local region is considered (Bringer and Despiegel, 2010), which is the main reason why this group was created in the first place. Furthermore, it can be calculated without prior alignment because of the translation and rotation invariance of the descriptors (Neumann *et al.*, 2012). The extent to which it is affected by noise and partial overlap however depends on the type of descriptor that is used, the comparison technique and the number of paired minutia points included in the calculation of the local similarity.

Many different types of local descriptors may be used to calculate the local similarity. However, only the following *three* descriptors, which are the most commonly used, are presented for the purposes of this review: (1) the *fixed length* descriptor, (2) the *fixed radius* descriptor, and (3) the *Minutia Cylinder Code (MCC)* descriptor.

2.2.2.1 Fixed length descriptor

The fixed length or n-closest neighbouring descriptor has a fixed number of features that are calculated using the characteristics of a fixed number of n-closest neighbouring minutia points. This descriptor can only be considered to calculate the local similarity as long as only a few neighbouring minutia points are considered and the minutia point density is medium to high. The features associated with the neighbouring minutia points are ordered, primarily based on the Euclidean distance from the reference minutia points. Each corresponding feature value in the two descriptors can be directly compared to calculate the local similarity.

Jiang and Yau (2000) include this local similarity in their similarity score calculation method by combining this similarity with the percentage of paired minutia points in the overlapping region. They only consider the two closest neighbouring minutia points in order to determine the local similarity between the two minutia points in question. They compute the local similarity (S_{local}) as follows,

$$S_{\text{local}}(M_{k_{\text{T}}}, M_{k'_{\text{Q}}}) = \begin{cases} \boldsymbol{\varrho} - \mid \mathbf{F}_k - \mathbf{F}_{k'} \mid, & \text{if } \mid \mathbf{F}_k - \mathbf{F}_{k'} \mid < \boldsymbol{\varrho} \\ 0, & \text{otherwise} \end{cases}, \tag{2.5}$$

where $\mathbf{F}_k$ is the local descriptor (feature vector) of reference minutia point $M_{k_{\text{T}}}$ with two neighbouring minutia points i and j in fingerprint T, and $\mathbf{F}_{k'}$ the local descriptor of reference minutia point $M_{k'_{\text{Q}}}$ with two neighbouring minutia points i' and j' in fingerprint Q, where

$$\mathbf{F}_k = [d_{ki}, d_{kj}, \beta_{ki}, \beta_{kj}, \delta_{ki}, \delta_{kj}, \eta_{ki}, \eta_{kj}, t_{ki}, t_{kj}] \tag{2.6}$$

and

$$\mathbf{F}_{k'} = [d_{k'i'}, d_{k'j'}, \beta_{k'i'}, \beta_{k'j'}, \delta_{k'i'}, \delta_{k'j'}, \eta_{k'i'}, \eta_{k'j'}, t_{k'i'}, t_{k'j'}] \tag{2.7}$$

with the features d_{ki}, d_{kj}, β_{ki}, β_{kj}, δ_{ki}, and δ_{kj} as depicted in Figure 2.2. The features t_{ki}, t_{kj}, $t_{k'i'}$ and $t_{k'j'}$ represent the different types of minutia point, while η_{ki}, η_{kj}, $\eta_{k'i'}$ and $\eta_{k'j'}$ constitute the ridge count between two different neighbouring minutia points and the reference minutia point in both descriptors. The ridge count is an image feature. Despite the fact that the type of minutia point is deemed unreliable by Cappelli *et al.* (2010*a*), these *eight* features (associated with t and η) are included to improve distinctiveness; the algorithm may however still be relatively proficient without them. The symbol $\boldsymbol{\varrho}$ denotes an empirical *threshold vector* for the above-mentioned features.

The fact that the features associated with these descriptors are first ordered, and then compared in said order, causes said approaches to be highly sensitive to noise and partial overlap (Liu and Mago, 2012; Feng, 2008). Furthermore, the limited number of neighbouring minutia points included in these descriptors may not be very distinctive in regions with a high minutia density. Therefore, the fixed length local descriptor should be avoided when high noise levels are present.

2.2.2.2 Fixed radius descriptor

The fixed radius local descriptor was created to address the noise sensitivity associated with the fixed length local descriptor. The features of a fixed radius local descriptor are based on the minutia points that lie within a fixed radius of a reference minutia point. The length of this descriptor is therefore variable. Fixed radius descriptors are compared by first pairing neighbouring minutia points that lie within the fixed radius and then calculating the similarity between them using a number of different approaches, including the percentage of paired minutia points.

Jain *et al.* (2008) proposed using a local similarity between minutia sets by employing a fixed radius descriptor. They apply thresholds on the differences between d, β and ϕ of neighbouring minutia points (as depicted in Figure 2.2) in order to determine the number of *paired* neighbouring minutia points. The following formula is then used to calculate the local similarity,

$$S_{\text{local}}(M_{k_\text{T}}, M_{k'_\text{Q}}) = \left(\frac{N_{\text{nm}_k} + 1}{N_{\text{nm}_k} + N_{\text{nn}_k} + 3}\right)\left(\frac{N_{\text{nm}_{k'}} + 1}{N_{\text{nm}_{k'}} + N_{\text{nn}_{k'}} + 3}\right), \qquad (2.8)$$

where N_nm is the number of *paired* neighbouring minutia points and N_nn the number of *unpaired* neighbouring minutia points in the overlapping region between the regions *surrounding* minutia points M_{k_T} and $M_{k'_\text{Q}}$ associated with minutia sets T and Q respectively.

The local similarity calculated through a fixed radius local descriptor is more robust with respect noise than is the case for a fixed length local descriptor, but this robustness has a few adverse consequences. Firstly, the calculation of the local similarity is more complex (Cappelli *et al.*, 2010*a*). Secondly, the local similarity may be affected by noise or density variations within the region when the similarity is calculated as a percentage of paired neighbouring minutia points. Thirdly, the fixed radius regions may cause border errors when a neighbouring minutia point is included in one descriptor, while the corresponding neighbouring minutia point in another descriptor falls outside said radius (Feng, 2008).

The size of the fixed radius descriptor may have a significant impact on the accuracy of the system. This is due to the fact that too small descriptors are not significantly distinctive, while too large descriptors are adversely affected by nonlinear distortion and border errors. Fu and Feng (2015) illustrate this relation through a plot of the EER against the radius of the local descriptor and showed that the minimum EER for their local similarity is achieved at approximately a radius of 90 pixels. Most methods, such as those proposed by Jain and Feng (2011) and Cappelli *et al.* (2010*a*) utilise a radius in the 75 to 90 pixel range. However, these lengths are specific for fingerprint images captured at 500 dpi and needed to be adjusted when the fingerprint images are captured at a different resolution.

2.2.2.3 Minutia Cylinder Code descriptor

One of the few fingerprint matching algorithms that successfully employs the local similarity score as a *stand-alone score* was developed by Cappelli *et al.* (2010*a*). They developed a new type of descriptor, the Minutia Cylinder Code (MCC), which converts the local region within a fixed radius about each minutia point into a three-dimensional cylinder where the vertical axis represents the orientation, while the two horizontal axes represent the x and y coordinates, as illustrated in Figure 2.3. The cylinder is subdivided into cuboids, where each cuboid is assigned a value between 0 and 1 based on the extent to which a neighbouring minutia point falls within the orientation and spatial

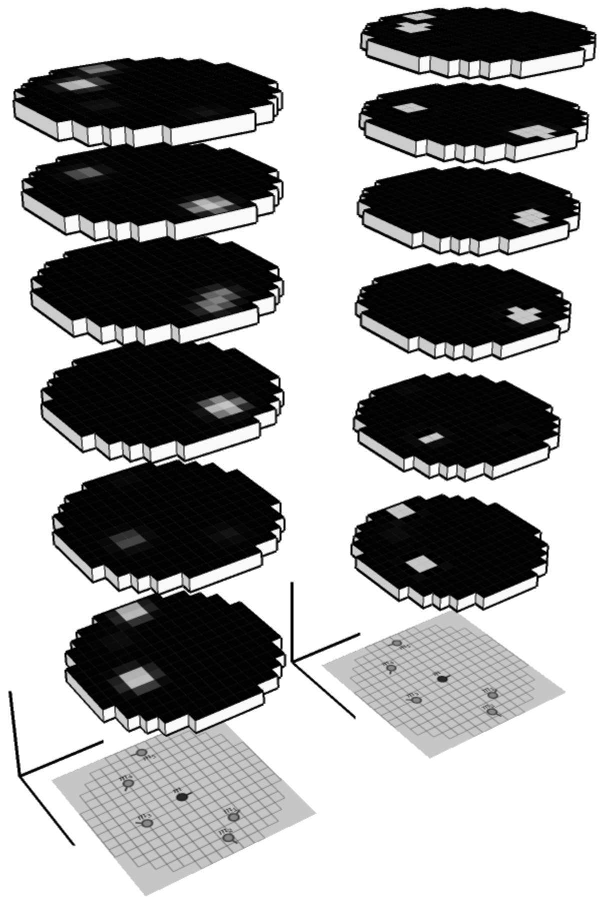


Figure 2.3: The local neighbourhood minutia point descriptor as created and illustrated by Cappelli *et al.* (2010*b*).

range of the cuboid in question. Said value is determined through normalization functions and statistical rules. The local similarity between two minutia point descriptors is calculated as follows,

$$S_{\text{local}}(M_{k_{\text{T}}}, M_{k'_{\text{Q}}}) = \begin{cases} 1 - \dfrac{\| \mathbf{C}_{k_{\text{T}}} - \mathbf{C}_{k'_{\text{Q}}} \|}{\| \mathbf{C}_{k_{\text{T}}} \| + \| \mathbf{C}_{k'_{\text{Q}}} \|}, & \text{if } \mathbf{C}_{k_{\text{T}}} \text{ and } \mathbf{C}_{k'_{\text{Q}}} \text{ are matchable} \\ 0, & \text{otherwise} \end{cases}, \quad (2.9)$$

where $\mathbf{C}_{k_{\text{T}}}$ and $\mathbf{C}_{k'_{\text{Q}}}$ are the cylinder codes for minutia points $M_{k_{\text{T}}}$ and $M_{k'_{\text{Q}}}$ which belongs to fingerprints T and Q respectively.

The local similarity value based on the MCC descriptor shares the general advantages associated with local similarity values based on other descriptors, but the MCC-based local similarity value can be calculated more efficiently since the descriptor has a fixed length. MCC descriptors may be more distinctive than other fixed length descriptors since they incorporate a larger region with more information. The MCC local descriptor deals well with border errors and accounts for partial overlap, since cuboids are identified as invalid when they fall outside the overlapping region and are only compared if valid in *both* cylinders. This method is however sensitive to noise as a result of the fact that the descriptor includes all the neighbouring minutia points within a fixed radius. Even so, a study conducted by Feng and Zhou (2011) showed that the MCC local descriptor is the best local descriptor for point matching and is only outperformed by hybrid approaches for fingerprint comparisons with a partial overlap.

2.2.3 Structural similarity score calculation methods

The similarity score calculation methods in this group measure the proficiency with which the paired minutia points are *aligned*. A measure that estimates the proficiency of the alignment is calculated for each minutia pair and the *average* of these measures constitutes the similarity score.

Conti *et al.* (2013) calculate the similarity score as the average of the 12 largest structural similarity values between the paired minutia points, where the structural similarity between a pair of minutia points after alignment is calculated as follows,

$$S_{\text{pair}}(M_{k_{\text{T}}}, M_{k'_{\text{Q}}}) = 0.75\left(1 - \frac{\gamma}{\gamma_{\text{T}}}\right) + 0.25\left(1 - \frac{\psi}{\psi_{\text{T}}}\right), \tag{2.10}$$

where $S_{\text{pair}}(M_{k_{\text{T}}}, M_{k'_{\text{Q}}})$ depicts the structural similarity between the paired minutia points, $M_{k_{\text{T}}}$ and $M_{k'_{\text{Q}}}$ represent minutia points from fingerprints T and Q respectively, and γ and ψ denote the Euclidean distance and orientation difference between the two minutia points respectively. The symbols γ_{T} and ψ_{T} are normalization constants. The similarity score calculated using this approach may be sensitive to partial overlap and nonlinear distortion. However, this similarity score should not be affected by noise when the matching minutia points are accurately paired.

Cappelli *et al.* (2010*a*) convert the *relaxation approach* proposed by Feng *et al.* (2006) in such a way that the structural similarity between paired minutia points can be calculated *without alignment*. For each set of paired minutia points, the structural similarity is calculated with respect to the best *other* paired minutia points. This approach is similar to the calculation of the local similarity measure, but occurs on a global level instead and involves only the *paired* minutia points and not the local neighbouring minutia points. Cappelli *et al.* (2010*b*) improve on this approach and combine the feature differences for each pair of minutia points into a value $\rho(k, i)$, where

$$\rho(k,i) = \prod_{s=1}^{3} \kappa(D_s, \upsilon_s^\rho, \tau_s^\rho) \tag{2.11}$$

$$D_1 = \frac{\mid d_{ki} - d_{k'i'} \mid}{d_{ki} + d_{k'i'}}, \tag{2.12}$$

$$D_2 = \mid d\Theta\left(\beta_{ki}, \beta_{k'i'}\right) \mid, \tag{2.13}$$

$$D_3 = \mid d\Theta\left(\phi_{ki}, \phi_{k'i'}\right) \mid, \tag{2.14}$$

$$\tag{2.15}$$

with υ_s^ρ and τ_s^ρ normalization constants and d_{ki}, β_{ki}, and ϕ_{ki} as defined in Figure 2.2, except that M_k and $M_{k'}$ represent the kth paired minutia points after alignment, while and M_i and $M_{i'}$ depict the ith paired minutia points and *not* the *neighbouring* minutia points. κ represents the symmetrical sigmoid function,

$$\kappa\left(x, \upsilon, \tau\right) = \frac{1}{1 + e^{-\tau(x-\upsilon)}}, \tag{2.16}$$

while the orientation difference in the range $[0, \pi]$ is defined as follows,

$$d\Theta\left(\theta_1, \theta_2\right) = \begin{cases} \theta_1 - \theta_2, & \text{if } -\pi \leq \theta_1 - \theta_2 < \pi \\ 2\pi + \theta_1 - \theta_2, & \text{if } \theta_1 - \theta_2 < -\pi \\ -2\pi + \theta_1 - \theta_2, & \text{if } \theta_1 - \theta_2 \geq \pi. \end{cases} \tag{2.17}$$

The final relaxed similarity measure for one minutia pair is calculated as follows,

$$\lambda_k^n = w_{\mathrm{R}}.\lambda_k^{n-1} + (1 - w_{\mathrm{R}}).\left(\sum_{i=1, i\neq k}^{n_{\mathrm{R}}} \rho\left(k,i\right).\lambda_i^{n-1}\right) / \left(n_{\mathrm{R}} - 1\right), \tag{2.18}$$

where $w_{\mathrm{R}} \in [0, 1]$ is a constant, λ_i^{n-1} initially denotes the local similarity measure for each pair (when $n = 0$), and then represents the relaxed similarity measure of the previous cycle (when $n > 0$). The index k represents the specific paired minutia points for which the structural similarity is being calculated, while the symbol λ denotes the ith of the *other* paired minutia points being considered and n_{R} the number of paired minutia points being compared. The paired minutia points are then ordered based on the efficiency, ε, where ε is a measure of the extent to which their local similarity also holds globally. The efficiency is calculated as follows,

$$\varepsilon = \frac{\lambda_k^{n_{\mathrm{rel}}}}{\lambda_k^0}, \tag{2.19}$$

where λ_k^0 represents the local similarity assigned to the kth pair of minutia points and $\lambda_k^{n_{\mathrm{rel}}}$ the relaxed global structural similarity of the kth paired minutia points after n_{rel} relaxation cycles. The relaxed structural similarity, which includes the local similarity, are *averaged* for a fixed number of pairs in order to obtain the *structural similarity score*.

The approach proposed by Cappelli *et al.* (2010*b*) is invariant with respect to translation and rotation, does not need alignment and may be robust with respect to noise to a certain extent. Additionally, the relevant distance feature also renders it more robust to nonlinear distortion than is the case for the approach proposed by Conti *et al.* (2013). The main problem associated with this approach is that it either estimates the number of paired minutia points or uses a fixed number. The overestimation of the number of paired minutia points may lead to sensitivity of the similarity score to partial overlap or noise, while an underestimation may fail to capture the dissimilarity of impostor comparisons. On the other hand, since it estimates the number of paired minutia points, it will assign an equal number of paired minutia points to impostor and genuine comparisons, which may better capture the dissimilarity of comparisons with high levels of global inter-class similarity.

2.2.4 Hybrid similarity score calculation methods and score fusion

Few existing methods combine two of the three types of similarity score calculation methods (as outlined in Sections 2.2.1 to 2.2.3) and the author is not aware of a method that combines *all three* methods. Score fusion constitutes the process of combining multiple features or matching score values into a single value. The purpose of score fusion is to determine the likelihood that a comparison is genuine, given different features or matching score values that may be robust to different types of intra-class variations or inter-class similarities. In the current literature, score fusion techniques are typically divided into *three* groups (Nanni *et al.*, 2014; He *et al.*, 2010): (1) *density-based* fusion, (2) *classifier-based* fusion, and (3) *transformed/combination-based* fusion.

Density-based, classifier-based and transformed/combination-based fusion techniques may accurately combine many features or matching score values, but these approaches have one drawback: they all require training data (Nanni *et al.*, 2012). Density-based fusion techniques require training data to estimate class-conditional density functions, classifier-based fusion techniques need training data to determine the optimal separation between the two classes (i.e. those associated with genuine and impostor comparisons) and most transformed/combination-based fusion techniques need training data to determine the weights associated with specific features or normalization parameters (Nandakumar *et al.*, 2008).

There do however exist transformed/combination-based fusion techniques such as the sum or multiplication rule, for which the incorporation of normalization techniques may improve system proficiency through fusion. Normalization constitutes the process by which a range of feature values are scaled to a new, more standardized range. Feature normalization may be achieved by dividing certain features with other features or by using existing normalization functions like the max-min normalization function, decimal scaling, the z-score, the median and the median deviation, the sigmoid function and tanh-estimators. A linear normalization function like the max-min function only changes the range of a feature, while nonlinear normalization functions like sigmoid functions or tanh-estimators may also change the feature's distribution (Jain *et al.*, 2005).

Certain existing methods combine one or more of the different types of similarities (outlined in Sections 2.2.1 to 2.2.3) with normalization techniques and the multiplication rule in order to improve system accuracy. An example of said methods is the similarity score calculation method proposed by Cappelli *et al.* (2010*b*), which combines the local similarity score calculation method with the structural similarity score calculation method. This is categorised as a hybrid strategy, since the local similarity score is robust to nonlinear distortion, while the structural similarity score calculation method is used to capture distinctiveness. Although the above technique is more frequently used within the point matching algorithm, it also addresses the different intra-class variations (to a certain degree) during similarity score calculation. This approach may however not address the problem of local inter-class similarity when both the local similarity score calculation method and the structural similarity score calculation method only consider the overlapping region.

The most popular trend is to combine the "percentage of the paired minutia points" similarity score calculation method and the local similarity score calculation method, as proposed by Jain *et al.* (2008), Zhu *et al.* (2005), Fu and Feng (2015), and Fu *et al.* (2013). Jain *et al.* (2008) account for partial overlap and local inter-class similarity by using a penalization factor on the "percentage of paired minutia points"-component of the score. These similarity score calculation methods do however not include the structural similarity of the paired minutia points on a global level and may therefore be sensitive to noise.

2.3 Performance of similarity score calculation methods

The current literature does not provide a clear answer as to how existing similarity score calculation methods perform. The performance measures employed for most existing minutia-based matching algorithms include *both* the errors incurred during the implementation of the point matching algorithm *and* the errors incurred during similarity score calculation. The performance of different similarity score calculation methods can therefore not be determined with confidence. Furthermore, each similarity score calculation method may be implemented on a very specific point matching algorithm. Khanyile *et al.* (2014) conducted the only study that the author is aware of, that compares *different* similarity score calculation methods by using the *same* point matching and minutia extraction algorithms. Their study however *only* focuses on "percentage of paired minutia points" similarity score calculation methods and incorporates point matching errors in the reported performance.

When the top 10 most accurate minutia-based matching algorithms (i.e. point matching and similarity score calculation algorithms) are considered, ranked according to their achieved EER as reported in the ongoing on-line Fingerprint Verification Competition (FVC) on 12 October 2015, only *two* algorithms have been formally published, namely the Local Greedy Similarity with Distortion-Tolerant Relaxation (LGS_DTR) algorithm by Cappelli *et al.* (2010*b*) and the M3gl algorithm by Medina-Pérez *et al.* (2012). The LGS_DTR algorithm averages a point matching relaxation step by Feng *et al.* (2006) on the MCC local similarity in order to calculate the similarity score, while the M3gl algorithm uses the conventional "percentage of paired minutia points" similarity score as proposed by Jea and Govindaraju (2005). The similarity score calculation methods employed by the above-mentioned top two minutia-based matching algorithms demonstrate that similarity score calculation methods have not been improved upon over the last decade and may not address all the different types of intra-variations and inter-class similarities.

There are *four* main reasons as to why similarity score calculation methods have not improved significantly over the last decade. Firstly, the system fails whenever the point matching algorithm fails, which has lead researchers to focus on making the point matching algorithms more robust with respect to intra-class variations. Secondly, multiple features or similarity scores can be more easily combined with machine learning techniques than through a manual method (Feng, 2008). Thirdly, the conventional "percentage of paired minutia points" similarity score calculation methods perform well when low levels of intra-class variations and inter-class similarities are present. Finally, the commercialization of automated verification systems limited the publication of advanced minutia-based matching algorithms.

The limited research on similarity score calculation methods (as outlined above) may result in a number of problems pertaining to minutia-based fingerprint matching. Firstly, since it is not clear which similarity score calculation method performs best, advanced minutia-based fingerprint matching algorithms continue to use conventional similarity score methods that cannot deal with different types of intra-class variations and inter-class similarities. Secondly, it is not exactly clear where the problem in minutia-based matching (i.e. point matching or similarity score calculation) lies, and how the similarity score calculation methods and/or point matching algorithms are affected by different types of intra-class variations and inter-class similarities. This leads to less proficient minutia-based fingerprint matching algorithms and inhibits progress.

2.4 Summary and conclusion

The last step of any fingerprint matching algorithm involves the calculation of the similarity score value in order to distinguish between genuine and impostor comparisons. Similarity scores can be categorised into three groups, based on how they are calculated, as well as their sensitivity and robustness with respect to intra-class variations and inter-class similarities: (1) percentage of paired minutia points, (2) local similarity score calculation methods, and (3) structural similarity score calculation methods. While each of these methods primarily deals with a specific type of intra-class variation and inter-class similarity, *no* method currently exists that combines *all* three types of similarity score calculation methods. Furthermore, the current literature does not establish whether the utilisation of existing similarity score calculation methods are sufficient in scenarios where high levels of intra-class variation or inter-class similarity are present and does not determine which similarity score calculation methods are the most proficient, since their performance measures also include those comparisons for which the point matching algorithm failed. The current literature does therefore *not* provide an indication as to the appropriate methodology for calculating a similarity score that can accurately distinguish between genuine and impostor comparisons when high levels of intra-class variations and inter-class similarities are present.

Chapter 3

Methodology

3.1 Introduction

A comparative study of the proficiency with which the similarity score values calculated through the methods detailed in Chapter 2, i.e. the literature review is conducted as follows.

Table 3.1: Identifiers assigned to each similarity score calculation method.

Paper	**Identifier**
Jain *et al.* (2008)	S_1
Cappelli *et al.* (2010*b*) (local method)	S_2
Cappelli *et al.* (2010*b*) (complete method)	S_3
Medina-Pérez *et al.* (2012)	S_4
Fu *et al.* (2013)	S_5
Khanyile *et al.* (2014)	S_6
Fu *et al.* (2013) (without the local similarity)	S_7

1. The minutia-based matching algorithms documented in the papers specified in Table 3.1 are used to calculate similarity score values for genuine and impostor comparisons using minutia points extracted from fingerprints considered for the 2002 and 2004 Fingerprint Verification Competitions (FVCs), i.e. FVC2002 and FVC2004, by implementing the FingerJet minutia point extraction software developed by DigitalPersona.

2. Each genuine comparison is classified into one of the following *eight* genuine comparison classes:

    - **GC1**, that contains genuine comparisons with high noise levels, i.e. scenarios where many unpaired minutia points are present within in the overlapping region of the convex hulls of the minutia points being compared;

- **GC2**, that contains genuine comparisons for which the minutia points from one of the fingers being compared are subjected to high levels of distortion, mostly in the form of stretching;
- **GC3**, that contains genuine comparisons with a small overlapping region;
- **GC4**, that contains genuine comparisons which satisfy all the criteria for classes **GC1** and **GC2**;
- **GC5**, that contains genuine comparisons which satisfy all the criteria for classes **GC1** and **GC3**;
- **GC6**, that contains genuine comparisons which satisfy all the criteria for classes **GC2** and **GC3**;
- **GC7**, that contains genuine comparisons which satisfy all the criteria for classes **GC1**, **GC2**, and **GC3**; and
- **GC8**, that contains genuine comparisons which do not satisfy any of the criteria for the above seven classes.

3. Each impostor comparison is classified into one of the following *three* impostor comparison classes:

   - **IC1**, that contains impostor comparisons with high levels of *global* inter-class similarity;
   - **IC2**, that contains impostor comparisons with high levels of *local* inter-class similarity; and
   - **IC3**, that contains impostor comparisons which do not meet any of the criteria for classes **IC1** and **IC2**.

4. The similarity score values are then used to estimate the False Match Rate (FMR) and False Non-Match Rate (FNMR) for different threshold values using Equations 3.1 and 3.2 for the FMR and FNMR respectively,

$$\mathrm{FMR}(t) = \frac{\mathrm{FP}(t)}{\mathrm{IC}}, \tag{3.1}$$

$$\mathrm{FNMR}(t) = \frac{\mathrm{FN}(t)}{\mathrm{GC}}, \tag{3.2}$$

   where:

   - t is the similarity score *threshold* value;
   - $\mathrm{FP}(t)$ is the number of false positives, i.e. the number of impostor comparisons with similarity score values *greater* than t;
   - IC is the total number of *impostor* comparisons;

- FN(t) is the number of false negatives, i.e. the number of genuine comparisons with similarity score values *less* than t; and
- GC is the total *number* of genuine comparisons.

5. The FMR and FNMR are then used to estimate the following performance indicators:

   - The Equal Error Rate (EER), that constitutes the value of the FMR or the FNMR, where their plots as functions of t intersect.
   - FMR_{zero}, that constitutes the value of the FNMR for which the FMR-value is equal to 0.
   - $\text{FNMR}_{\text{zero}}$, that constitutes the value of the FMR for which the FNMR-value is equal to 0.

   Note that:

   - The EER is used to estimate the proficiency of a specific similarity score value being used to tell impostor and genuine comparisons apart, but is only applicable to the specific database on which the FMR and FNMR are estimated.
   - The FMR_{zero} and $\text{FNMR}_{\text{zero}}$ further quantify the effect of impostor comparisons with similarity score values above the threshold or genuine comparisons with similarity score values below the threshold. Again this only applies to the specific database on which the FMR and FNMR are estimated.

6. Lastly, all impostor and genuine comparisons with similarity score values (as calculated through the methods outlined in Table 3.1) that are greater than or less than the similarity score value associated with the EER (for each respective method) are identified as *problematic* comparisons. The number of problematic comparisons for each similarity score calculation method is recorded and used to separate said methods into different genuine and impostor comparison classes in order to get an idea as to which inter-class similarities or inter-class variations these different methods are sensitive to, either individually or in combination.

The similarity score calculation methods documented in the papers specified in Table 3.1 are selected for comparison purposes, since these papers provide significant detail on how the similarity score in question is calculated. They also clearly specify whether the methods presented in Chapter 2 are implemented individually or in combination. FingerJet is used to extract minutia points from the fingerprints since this software is freely available, is MINEX certified, and has been shown to perform in the top 10% of minutia point extraction software when software from 68 different vendors are considered.

The above process is performed on only a few selected FVC2002 and FVC2004 fingerprint databases, and steps 2 and 6 are only performed on one database from each of the FVC2002 and FVC2004 fingerprint databases. Impostor and genuine comparisons are identified for each database, using the FVC protocol.

This chapter provides details on the:

- FVC2002 and FVC2004 fingerprint databases, specifying the databases on which the above-mentioned procedure is implemented, and providing a justification for excluding certain databases;
- Criteria used to assign genuine and impostor comparisons to the classes as defined above; and
- Algorithms that are implemented on paired minutia points in order to ensure that minutia point matching errors are not incorporated into the results; the calculation of a similarity score is based on the *same* paired minutia points for genuine comparisons. This procedure is followed since most of the matching algorithms pair minutia points using *different* methods, which may lead to differences in performance.

3.2 The fingerprint databases considered for proficiency testing

In order to evaluate the proficiency of the different similarity score calculation methods in accurately discriminating between genuine and impostor comparisons, fingerprints from the 2002 and 2004 Fingerprint Verification Competitions (FVCs) were considered. This chapter does not provide a detailed report on the data collection and image acquisition protocol, since said protocol is documented in Maio *et al.* (2002) and Maio *et al.* (2004). However, this chapter does mention the specifications that may have an influence on the different types of intra-class variations and inter-class similarities.

The fingerprints for both competitions are arranged into *four* databases, DB1, DB2, DB3, and DB4, which are further sub-divided into database A and database B. All the A-databases contain 800 fingerprints from 100 fingers with eight fingerprints per finger, while the B-databases contain 80 fingerprints from 10 fingers with eight fingerprints per finger. However, only databases DB1_A, DB2_A, and DB3_A (associated with *both* competitions) are used for evaluation purposes, since database DB4 contains synthetic fingerprints and the B-databases do not contain a sufficient number of samples.

3.2.1 The FVC2002 fingerprint databases

Databases DB1_A, DB2_A, and DB3_A for FVC2002 contain fingerprints of the forefingers and middle fingers of both hands of three randomly selected groups of 30 students with an average age of 20 years. No intentional acquisition differences were introduced between these three groups. Furthermore, no minimum quality standard was required for the fingerprint images to be included in the databases and the surfaces of the sensors were not cleaned between acquisitions. During three different sessions, two weeks apart, volunteers had to present four impressions of each of the four fingers in question. The volunteers were requested to translate and rotate their fingers during the second session when two of the four fingerprints were captured, but the rotation was restricted to 35°. During the third session, the first and second fingerprints were captured after drying the finger, while the third and fourth fingerprints were captured after the finger was moistened. Therefore, 12 fingerprints were collected from a total of 120 fingers.

Fingerprints of the same finger were then categorised into groups, after which said groups were arranged according to the average quality of the fingerprints within each group. The fingerprints among all 10 groups with the highest image quality, as well as the four fingerprints within each group with the highest quality were subsequently removed. The database was consequently reduced to fingerprints from 110 fingers with eight fingerprints for each finger, and divided into databases A and B as outlined above.

3.2.2 The FVC2004 fingerprint databases

The same procedure was followed for the FVC2004 databases in order to acquire fingerprints for databases DB1_A, DB2_A, and DB3_A. The average age of a student volunteer was however 24 years. During the second session, the volunteers were requested to introduce distortion during the acquisition of fingerprints 1 and 2, as well as to rotate their fingers during the acquisition of fingerprints 3 and 4. All the images captured in both of these databases were captured at a resolution of approximately 500 dpi.

3.3 Classification of impostor comparisons

In this study the impostor comparisons are classified into three groups, **IC1-IC3**, by applying thresholds to two measures that capture the global and local inter-class similarities. The *global* inter-class similarity measure, S_{GS}, is calculated as follows,

$$S_{\mathrm{GS}} = \left[\frac{2N_{\mathrm{m}}}{N_{\mathrm{TO}} + N_{\mathrm{QO}}}\right] \cdot \left[\frac{N_{\mathrm{TO}} + N_{\mathrm{QO}}}{N_{\mathrm{T}} + N_{\mathrm{Q}}}\right], \tag{3.3}$$

while the *local* inter-class similarity measure, S_{LS}, is calculated as follows,

$$S_{\mathrm{LS}} = \left[\frac{2N_{\mathrm{m}}}{N_{\mathrm{TO}} + N_{\mathrm{QO}}}\right] \cdot \left[1 - \left(\frac{N_{\mathrm{TO}} + N_{\mathrm{QO}}}{N_{\mathrm{T}} + N_{\mathrm{Q}}}\right)\right], \tag{3.4}$$

with N_{m}, N_{TO}, N_{QO}, N_{T}, and N_{Q} as defined in Chapter 2. If a comparison is associated with a S_{GS}-measure that is equal to or greater than 0.1896, it is classified as a comparison associated with high levels of *global* inter-class similarity (group **IC1**), while comparisons with a S_{LS}-measure equal to or greater than 0.1298 are classified as comparisons associated with high levels of *local* inter-class similarity (group **IC2**). When a comparison does not satisfy either one of the above two conditions, it is classified as belonging to group **IC3**. The protocol that is followed to determine the above thresholds is explained in more detail in Appendix A.

3.4 Classification of genuine comparisons

Each genuine comparison is compared to three thresholds in order to determine to which of the eight groups, **GC1**-**GC8**, as mentioned in the introduction, it belongs. These three thresholds are applied to the following three measures:

- The ratio of paired to unpaired minutia points in the overlapping region; this is used to estimate the level of *noise* that is present in a fingerprint comparison.
- The average Euclidean distance between minutia points that form pairs; this is used to estimate the level of *nonlinear distortion* that is present in a fingerprint comparison.
- The size of the overlapping region; this is used to estimate the *extent of the partial overlap* in a fingerprint comparison.

The thresholds applied to the above-mentioned three measures are 0.50, 7.00 pixels, and 2500 squared pixels, respectively. The protocol that is followed to determine the above thresholds is also explained in more detail in Appendix A.

3.5 Matching of minutia points

The similarity score calculation methods documented in the papers listed in Table 3.1 require a list of paired minutia points and also involves the calculation of the local similarity between the paired points. Additionally, the paired minutia points for similarity score calculation methods S_1 and $S_4 - S_7$ are based on the *spatial and orientation* similarity after alignment, while the paired minutia points for similarity score calculation methods S_2 and S_3 are based on the *local similarity* between the minutia points. The pairs in the *first* group of similarity score calculation methods are therefore identified by applying two thresholds (i.e. 15 pixels and 30°) to the aligned minutia sets. The pairs of minutia points in the *second* group of similarity score calculation methods are obtained by applying the Local Greedy Similarity (LGS) assignment method, as developed by Cappelli *et al.* (2010*b*), to the local similarity matrix. The procedure for determining this alignment and the local similarity matrix is explained in the following subsection.

3.5.1 Computation of alignment parameters

The parameters required to transform (i.e. rotate and translate) minutia points of one fingerprint in such a way that they form a similar pattern than (are *aligned* to) those of another fingerprint (it is being compared to) are determined as follows:

1. The local similarity values (as described in Chapter 2) between each minutia point of one fingerprint and all the minutia points of another fingerprint (it is being compared to) is first calculated. The resulting local similarity values are stored in a similarity matrix with both the number of rows and columns equal to the number of minutia points within the fingerprints being compared.

2. The three minutia points that matches the best are then identified by processing the similarity matrix of step 1.

3. Finally, the Kabsch algorithm (Kabsch, 1976) is applied in order to calculate the extent of rotation and translation that is required to align the three most similar pairs of minutia points (as identified in step 2).

The local similarity between minutia points is calculated using the Minutia Cylinder Code (MCC) descriptor within version 1.4 of the MCC Software Development Kit (SDK) with parameters as specified by Cappelli *et al.* (2010*b*). In order to identify the three best matching minutia points, the Local Greedy Similarity (LGS) assignment algorithm proposed by Cappelli *et al.* (2010*b*) is applied to the similarity matrix, so that a list of the n-best pairs of matching minutia points are produced in descending order based on the local similarity value, where n equals the number of minutia points in the fingerprint with the fewer minutia points. The distortion tolerant relaxation (DTR) algorithm proposed by Cappelli *et al.* (2010*b*) is subsequently applied to the local similarity of the paired minutia points in order to calculate similarity values that are based on how well the paired minutia points align. The paired minutia points are rearranged in descending order according to the ratio of the *new* similarity to the *local* similarity, and the three best matching minutia points are identified from the new list.

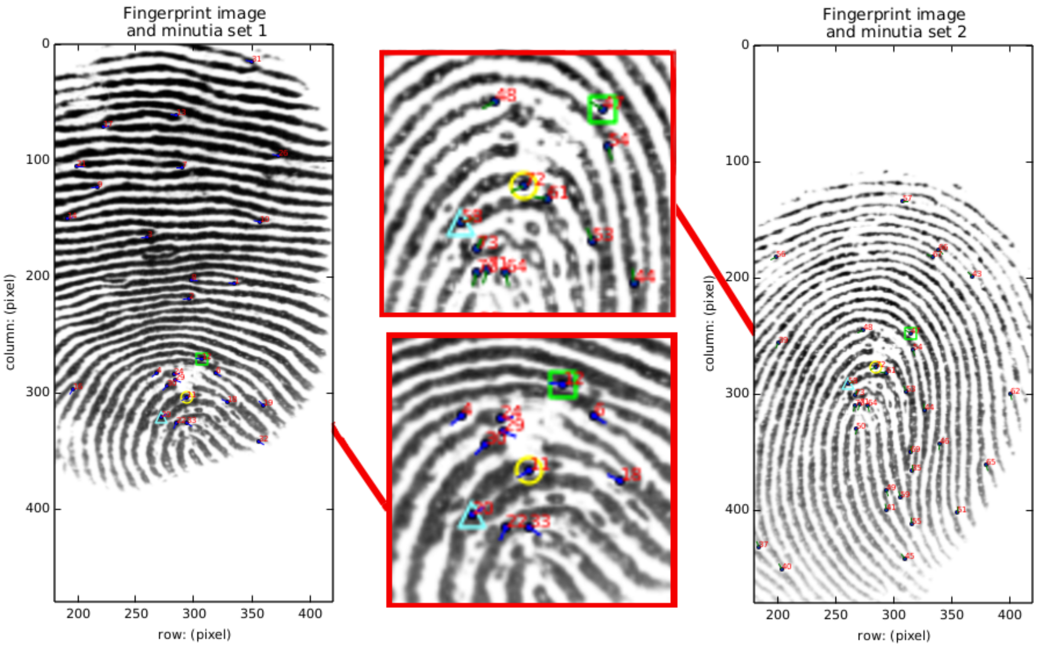


Figure 3.1: A comparison that is *manually* inspected in order to determine whether the point matching algorithm's three most similar minutia pairs correspond or not, and in so doing determine whether the point matching algorithm succeeded or not.

3.5.2 Comparisons with incorrectly paired minutia points

Genuine comparisons with incorrectly paired minutia points are identified by *manually* comparing the locations of the three most similar paired minutia points, as explained in Section 3.5.1. Figure 3.1 depicts typical images that are used to visually compare the three most similar pairs of minutia points. This comparison is however only performed on a subset of genuine comparisons. This subset consists of comparisons for which the similarity score values (as calculated using the methods proposed by Fu *et al.* (2013), Jain *et al.* (2008), and Cappelli *et al.* (2010*b*)) are lower than the similarity score value at the EER of *at least one* of the score value distributions for *each* similarity score calculation method when implemented on the database that the comparison is associated with. Only these subsets of genuine comparisons are visually compared due to the impracticality of manually comparing all of the 2800 genuine comparisons within all six databases. The *erroneous comparisons*[1] that are removed are listed in Appendix B.

[1]An erroneous comparison is defined as a comparison for which the point matching algorithm failed to correctly identify the three most similar paired minutia points.

3.6 Summary

This chapter outlined the methodology followed in this thesis to compare the performance of seven existing similarity score calculation methods and constitutes *three* significant improvements to existing methodologies. Firstly, it considers the *same* minutia extractor and point matching algorithm in order to produce the data for the similarity score calculation methods. Secondly, it *removes* comparisons where the alignment procedure and point matching algorithm completely fail and therefore prevents these erroneous comparisons from adversely affecting the proficiency of the similarity score calculation methods being investigated. Thirdly, it analyses the genuine and impostor comparisons that these similarity score calculation methods fail to correctly classify and therefore provides some insight into the strengths and weaknesses of these similarity score calculation methods as will be discussed in more detail in the following chapter.

Chapter 4

Results and analysis

This chapter presents the results that are obtained when the methodology presented in Chapter 3 is implemented and also provides a detailed analysis of said results. Guidelines can therefore be formulated for calculating a similarity score that is able to more accurately tell genuine and impostor comparisons apart, taking both inter-class similarities and intra-class variations into account.

4.1 Results

4.1.1 Erroneous comparisons

As explained in Chapter 3, the methodology consists of three stages: (1) minutia extraction, (2) point matching, and (3) similarity scores calculation. A comparison may therefore be incorrectly classified as either a genuine or an impostor comparison due to the failure of *any* of the above three stages. However, as previously explained, the erroneous comparisons identified during the first two stages are *removed* from the experiment in order to prevent these errors from affecting the final stage. The number of erroneous comparisons identified during each stage are listed in Table 4.1.

It is clear that the minutia extractor successfully extracts the minutia points for all the fingerprints in the FVC2002 DB1_A and FVC2002 DB2_A databases, but however fails for a certain number of fingerprints in the other databases. The point matching algorithm and *best* similarity score calculation method produce a similar number of erroneous comparisons in each of the FVC2002 fingerprint databases, while the point matching algorithm produces slightly more erroneous comparisons when implemented on the FVC2004 databases. These results do however confirm that the similarity score calculation method utilised also contributes towards the error rate for minutia-based matching algorithms.

Table 4.1: The number of erroneous comparisons within each database identified during the different stages of the experiment.

Database	Stage 1	Stage 2	Stage 3
FVC2002 DB1_A	0	21	23
FVC2002 DB2_A	0	20	29
FVC2002 DB3_A	341	96	97
FVC2004 DB1_A	41	193	157
FVC2004 DB2_A	124	189	118
FVC2004 DB3_A	886	156	74

4.1.2 Performance evaluation

The values obtained for the EER, FMR_{zero}, and $\text{FNMR}_{\text{zero}}$ when the similarity score calculation methods proposed in each of the seven papers listed in Table 3.1 is implemented for genuine and impostor comparisons on each of the FVC2002 and FVC2004 fingerprint databases, are shown in Tables 4.2 and 4.3. The following observations are made.

1. The EER, FMR_{zero}, and $\text{FNMR}_{\text{zero}}$ obtained for the FVC2002 fingerprint databases are generally lower than those for the FVC2004 fingerprint databases. This indicates that it is more difficult to tell genuine and impostor comparisons apart for the fingerprints in the FVC2004 databases than is the case for those in the FVC2002 databases.

2. The EERs obtained when similarity score calculation method S_3 is used, is the *lowest* for five out of the six FVC2002 and FVC2004 fingerprint databases.

3. The EERs obtained when similarity score calculation method S_4 is used, is the *highest* for all of the FVC2002 fingerprint databases.

4. The EERs obtained when similarity score calculation method S_2 is used, is the *highest* for two of the three FVC2004 fingerprint databases.

5. The ranking of the similarity score calculation methods based on the EER is not consistent throughout all the fingerprint databases.

6. The $\text{FNMR}_{\text{zero}}$-values obtained when similarity score calculation method S_2 is used, is generally higher than those obtained through other methods.

7. The lowest $\text{FNMR}_{\text{zero}}$ is obtained when the following similarity score calculation methods are used:

    - S_5 for two of the three FVC2002 fingerprint databases;

Table 4.2: The performance measures for the different similarity score calculation methods when implemented on the fingerprint databases for *FVC2002*. For each database the boxed entries represent the best three performances, when *all* three measures are considered. The three lowest and highest error rates for each *individual* measure are respectively denoted in boldface and underlined.

Method	EER %	FNMR$_{\text{zero}}$ %	FMR$_{\text{zero}}$ %
		DB_1	
S_1	**0.47**	**9.88**	**3.99**
S_2	0.73	$\underline{65.96}$	**2.20**
S_3	**0.47**	38.30	$\boxed{\mathbf{1.76}}$
S_4	$\underline{2.23}$	41.72	13.57
S_5	$\boxed{\mathbf{0.43}}$	$\boxed{\mathbf{7.52}}$	9.86
S_6	1.92	41.29	12.49
S_7	0.76	**14.95**	$\underline{52.21}$
		DB_2	
S_1	**0.64**	**17.64**	**2.66**
S_2	0.74	$\underline{76.55}$	10.61
S_3	$\boxed{\mathbf{0.36}}$	$\boxed{\mathbf{11.11}}$	$\boxed{\mathbf{1.15}}$
S_4	$\underline{1.64}$	33.47	12.45
S_5	**0.53**	**16.85**	**3.78**
S_6	1.39	32.77	11.22
S_7	0.83	22.77	$\underline{13.78}$
		DB_3	
S_1	**1.46**	**25.18**	**13.32**
S_2	2.63	$\underline{96.76}$	**13.71**
S_3	$\boxed{\mathbf{1.36}}$	38.21	$\boxed{\mathbf{6.52}}$
S_4	$\underline{4.60}$	60.49	24.26
S_5	**1.61**	$\boxed{\mathbf{16.22}}$	29.84
S_6	4.09	60.05	21.41
S_7	2.15	**21.65**	$\underline{88.63}$

- S_6 for two of the three FVC2004 fingerprint databases; and
- S_3 for one of the three FVC2002 fingerprint databases and for one of the three FVC2004 fingerprint databases.

8. For all the databases, the lowest FMR$_{\text{zero}}$ is obtained when similarity score calculation method S_3 is implemented.

9. The EER, FMR$_{\text{zero}}$, and FNMR$_{\text{zero}}$ obtained when similarity score calculation method S_7 is used, are higher than those obtained for similarity

Table 4.3: The performance measures for the different similarity score calculation methods when implemented on the fingerprint databases for *FVC2004*. For each database the boxed entries represent the best three performances, when *all* three measures are considered. The three lowest and highest error rates for each *individual* measure are respectively denoted in boldface and underlined.

Method	**EER** %	**FNMR**$_{\text{zero}}$ %	**FMR**$_{\text{zero}}$ %
		DB_1	
S_1	**3.15**	<u>100.0</u>	**28.18**
S_2	4.27	99.11	30.63
S_3	$\boxed{\mathbf{2.09}}$	$\boxed{\mathbf{77.25}}$	$\boxed{\mathbf{14.81}}$
S_4	3.77	<u>100.0</u>	**28.68**
S_5	3.75	<u>100.0</u>	34.41
S_6	**3.66**	**88.34**	30.28
S_7	<u>4.70</u>	**78.73**	<u>50.58</u>
		DB_2	
S_1	**2.88**	37.68	**33.31**
S_2	<u>5.20</u>	<u>98.48</u>	48.29
S_3	$\boxed{\mathbf{1.62}}$	42.32	$\boxed{\mathbf{19.49}}$
S_4	3.97	**36.85**	43.39
S_5	3.69	49.74	45.48
S_6	**3.57**	$\boxed{\mathbf{32.48}}$	**37.16**
S_7	3.98	**37.15**	<u>88.59</u>
		DB_3	
S_1	**2.66**	51.26	43.50
S_2	<u>5.25</u>	<u>95.58</u>	29.59
S_3	$\boxed{\mathbf{1.16}}$	62.83	$\boxed{\mathbf{7.04}}$
S_4	**2.69**	**47.97**	**22.22**
S_5	2.83	**49.44**	77.33
S_6	2.72	$\boxed{\mathbf{47.21}}$	**21.85**
S_7	3.27	52.64	<u>93.29</u>

score calculation method S_5, except for two occasions in the case of the FNMR$_{\text{zero}}$. Note that similarity score calculation method S_7 is equivalent to method S_5, except that the local similarity component is not present.

10. The EERs obtained when similarity score calculation method S_5 is used, always rank within the top *three* lowest EERs for the FVC2002 fingerprint databases. They also always rank within the top *four* highest EERs when the FVC2004 fingerprint databases are considered.

11. The EERs obtained when similarity score calculation methods S_1 and S_3 are used, always rank within the top *three* lowest EERs for the FVC2002 and FVC2004 fingerprint databases.

12. The EERs obtained when similarity score calculation method S_6 is used, rank within the top *three* lowest EERs for two of the three FVC2004 fingerprint databases.

13. Similarity scores obtained through methods that combine the *local* similarity score with either the *structural* similarity score or the *percentage of paired minutia points*, generally result in a lower EER, FMR_{zero}, and $\text{FNMR}_{\text{zero}}$.

4.1.3 Classification of impostor and genuine comparisons

Table 4.4: Results for *genuine* comparisons. The number of *genuine* comparisons that is categorised into each group for the FVC2002 DB1_A and FVC2004 DB1_A databases are depicted in boldface, while the number of *problematic* comparisons in each group produced by implementing the different similarity score calculation methods are shown in normal font.

	GC1	GC2	GC3	GC4	GC5	GC6	GC7	GC8	Total
				FVC	**2002**	DB1_A			
	3	**256**	**684**	**11**	**1**	**28**	**0**	**1796**	**2775**
S_1	2	0	1	9	1	0	0	0	13
S_2	0	1	16	0	0	3	0	0	20
S_3	0	0	12	0	0	1	0	0	13
S_4	1	1	53	4	0	2	0	1	62
S_5	2	0	1	9	0	0	0	0	12
S_6	1	1	44	4	0	2	0	1	53
S_7	3	5	0	11	1	0	0	1	21
				FVC	**2004**	DB1_A			
	71	**678**	**296**	**234**	**16**	**58**	**19**	**1194**	**2566**
S_1	10	0	0	56	9	0	5	0	80
S_2	21	12	9	31	6	9	9	9	106
S_3	2	7	4	20	6	5	9	1	54
S_4	10	5	20	33	12	4	14	0	98
S_5	19	0	0	70	5	0	1	0	95
S_6	11	5	17	33	10	4	13	1	94
S_7	21	0	0	91	7	0	2	0	121

Table 4.5: Results for *impostor* comparisons. The number of *impostor* comparisons that is categorised into each group for the FVC2002 DB1_A and FVC2004 DB1_A databases are depicted in boldface, while the number of *problematic* comparisons in each group produced by implementing the different similarity score calculation methods are shown in normal font.

Method	**IC1**	**IC2**	**IC3**	**Total**
		FVC2002	DB1_A	
	940	**235**	**3775**	**4950**
S_1	17	6	0	23
S_2	22	0	15	37
S_3	21	0	2	23
S_4	110	0	0	110
S_5	10	11	0	21
S_6	96	0	0	96
S_7	24	14	0	38
		FVC2004	DB1_A	
	840	**390**	**3720**	**4950**
S_1	141	17	0	158
S_2	95	15	108	218
S_3	77	10	16	103
S_4	184	0	0	184
S_5	146	42	0	188
S_6	181	0	0	181
S_7	176	56	0	232

The number of genuine and impostor comparisons from the FVC2002 DB1_A and FVC2004 DB1_A fingerprint databases that are categorised into the different genuine and impostor comparison classes are shown in Tables 4.4 and 4.5 respectively. The number of genuine and impostor comparisons within each class that are *incorrectly* classified are also shown. For each method, the similarity score threshold which results in FMR = FNMR is imposed.

The following are observed from the data presented in Tables 4.4 and 4.5.

1. There are more genuine comparisons with high levels of intra-class variations in the FVC2004 DB1_A fingerprint database. This possibly explains why a lower EER, FMR_{zero}, and $\text{FNMR}_{\text{zero}}$ are obtained for all the similarity score calculation methods when they are implemented on fingerprints in the FVC2002 DB1_A database than is the case for the FVC2004 DB1_A database.

2. The number of imposter comparisons in the FVC2002 DB1_A and FVC2004 DB1_A databases that are categorised into the first two classes of impostor comparisons are comparable (e.g. 1175 versus 1230). However, there are

    - *more* impostor comparisons from the FVC2002 DB1_A database that fall under the **IC1** class (i.e. 940) than is the case for the FVC2004 DB1_A database (i.e. 840), and
    - *less* imposter comparisons from the FVC2002 DB1_A database that fall under the **IC2** class (i.e. 235) than is the case for the FVC2004 DB1_A database (i.e. 390).

3. Genuine comparisons that fall under the **GC3** class, i.e. genuine comparisons with a partial overlap, are classified incorrectly when similarity score calculation method S_2, S_3, S_4, or S_6 is implemented.

4. Genuine comparisons that fall under the **GC1** class, i.e. genuine comparisons where false minutia points are extracted from one or both of the fingerprints being compared, are less frequently incorrectly classified when similarity score calculation method S_1, S_3, S_4, or S_6 is implemented.

5. Genuine comparisons that fall under the **GC2** class, i.e. genuine comparisons where one or both fingerprints being compared are subjected to nonlinear distortion, are less frequently incorrectly classified when similarity score calculation method S_1 or S_5 is implemented.

6. Imposter comparisons that fall under the **IC3** class, i.e. imposter comparisons that do not meet the criteria for both **IC1** and **IC2**, are frequently incorrectly classified when similarity score calculation method S_2 or S_3 is implemented. This may explain the fact that the $\text{FNMR}_{\text{zero}}$ obtained when method S_2 or S_3 is implemented, is generally higher than is the case for other methods.

7. Imposter comparisons that fall under the **IC2** class, i.e. imposter comparisons with high levels of *local* inter-class similarity, are correctly classified when method S_4 or S_6 is implemented.

8. Imposter comparisons that fall under the **IC1** class, i.e. imposter comparisons with high levels of *global* inter-class similarity, are incorrectly classified when method S_4, S_6, or S_7 is implemented.

4.2 Analysis

It can be concluded that not one of the seven tested similarity score calculation methods is superior to all the others. It does however appear that S_3 is the best performing similarity score calculation method, since it has the lowest EER for five of the six databases, even though the margin is very small for two of these databases. Therefore, for the seven similarity scores tested, there is no *single* score that can be used in *isolation* to construct basic guidelines for developing a proficient similarity score calculation method.

However, when the three best similarity scores for FVC2002 DB1_A and FVC2004 DB1_A, as well as the number of comparisons with high levels of intra-class variations and inter-class similarities in said databases are considered, a pattern starts to emerge. The results show that both of these databases are characterised by different *levels* and different *types* of intra-class variations and inter-class similarities. FVC2002 DB1_A contains more partial overlapping sets and higher levels of *global* inter-class similarity, while FVC2004 DB1_A contains more noise, nonlinear distortion and *local* inter-class similarity. The emerging pattern lies in the fact that, for both of these databases, the similarity scores that better address inter-class similarity and are not sensitive to intra-class similarity perform the best.

For example, S_5 performs the best on FVC2002 DB1_A, since it is not affected by partial overlap – it only considers the overlapping region – but still addresses global inter-class similarity, since it considers the complete overlap and also takes the local similarity of each pair of minutia points into account. However, since S_5 is highly sensitive to local inter-class similarity, it is less proficient when implemented on FVC2004 DB1_A.

On the other hand, S_3, S_1, and S_6 all address local inter-class similarity, where S_1 and S_6 include penalization factors and S_3 uses more pairs of minutia points than is present in the small overlapping region. This may contribute to these scores performing better (than S_5) on FVC2004 DB1_A, but it also increases their sensitivity to partial overlap which contributes to them performing worse (than S_5) on FVC2002 DB1_A. Furthermore, S_3 is robust with respect to noise, while S_1 and S_6 are somewhat robust with respect to nonlinear distortion, which explains why these methods are proficient when implemented on the databases considered.

The question as to why S_1 and S_3 are highly proficient across both databases remain. The answer is clear from the statistics for the *problematic* comparisons (see Tables 4.4 and 4.5), since *both* of these similarity scores address *both* types of inter-class similarity. Note that global inter-class similarity causes most of the problematic comparisons for all of the similarity score. These similarity scores are also not sensitive to many types of intra-class variations. S_3 uses the structural similarity combined with the local similarity and is therefore more robust with respect to noise for genuine comparisons, while it also captures the dissimilarity between the paired minutia points in cases of global inter-class similarity. It also addresses local inter-class similarity (as explained earlier). S_1 performs the best out of the remaining similarity score calculation methods, since (like S_5) it addresses global inter-class similarity, but also (like S_6) addresses local inter-class similarity through a penalization factor.

This also shows that the two similarity score calculation methods that address *both* global *and* local inter-class similarities may be more proficient across a range of databases, but not necessarily optimal for a specific database. This may be due to the fact that since they address *both* types of inter-class similarities, they might be slightly sensitive to *extreme* cases of intra-class variations. Therefore, in extreme cases of partial overlap and local inter-class similarity, the minutia point comparisons are very similar as is the case with high levels of noise, nonlinear distortion and global inter-class similarity. This may explain why most existing approaches focus on improving the point matching stage and address nonlinear distortion during that stage.

From these results, the following *guidelines* are proposed for calculating a proficient similarity score:

- Firstly, a *combination* of the *local* and *structural* similarity as proposed by Cappelli *et al.* (2010*b*) results in an accurate similarity score, since it constitutes the most robust type of score with respect to noise, while it also best captures the dissimilarity of comparisons with high levels of global inter-class similarity.

- Secondly, *penalization factors* such as those proposed by Jain *et al.* (2008) and Khanyile *et al.* (2014) should be included in order to prevent impostor comparisons with high degrees of local inter-class similarity to be assigned high similarity score values.

- Thirdly, the *percentage of paired minutia points* and the *local similarity* should be employed, since they are more robust to nonlinear distortion than is the case for the structural similarity, and also include *all* the minutia points. This percentage of paired minutia points term should however be carefully combined with the structural similarity so that its sensitivity to noise does not affect said structural similarity.

- The percentage of paired minutia points in the overlap achieves a higher performance when combined with the local similarity, since it better captures the dissimilarity for impostor comparisons with high levels of local and global inter-class similarity.

It is important to note that the results on which these guidelines are based are affected by the minutia extraction and point matching algorithms. Therefore, for point matching algorithms that are not affected by nonlinear distortion and/or do not produce comparisons with high levels of local inter-class similarity, these guidelines need to be changed.

The next step is to consider possible problems associated with the abovementioned guidelines. Firstly, the local similarity forms an important part of the best similarity score calculation methods, but achieves a low accuracy when implemented in a stand-alone fashion on the FVC2004 databases. The hypothesis is that the method proposed by Cappelli *et al.* (2010*b*) is highly sensitive to noise which may cause an increase in EER. However, there are many ways to calculate the local similarity between two minutia points. *Different strategies for calculating the local similarity is therefore compared in Chapter 5.*

Furthermore, the results in this chapter indicate that the structural similarity is robust to noise, but is also sensitive to nonlinear distortion, while the percentage of paired minutia points is slightly robust to nonlinear distortion, but also sensitive to noise. Therefore, if these two similarity scores can be *combined* in order to capture the strengths of *both* methods, the performance may be improved. *Chapter 6 therefore investigates the combination of different improved similarity score calculation methods in such a way that training is not required.*

In summary, this chapter provided some basic *guidelines* for calculating a similarity score. Furthermore, it explained the weaknesses and strengths of the best *existing* similarity scores and how these are affected by intra-class variations and inter-class similarities. These results demonstrated that no single existing similarity score is superior to all the others, since the performance depends on the database being considered. However, it appears that similarity score calculation methods that combine the local similarity with either the structural similarity or the percentage of paired minutia points, and then combine either of these results with a penalization factor to account for local inter-class similarity produce the best EER, $\text{FNMR}_{\text{zero}}$, and FMR_{zero}, since these strategies appear to be more robust with respect to noise and nonlinear distortion.

Chapter 5

Comparing local similarity score calculation methods

5.1 Introduction

Local similarity score calculation forms an integral part of any accurate minutia-based similarity score calculation method. This is clear from the results presented in Chapter 4 where the best performing similarity score calculation methods combine the *local* similarity score calculation method with either the percentage of paired minutia points or the structural similarity score calculation method. The explanation for these results (as detailed in Chapter 4) is that the local similarity measure deals well with global inter-class similarity. However, these results also show that, on an individual basis, the Minutia Cylinder Code (MCC), i.e. local similarity score calculation method S_2, does not perform the best on the FVC2002 fingerprint databases and performs the worst on the FVC2004 fingerprint databases based on the equal error rate (EER), since it is highly sensitive to noise. Consequently, improving the local similarity score calculation method may have a significant impact on the performance of the best available similarity score calculation methods.

Local similarity score calculation involves two *stages*: (1) the *computation* of the local similarity values and (2) the *combination* of the local similarity values for a number of paired minutia points in order to obtain the *final* local similarity score. The local similarity value computed during stage 1 represents the degree of similarity between two minutia points from *different* sets based on the feature differences between their neighbouring minutia points. During stage 2, the pairs of minutia points between the two sets are either identified through an assignment-based method which is implemented on the local similarity matrix or by employing a tolerance box on the aligned minutia sets. The local similarity between these identified paired minutia points may then be *averaged* in order to obtain the final local similarity score. For a more detailed explanation of the two stages relevant to local similarity score calculation, see

Section 2.2.2.

There are several different approaches to implementing the above two stages in such a way that the approach is robust with respect to noise. During stage 1, the neighbouring minutia points can be paired either by applying thresholds as proposed by Jain *et al.* (2008), or by considering their distance from, or orientation with respect to the reference minutia points as proposed by Jiang and Yau (2000). Both of these approaches may be more robust with respect to noise than is the case for the MCC-based local similarity measure. Furthermore, most approaches employ a tolerance box-based approach during stage 2 to identify the paired minutia points, which is also more robust with respect to noise as is the cause when a fixed number of the best minutia pairs are chosen. Alternatively, these different strategies being employed during stage 1 may be combined with stage 2 by implementing a tolerance box-based approach.

No existing studies in the current literature compare different strategies during stage 1 and/or stage 2 of local similarity score calculation. Studies such as the one conducted by Peralta *et al.* (2015) compare different local minutia-based matching methods. These methods focus more on comparing the point matching capabilities of local descriptors than on how accurate they are in distinguishing between genuine and impostor comparisons during the similarity score calculation stage.

The testing in this chapter therefore focuses on two questions. The first question is as follows: Which method best calculates the local similarity between two descriptors? This question is investigated by considering the similarity between descriptors based on the similarity of neighbouring minutia pairs and the percentage of paired neighbouring minutia points. However, we also consider the average neighbouring similarity between different numbers of pairs in order to gauge the effect of noise and investigate distinctiveness.

The second question is as follows: How many minutia pairs between the two sets' local similarity have to be averaged in order to obtain an accurate local similarity score? Is it better to consider *all* the minutia pairs, which may be sensitive with respect to noise, but also capture the dissimilarity for impostor comparisons, or is it better to only use a small subset of minutia pairs, which is robust with respect to noise, but is not particularly distinctive? Finally, how does the performance of a threshold-based approach to choosing the number of paired minutia points compare to the assignment-based approach?

The rest of this chapter is divided into five sections, where Section 5.2 provides an overview of the methodology. In Section 5.3 a new descriptor is created to investigate the adoption of the different strategies during stage 1 and to suggest and investigate certain improvements such as increasing the robustness with respect to noise for the local similarity calculations. Section 5.4 details the different strategies utilised during stage 2, while the results are presented in Section 5.5. A discussion and conclusion based on these results are finally presented in Section 5.6.

5.2 Methodology

The core of the methodology lies in the comparison of the performance of *local* similarity score calculation methods utilising *different* strategies during stage 1 and 2. These local similarity score calculation methods are implemented on *six* databases (the databases labelled "A" from FVC2002 and FVC2004) with the testing protocol and preprocessing procedures as described in Chapter 3. However, for all the local similarity scores that require an assignment approach for identifying the paired minutia points, only the pairs within the overlap are considered in the assignment matrix. A more detailed explanation of the individual local similarity calculation methods and descriptors are provided in Section 5.3, while Section 5.4 provides an overview of the strategies employed during stage 2. The parameters used in the proposed local similarity score calculation methods are listed in Table 5.1. However, these parameters are specifically for fingerprint images captured at 500 dpi and needed to be adjusted for fingerprint captured at different a resolution. After the similarity score values have been calculated for all the comparisons, the EER and area under the Receiver Operating Characteristic Curve (AUC) are employed as performance measures for each variant of the local similarity score. These performance measures are subsequently compared for all the different variations of local similarity score calculation methods in order to determine which method is the most proficient. The final part of this protocol involves the determination as to whether the best performing local similarity measure performs *significantly* better than the MCC-based local similarity. For this purpose, the dependent (paired) t-statistic is implemented by considering the EER at a 0.05 level of significance.

Table 5.1: The parameters for the *local* similarity algorithm.

Parameter	**Value**
Λ	90.0 pixels
ω	1.2 radians
ϵ_d^ρ	6.0
τ_d^ρ	-1.2
ϵ_δ^ρ	0.15
τ_δ^ρ	-20.0
ϵ_ϕ^ρ	0.15
τ_ϕ^ρ	-20.0

5.3 Stage 1: Local similarity computation

The proposed local similarity algorithm as employed during the first stage involves two steps. Section 5.3.1 details the first step, namely the *creation* of the descriptors, while Section 5.3.2 elaborates on the second step, namely the process of *comparing* two descriptors. Section 5.3.2.1 outlines the advantages and disadvantages of this algorithm and elaborates on how it differs from existing local similarity calculation methods.

5.3.1 Local valid sector descriptor

We propose a fixed radius descriptor that is robust with respect to partial overlapping sets and includes the neighbouring minutia points within a radius of Λ. Each descriptor is divided into 24 sectors as demonstrated in Figure 5.1. They involve *eight* orientation sectors of equal size and *three* radial sectors at radial distances of 40.0, 70.0 and 90.0 pixels (specifically for fingerprint images at 500 dpi). These sectors are numbered based on their orientation and distance from the reference minutia point. Test points are considered within each sector in order to determine whether there is a common region between the sector in question and the convex hull associated with the reference minutia point. Should this be the case, the sector is labelled as *valid*. This method subsequently assigns this sector number to each neighbouring minutia point that falls within said sector.

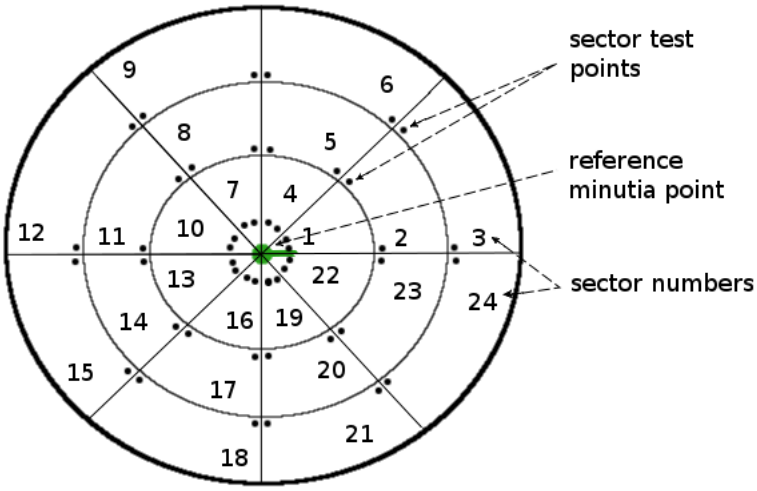


Figure 5.1: A descriptor divided into sectors in order to deal with partial overlapping descriptors.

In addition to this, three features are associated with each of the neighbouring minutia points, i.e. d, β and ϕ, as illustrated in Figure 2.2. These three translation and rotation invariant features accurately capture the spatial and orientation characteristics of each neighbouring minutia point relative to the reference minutia point. In addition to these features, other methods may also use δ, as illustrated in Figure 2.2, but since β, ϕ and δ form a triangle, only two of them are required to capture the necessary information. The final proposed descriptor therefore includes the neighbouring minutia points of the above-mentioned three features, the sector number assigned to each neighbouring minutia point, the numbers of the valid sectors, as well as the orientation of the reference minutia point.

5.3.2 Comparing two local descriptors

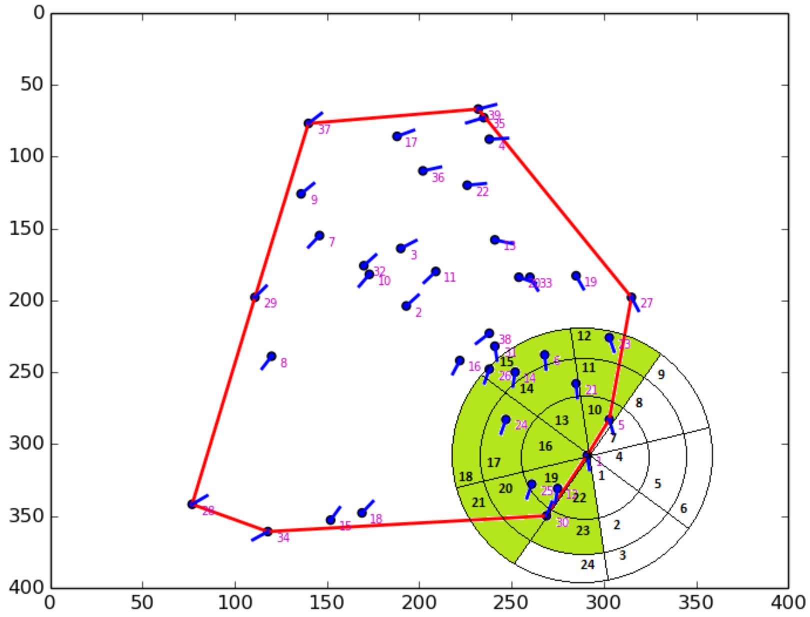
In order to be compared, two descriptors have to satisfy the following two criteria. Firstly, the orientation differences between the two reference minutia points must be less than a predetermined threshold, ω. Secondly, in both descriptors at least one neighbouring minutia point must be matchable. All the neighbouring minutia points that fall within sectors that are valid in both descriptors are deemed *matchable* neighbouring minutia points as indicated in Figures 5.2 (a) and 5.2 (b). If these two conditions are met, the two descriptors are compared, otherwise the local similarity is assigned a value of zero. The proposed approach calculates a neighbouring similarity, $S_{nm}\left(M_{i_k}, M_{i'_{k'}}\right)$, between each pair of matchable neighbouring minutia points as follows,

$$S_{nm}\left(M_{i_k}, M_{i'_{k'}}\right) = \begin{cases} \rho(i,i'), & \text{if sectors are matchable} \\ 0, & \text{otherwise} \end{cases},$$

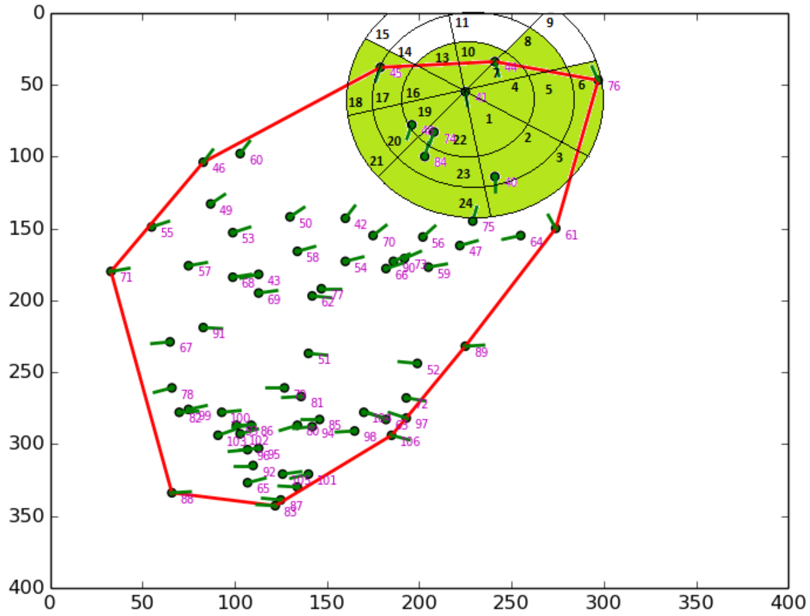
where M_{i_k} and $M_{i'_{k'}}$ represent the *neighbouring* minutia points i and i' in descriptors of the *references* minutia points M_k and $M_{k'}$ within sets T and Q respectively. Furthermore, $\rho(i,i')$ denotes the geometric mean of the three normalized feature differences between the two neighbouring minutia points associated with the two descriptors,

$$\begin{aligned} \rho(i,i') &= \sqrt[3]{\Pi_{s=1}^{3} \kappa(D_s, \epsilon_s^{\rho}, \tau_s^{\rho})} \\ D_1 &= \mid d_{ki} - d_{k'i'} \mid, \\ D_2 &= \mid d\Theta\left(\delta_{ki}, \delta_{k'i'}\right) \mid, \\ D_3 &= \mid d\Theta\left(\phi_{ki}, \phi_{k'i'}\right) \mid, \end{aligned}$$

where d, δ and ϕ are the features illustrated in Figure 2.2, ϵ and τ represent normalization constants for the symmetrical sigmoid normalization function κ, and $d\Theta$ denotes the angle difference function as defined in Equation 2.17. This results in a *similarity matrix* that contains the neighbouring minutiae.



(a) FVC2002 Db1_a: Individual 69, impression 2.



(b) FVC2002 Db1_a: Individual 69, impression 5.

Figure 5.2: Example of the comparison of two partial overlapping sets using the descriptor explained in Figure 5.1.

The neighbouring minutia pairs between descriptors that are required to calculate the *final* local similarity through different strategies may be identified in two ways: The *first* method involves the *assignment-based approach* where the LGS assignment algorithm is applied to the neighbouring minutia similarity matrix, which results in a unique *list* containing the maximum number of neighbouring minutia pairs between the descriptors. The *second* method involves a *threshold-based approach* where thresholds are applied to the three feature differences in order to determine which neighbouring minutia points between two descriptors are paired.

In order to investigate the protocol outlined above, the local similarity is calculated using the following *six* different strategies during stage 1:

1. The *first* strategy averages the similarity of the neighbouring minutia points across *all* the neighbouring minutia pairs within the list obtained through the *assignment*-based approach.

2. The *second* strategy averages the similarity of the neighbouring minutia points of *75%* of the neighbouring minutia pairs associated with the *highest* similarity within the list obtained through the *assignment*-based approach.

3. The *third* strategy averages the similarity of the neighbouring minutia points of the *three most similar* neighbouring minutia points within the list obtained through the *assignment*-based approach.

4. The *fourth* strategy calculates the percentage of paired neighbouring minutia points by employing a *threshold*-based approach.

5. The *fifth* strategy *combines* the percentage of paired neighbouring minutia points with the neighbouring similarity between these pairs.

6. The *sixth* strategy uses the MCC-based local similarity, as outlined in Chapter 2.

The assignment-based and threshold-based (or tolerance box-based) approaches are briefly explained in Section 3.5. In this scenario however, these approaches are only implemented on the neighbouring minutiae between descriptors.

5.3.2.1 Advantages and disadvantages

The proposed local similarity algorithm has a few advantages that distinguish it from other algorithms, since it proposes to combine the *strengths* of several different algorithms. Firstly, it is robust with respect to partial overlap, as is also the case for Jain *et al.* (2008) (without alignment), since it constitutes an adjusted version of the local descriptor employed by Cappelli *et al.* (2010*a*), which determines the common region with valid sectors in both descriptors. However, the sector-based approach proposed by Cappelli *et al.* (2010*a*) is sensitive to noise. Our approach therefore only considers a *valid* sector in order to determine which neighbouring minutia points are matchable, but still compares individual neighbouring minutia points in a similar way to Jain *et al.* (2008) and Jiang and Yau (2000). In addition to this, the proposed approach assigns the neighbouring minutia points being paired either through thresholds (as proposed by Jain *et al.* (2008)) or by using an assignment method (as proposed Fu *et al.* (2013)), which also makes it more robust with respect to noise than is the case for Jiang and Yau (2000). As opposed to Jiang and Yau (2000), the assignment is not only performed for the closest or best pairs, but for *all* the possible neighbouring minutia pairs which increases the distinctiveness of the proposed approach. Finally, although our approach is very similar to the approach followed by Fu *et al.* (2013), it incorporates a *different* strategy for computing the similarity of neighbouring minutia points. This is achieved by assigning neighbouring minutia pairs and allowing for different strategies for computing the local similarity, which is not the case for Fu *et al.* (2013) and Cappelli *et al.* (2010*a*). The proposed algorithm is therefore, to a certain extent, robust with respect to partial overlap and high noise levels, as well as variations in translation and rotation, but still highly distinctive, while also allowing for different strategies for computing the local similarity.

On the other hand, this way of computing the local similarity may have a few drawbacks. The effectiveness with which it deals with partial overlap depends on the size of the sectors, and one of the relatively large sectors we propose here may contain minutia points that are only present in one of the sets due to partial overlap, as is the case for sector 23 in Figure 5.2. This problem may be solved to a certain extent by increasing the number of sectors, but may have the following adverse effect: The utilisation of sectors may lead to border errors and when the number of sectors is increased, the number of border errors may increase as well. Border errors are matching minutia points from two compared descriptors that fall in different sectors and these are usually the result of nonlinear distortion or minutia extraction position errors. The impact of nonlinear distortion is usually small, since we only consider local regions, but minutia extraction position errors may still be problematic.

5.4 Stage 2: Local similarity combination

Section 5.3 elaborated on the different strategies proposed for computing the local similarity between two minutia points that form a pair. However, in order to calculate the *final* local similarity score, these similarities have to be *averaged* for a *fixed* number of minutia pairs. We explained in Section 5.3 that, during preprocessing, two different methods are utilized to determine the paired minutia points. This section provides a short description of how the list of five different pairs of minutia points (as obtained through different stage 2 strategies) is constructed from two lists containing uniquely paired minutia points.

Strategy 1 of stage 2 uses the complete list of paired minutia points as identified with the *tolerance box-based approach*, whereas strategies 2-5 of stage 2 use the list of paired minutia points as identified with the *assignment-based approach*. Strategy 2 uses *all* the paired minutia points in the overlapping region, strategy 3 uses the *most similar* pairs in the overlapping region as obtained through the formula proposed by Cappelli *et al.* (2010*a*), strategy 4 uses *75%* of the *most similar* pairs in the overlapping region, and strategy 5 uses the *three most similar pairs* in the overlapping region.

Although strategy 1 of stage 2 may be more robust with respect to nonlinear distortion, it is more sensitive to noise than the other strategies, since it is captures the best pairs based on spatial position and orientation. Strategy 2 (that incorporates all the minutia points in the overlapping region) may be sensitive to noise, but best captures the dissimilarity for impostor comparisons. Furthermore, strategy 5 should be more robust with respect to noise, but not very distinctive. Strategies 3 and 4 aim to find a *balance* between robustness with respect to noise and capturing the dissimilarity for impostor comparisons.

5.5 Results

The results in Tables 5.2 and 5.3 indicate that *no single* combination of stage 1 and stage 2 strategies outperforms *all* the other combinations for *all* six databases, however the results do point towards individual stage 1 and stage 2 strategies that *generally* perform better across all six databases. We therefore start out by first considering the EER of each stage 1 strategy for *all* the stage 2 strategies, and then proceed to consider the EER of each stage 2 strategy for *all* the stage 1 strategies. We finally consider the area under the ROC curve (AUC) for the *combination* of these strategies.

The five different stage 2 strategies across all six databases result in a total of 30 scenarios for which each of the different stage 1 strategies are compared. For 21 of these 30 scenarios strategy 1 of stage 1 has the lowest EER. The other stage 1 strategies that produce the lowest EERs for specific databases are strategies 2, 4 and 6. Strategies 4 and 6 do however not perform consistently and may produce high EERs in certain scenarios, specifically when strategy 6 is implemented on the FVC2004 DB2_A database. Therefore, across all six databases, strategy 1 of stage 1 *generally* produces the lowest EERs.

When considering the EERs for the different stage 2 strategies, strategies 3 and 4 produce the lowest EERs for most of the local similarities. However, there are scenarios for which strategies 1,2 and 5 have the lowest EER. When strategy 1 of stage 1 is considered, the results indicate that in two of the databases, strategy 1, 3 and 4 of stage 2 produce the lowest EER. On the other

Table 5.2: The *EER* for the six different databases. The best three performances for each database are in boldface, while the best strategy followed during stage 1 for each stage 2 strategy is boxed.

Stage 2 strategy:	**Stage 1 strategy:** 1	2	3	4	5	6
	FVC2002 DB1_A					
1	0.8380	0.9043	2.2165	1.1694	1.4267	$\boxed{0.7515}$
2	$\boxed{0.5707}$	0.6369	1.1515	0.6571	1.2840	0.6830
3	$\boxed{\mathbf{0.4763}}$	0.5987	1.4368	0.7234	0.9021	0.7796
4	$\boxed{\mathbf{0.4662}}$	$\boxed{\mathbf{0.4662}}$	1.0088	0.5426	0.8200	0.7032
5	0.7133	0.8279	2.2367	1.2278	2.1366	$\boxed{0.6953}$
	FVC2002 DB2_A					
1	0.7334	$\boxed{0.7233}$	2.6620	1.2298	1.9150	1.0165
2	$\boxed{\mathbf{0.4279}}$	0.5806	1.4747	0.7615	2.5620	0.9243
3	$\boxed{\mathbf{0.3897}}$	0.5043	1.3984	0.8177	1.9690	0.5908
4	$\boxed{\mathbf{0.3515}}$	0.4481	1.5511	0.7233	2.3868	0.7817
5	0.9322	1.1995	2.8608	1.8241	3.8098	$\boxed{0.6851}$
	FVC2002 DB3_A					
1	$\boxed{\mathbf{1.5807}}$	1.5897	4.8819	2.4627	3.5101	3.1163
2	$\boxed{1.8827}$	2.0450	4.1922	1.9834	3.0007	3.3989
3	$\boxed{\mathbf{1.4079}}$	1.6107	3.0998	1.8947	3.1284	2.3500
4	$\boxed{1.6002}$	1.6919	3.5972	1.7415	2.6175	2.9075
5	$\boxed{\mathbf{1.5296}}$	1.6919	4.1922	2.2373	4.1682	2.6400
	FVC2004 DB1_A					
1	$\boxed{4.6436}$	4.6725	8.2436	6.4674	7.2562	4.8378
2	$\boxed{4.7028}$	4.7418	5.8236	5.2989	6.1679	6.0841
3	$\boxed{\mathbf{3.4224}}$	3.4823	4.5231	4.8709	5.4945	3.5704
4	3.4628	$\boxed{\mathbf{3.4332}}$	4.6819	4.4451	5.0356	4.4364
5	4.0777	4.1766	7.0333	6.2401	7.3147	$\boxed{\mathbf{3.3258}}$
	FVC2004 DB2_A					
1	$\boxed{\mathbf{3.3951}}$	**3.5261**	7.0118	4.1305	4.8865	5.8532
2	5.4805	5.6718	7.0623	$\boxed{4.9163}$	6.1154	7.4546
3	$\boxed{\mathbf{3.4656}}$	3.5663	5.2892	3.7577	5.0070	4.4730
4	$\boxed{3.9391}$	4.0095	6.2462	$\boxed{3.9391}$	4.9871	6.2463
5	4.0196	$\boxed{4.0097}$	6.1455	4.7954	6.8204	4.6643
	FVC2004 DB3_A					
1	$\boxed{\mathbf{3.1855}}$	3.3263	6.6643	4.8061	5.4682	5.3401
2	4.9647	5.1433	8.0418	$\boxed{4.9352}$	5.9184	6.6681
3	$\boxed{\mathbf{3.2617}}$	**3.3146**	5.5594	3.4787	4.3750	4.3750
4	$\boxed{3.6222}$	3.6428	6.5118	3.6573	4.1268	4.9413
5	$\boxed{3.6222}$	3.7747	6.1747	4.7505	6.1107	4.1702

Table 5.3: The *AUC* for the six different databases. The best three performances for each database are in boldface, while the best strategy followed during stage 1 for each stage 2 strategy is boxed.

Stage 2 strategy:	**Stage 1 strategy:** 1	2	3	4	5	6
	FVC2002 DB1_A					
1	0.9994	0.9991	0.9966	0.9986	0.9973	$\boxed{0.9995}$
2	$\boxed{0.9997}$	0.9997	0.9991	0.9995	0.9978	0.9989
3	$\boxed{\mathbf{0.9998}}$	0.9996	0.9985	0.9997	0.9979	0.9984
4	$\boxed{\mathbf{0.9998}}$	**0.9997**	0.9992	0.9995	0.9986	0.9993
5	$\boxed{0.9996}$	0.9993	0.9970	0.9992	0.9961	0.9992
	FVC2002 DB2_A					
1	$\boxed{0.9995}$	0.9992	0.9965	0.9989	0.9967	0.9993
2	$\boxed{0.9997}$	0.9997	0.9989	0.9992	0.9954	0.9991
3	$\boxed{\mathbf{0.9998}}$	0.9998	0.9991	0.9991	0.9933	0.9991
4	$\boxed{\mathbf{0.9998}}$	**0.9998**	0.9989	0.9994	0.9961	0.9995
5	$\boxed{0.9992}$	0.9991	0.9968	0.9977	0.9886	0.9990
	FVC2002 DB3_A					
1	$\boxed{0.9983}$	0.9978	0.9898	0.9962	0.9916	0.9939
2	$\boxed{0.9977}$	0.9976	0.9914	0.9977	0.9931	0.9903
3	$\boxed{\mathbf{0.9989}}$	**0.9988**	0.9941	0.9974	0.9929	0.9959
4	$\boxed{0.9985}$	0.9985	0.9941	0.9981	0.9948	0.9946
5	$\boxed{\mathbf{0.9985}}$	0.9985	0.9916	0.9967	0.9903	0.9946
	FVC2004 DB1_A					
1	$\boxed{0.9924}$	0.9915	0.9759	0.9843	0.9803	0.9913
2	$\boxed{0.9926}$	0.9921	0.9876	0.9889	0.9830	0.9823
3	$\boxed{\mathbf{0.9953}}$	0.9950	0.9919	0.9902	0.9862	0.9931
4	$\boxed{\mathbf{0.9957}}$	**0.9955**	0.9914	0.9919	0.9880	0.9893
5	0.993402	0.9931	0.9837	0.9850	0.9782	$\boxed{0.9940}$
	FVC2004 DB2_A					
1	$\boxed{\mathbf{0.9947}}$	0.9941	0.9821	0.9919	0.9877	0.9877
2	0.9900	0.9892	0.9817	$\boxed{0.9906}$	0.9828	0.9759
3	$\boxed{\mathbf{0.9956}}$	**0.9953**	0.9909	0.9937	0.9869	0.9892
4	$\boxed{0.9938}$	0.9935	0.9870	0.9927	0.9873	0.9833
5	$\boxed{0.9933}$	0.9930	0.9860	0.9899	0.9791	0.9875
	FVC2004 DB3_A					
1	$\boxed{\mathbf{0.9958}}$	**0.9953**	0.9818	0.9910	0.9863	0.9896
2	$\boxed{0.9915}$	0.9911	0.9773	0.9909	0.9822	0.9815
3	$\boxed{\mathbf{0.9953}}$	0.9952	0.9884	0.9946	0.9887	0.9920
4	$\boxed{0.9944}$	0.9943	0.9847	0.9939	0.9896	0.9892
5	$\boxed{0.9940}$	0.9938	0.9859	0.9901	0.9807	0.9917

hand, strategy 2 of stage 2 invariably produces high EERs when implemented on the FVC2002 and FVC2004 fingerprint databases.

The AUC-based results are similar to those for the EERs. All the algorithms with the lowest AUC involve strategy 1 of stage 1, except for a single case that involves strategy 6. Strategy 4 of stage 1 produced the largest AUC in four of the six databases, while strategy 1 and 4 of stage 2 both produced the largest AUC in one of the six databases. The AUC for strategies 3 and 4 of stage 2 do however differ slightly. Furthermore, smaller EERs and larger AUCs are generally observed for the FVC2002 fingerprint databases than is the case for the FVC2004 databases.

In summary, these results do not point towards a *single* algorithm that performs *best* for all the EER-based and AUC-based performance metrics when estimated by considering the databases in question. The algorithm that produces the lowest EERs across *all* the databases *combines* strategy 1 of stage 1 with strategy 3 of stage 2, however, all the different variations of stage 1 and stage 2 produce much higher error rates when implemented on the FVC2004 fingerprint databases than is the case for the FVC2002 fingerprint databases.

5.5.1 Statistical significance testing

A better EER is reported when *strategy 1* of stage 1 is implemented instead of *strategy 6* of stage 1. A statistical significance test should therefore be conducted to investigate the *significance* of this improvement. In this study a dependent (paired) t-statistic for a 0.05 level of significance, as outlined by Dowdy *et al.* (2011), is applied. This test determines the likelihood that local parameters of two dependent, normally distributed data samples A and B are representative of two populations with unequal means. The following steps are followed:

1. In our case samples A and B are the sets of EER-based performance measures obtained from two different local similarity score calculation methods. The expected improvement from using similarity score calculation method A, instead of B, is as follows,

$$\mu_B^A = \mu(A) - \mu(B), \tag{5.1}$$

   where μ_B^A represents the difference between the sample means. Furthermore, let μ_A and μ_B denote the population means associated with samples A and B respectively.

2. (Hypothesis statement) The two hypotheses tested are:

   - (Null Hypothesis, H_0) $\mu_A - \mu_B \leq 0$;
   - (Alternative Hypothesis, H_A) $\mu_A - \mu_B > 0$.

As we are only interested in whether *strategy 1* of stage 1 has a smaller EER than *strategy 6* of stage 1, a right-tailed is used.

3. The formula for the test statistic t_{stat} is as follows,

$$t_{\text{stat}} = \frac{\mu_B^A - \mu_0}{s_d s \sqrt{n}}, \tag{5.2}$$

where μ_B^A and s_d represent the mean and standard deviation of the differences between the paired samples respectively, while n is the number of samples.

4. (Decision) At an $\alpha = 0.05$ level of significance the test statistic must exceed the associated critical value t_{crit} for the null hypothesis to be rejected, i.e.

$$\begin{aligned} t_{\text{stat}} &> t_{\text{crit}} && (5.3)\\ &= t_{(\alpha,n-1)} && (5.4)\\ &= t_{(0.05,2)} && (5.5)\\ &= 2.920. && (5.6) \end{aligned}$$

Consequently, if $t_{\text{stat}} > 2.92$, we can confirm that method A results in a higher performing local similarity score calculation method than is the case for method B.

This statistical significance test can be conducted for any two stage 1 strategies, while keeping the preprocessing and stage 2 strategy the same. However, we are only interested in the t_{stat} between *strategy 1* of stage 1 and *strategy 6* of stage 1. The t_{stat} values are listed in Table 5.4. For the FVC2002 fingerprint databases $t_{\text{stat}} < t_{\text{crit}}$ and therefore, at a significance level of 0.05, we cannot reject the null hypothesis and subsequently conclude that, based on statistical evidence, there is *not a significant difference* between the proficiency of these two methods. However, for the FVC2004 fingerprint databases $t_{\text{stat}} > t_{\text{crit}}$, which enables us to reject the null hypothesis and conclude that *strategy 1* of stage 1 is *significantly more proficient* than *strategy 6* of stage 1 with a confidence of 95%.

Table 5.4: The t_{stat}-values between *strategy 1* of stage 1 and *strategy 6* of stage 1 when evaluated on the FVC2002 and FVC2004 fingerprint databases.

Fingerprint databases	t_{stat}
FVC2002	2.546854
FVC2004	3.030137

5.6 Discussion and conclusion

In this chapter we compared different *local* similarity calculation methods (stage 1 strategies) and investigated how many paired minutia points should be used to calculate the final local similarity score (stage 2 strategies). The results did *not* point towards a combination of these two strategies that are superior to *all* the others across *all* six of the databases considered. However, when strategy 1 of stage 1 was implemented in conjunction with strategies 3 or 4 of stage 2, the lowest EERs were obtained across all the databases considered.

The main conclusion is therefore that the proposed strategy 1 of stage 1 performs significantly better than the MCC-based local similarity on the FVC2004 fingerprint databases (i.e. at a significance level of 0.05). Although it also produces lower EERs when implemented on the FVC2002 fingerprint databases, there is not sufficient evidence to conclude that this strategy *generally* performs significantly better. The above-mentioned improvement may however point towards the fact that the proposed algorithm is more robust with respect to noise as is the case for the MCC-based strategy, since the FVC2004 fingerprint databases contain more low quality fingerprints than is the case for the FVC2002 fingerprint databases (Maltoni *et al.*, 2009), which may result in comparisons with higher noise levels.

These results do not provide clear guidelines on how to calculate the local similarity score, but rather point towards how a similarity score should *not* be calculated: Firstly, using the similarity between the three best neighbouring minutia points is not distinctive enough for calculating the similarity between descriptors and can therefore not compete with methods that consider *all* the neighbouring minutia points within a 90-pixel radius. It is interesting to note that the literature states that fixed length local descriptors are not accurate because of their sensitivity to noise (Liu and Mago, 2012; Feng, 2008). However, these results demonstrate that even if the pairs are assigned in such a way that they are less sensitive to noise, the use of three neighbouring minutia points is still not distinctive enough to compare favourably with other methods. This study therefore suggests that it is better to consider *all* the neighbouring minutia points within a fixed radius than only a certain percentage of paired minutia pairs.

Secondly, what the local similarity score calculation is concerned, this study suggests that it is better to calculate the *average* of the neighbouring similarity between *all* the pairs, than to use a *percentage* of paired neighbouring minutia points or the MCC-based similarity. Although the performance of the proposed method is only marginally better, cases do exist where the other two methods perform significantly worse. The proposed approach is therefore reliable in various scenarios.

When considering how to choose the paired minutia points and how many pairs to use, this study suggests the following. The use of all the minutia points (strategy 2 of stage 2) is highly sensitive to noise, which results in strategies 3 and 4 of stage 2 performing better on comparisons with low noise levels. It is therefore, in the first place not necessary to include all the paired minutia points. Secondly, the identification of minutia points through a tolerance box-based approach performs well on most databases, but produces poor results on DB1_A of FVC2004. The second suggestion is therefore to rather use a percentage of the minutia points within the overlapping region than to incorporate the tolerance box-based approach, specifically for the purpose of local similarity score calculation.

In summary, this chapter proposed a *new* local similarity algorithm based on descriptors that is robust to partial overlap, but still compares individual minutia points in order to be robust with respect to noise. This method performs best when the local similarity is based on the average similarity between *all* the neighbouring pairs. Even though the proposed local similarity method often performs only marginally better than other local similarity methods, it appears to be more stable across *all* databases. Furthermore, when averaging a percentage (around 50%) of the pairs in the overlap, the local similarity proposed by Cappelli *et al.* (2010*a*) produces the best results.

Chapter 6

Combining different similarity score calculation methods

6.1 Introduction

The results in Chapter 3 indicate that the similarity score calculation method based on the Minutia Cylinder Code (MCC) Local Greedy Similarity with Distortion Tolerant Relaxation (LGS_DTR) as proposed by Cappelli *et al.* (2010*b*), i.e. S_3, achieves the highest performance for most of the databases considered. However, the performance of this method decreases for databases containing many comparisons with high levels of intra-class variations and/or inter-class similarities, as is the case for the FVC2004 fingerprint databases. Furthermore, this method does not perform the best for all the databases considered. This indicates that said similarity score calculation method is more sensitive to specific intra-class variations or inter-class similarities than is the case for some of the other similarity score calculation methods. The *combination* of different similarity score calculation methods may therefore improve the performance of existing similarity score calculation methods and is the focus of the current chapter.

The research questions posed in this chapter are as follows:

- Firstly, does the *combination* of the three types of similarity score calculation methods (as introduced in Section 2.2) improve the accuracy of the individual similarity score calculation methods?
- Secondly, does this *improved* method also address intra-class variations and inter-class similarities and therefore produce a similar performance when implemented on the FVC2004 and FVC2002 fingerprint databases?
- Finally, is the performance of this fused similarity score calculation method better than that of the *most proficient* existing similarity score calculation method, i.e. LGS_DTR?

In this chapter, *four* different similarity score calculation methods are combined into a *new* similarity score calculation method. This study suggests that the combined method may better address the different types of intra-class variations and inter-class similarities than is the case for existing, individual similarity score calculation methods and may therefore be more accurate. Section 6.2 explains the methodology, while Sections 6.3.1 and 6.3.2 elaborate on the individual similarity score calculation methods and the fusion process respectively. Section 6.4 presents the results and Section 6.5 provides a discussion and conclusions.

6.2 Methodology

The methods considered here are similar to those discussed in Chapter 4, but for the fact that the similarity score calculation stage differ. Different similarity score calculation methods are implemented on six databases with the testing protocol and preprocessing procedures as described in Chapter 3. Section 6.3.1 explains the individual similarity score calculation methods (adjusted versions of existing similarity scores), while section 6.3.2 elaborates on how the fusion process that combines the four individual similarity score calculation methods, works. After the calculation of the similarity score values, the next step is to calculate the EER, FMR_{zero}, $\text{FNMR}_{\text{zero}}$, FMR_{100}, FMR_{1000}, FMR_{10000}, and area under the Receiver Operating Characteristic Curves (AUC) as performance measures for each similarity score calculation method on the different databases considered. These performances are then compared to those of existing similarity score calculation methods as reported in Chapter 4. Finally, should the proposed calculation methods produce better results than existing similarity score calculation methods, the *significance* of this improvement is determined. In order to verify this, we implement the dependent (paired) t-statistic at a significance level of 0.05 on the EER.

6.3 Proposed similarity score calculation method

As previously mentioned, four different similarity score calculation methods are combined in this chapter. The first three methods focus on the similarity in the overlapping region, while the fourth takes the complete sets and the size of the overlapping region into account. Section 6.3.1 elaborates on the individual similarity score calculation methods and explains how these methods may address different types of intra-class variations and inter-class similarities. Subsequently, Section 6.3.2 explains how the different similarity score calculation methods may be *fused* in order to produce a variety of *combined* similarity score calculation methods.

6.3.1 Individual similarity score calculation methods

Similarity score calculation method 1, $\mathbf{S}_{L1}$

The first similarity score calculation method constitutes a *combination* of the *local* and *structural* similarity score calculation methods. It implements the best local similarity algorithm as established in Chapter 5 on the paired minutia points (with parameters as indicated in Table 5.1). Thereafter, a slightly adapted version of the distortion tolerant relaxation (DTR) method as proposed by Cappelli *et al.* (2010*b*) is used to compute the global relaxed similarity between the paired minutia points (with parameters as indicated in Table 6.1). As is the case for the MCC LGS_DTR algorithm, the minutia pairs are ordered based on their efficiency, after which the relaxed similarity of 60% of the most efficient pairs are averaged to produce the final similarity score value.

Table 6.1: The parameters for the DTR algorithm.

Parameter	**Value**
ϵ_p	30.0
τ_p	$\frac{2}{5}$
min_{np}	3
max_{np}	10
ϵ_d^{ρ}	$\frac{1}{30}$
τ_d^{ρ}	-150.0
ϵ_δ^{ρ}	$\frac{\pi}{4}$
τ_δ^{ρ}	-15.0
ϵ_ϕ^{ρ}	$\frac{\pi}{18}$
τ_ϕ^{ρ}	-40.0

This similarity score calculation method is similar to the best performing method, i.e. the MCC LGS_DTR method (S_3), but for a few small differences. Firstly, the proposed method only considers the minutia points in the overlap in order to ensure robustness with respect to partial overlap. Secondly, the *best* local similarity is used instead of the MCC LGS local similarity, which increases robustness with respect to noise. Finally, the features are combined using the *geometric mean* instead of multiplication during relaxation. This results in a better range of similarity score values that can be more easily combined during the fusion stage.

However, the fact that only a percentage of the possible minutia pairs in the overlap is considered (to increase robustness with respect to noise) may have certain disadvantages. Firstly, this approach may be sensitive to extreme cases of local and global inter-class similarities. Furthermore, nonlinear distortion may still affect this similarity score calculation method, in which case other similarity score calculation methods may be required.

Similarity score calculation method 2, S_{L2}

The second method is a *local* similarity score calculation method. Similar to the first method, it calculates the local similarity with the most accurate method established in Chapter 5 for the list of paired minutia points from the overlap using the assignment approach (with parameters as indicated in Table 5.1). It subsequently averages said local similarity for 90% of the most similar pairs as identified by the point similarity algorithm in order to obtain the local similarity score value. This method (S_{L2}) is more robust to nonlinear distortion than is the case for the first method (S_{L1}), since it only considers local regions, provides the similarity between different pairs, and does not rearrange the list according to efficiency. This method is however not as distinctive as the first method, since it only considers local regions within the overlap and is therefore still sensitive to local inter-class similarity.

Note that both of these first two similarity score calculation methods therefore incorporate only a percentage of the possible paired minutia points within the overlap. The second approach (S_{L2}) furthermore determines the paired minutia points using the local similarity, after which the similarity score value is calculated. This implies that it uses the most similar local minutia pairs and therefore does not capture the true dissimilarity. Global inter-class similarity may therefore still adversely affect the proficiency of these approaches.

Similarity score calculation method 3, S_{L3}

This method is similar to the similarity score calculation method proposed by Fu *et al.* (2013). It identifies the number of paired minutia points between two sets using a tolerance box-based approach, as well as the number of minutia points in the overlapping region, for each set. Thereafter, it calculates the best proposed local similarity, S_{local_n}, for each pair of minutia points as established in Chapter 5 (with parameters as indicated in Table 5.1). The final local similarity is as follows,

$$S_3 = \frac{N_m \cdot \overline{S_{\text{local}_n}}}{(N_{TO} + N_{QO})/2}, \tag{6.1}$$

with N_m, N_{TO}, and N_{QO} as defined in Chapter 2. $\overline{S_{\text{local}_n}}$ is the *mean* local similarity value for all the identified pairs of minutia points.

This approach therefore constitutes a combination of a "percentage of paired minutia points"-based similarity score calculation method and a local similarity score calculation method. The "percentage of paired minutia points"-based similarity score calculation method considers the entire overlapping region and therefore addresses global inter-class similarity to a certain degree. This method should however be sensitive to noise and extreme levels of nonlinear distortion, while also being robust to extraction position errors, due to the implementation of the tolerance box-based approach for identifying paired minutia points. The local similarity of the pairs is incorporated in order to increase the distinctiveness of the method and may also capture different local similarity values, since it identifies the pairs of the minutia points in a different way. This method should however not replace similarity score calculation method S_{L2}, since it identifies fewer pairs of minutia points for impostor comparisons, and may (in some cases) not adequately capture the dissimilarity. Since this similarity score calculation method (S_{L3}) only considers the overlapping region (as is the case for methods S_{L1} and S_{L2}), it should not be robust to inter-class similarity.

Similarity score calculation method 4, S_{L4}

We finally employed the similarity score calculation method proposed by Khanyile *et al.* (2014). This method deals with local inter-class similarity, since it penalizes comparisons with a low percentage of minutia points in the overlap. This method may be sensitive to noise, nonlinear distortion, partial overlap, and global inter-class similarity, but addresses the main problem that the other three similarity score calculation methods do not address properly, namely local inter-class similarity.

6.3.2 Similarity score fusion

The *four* different *individual* similarity score calculation methods deal with different intra-class variations and inter-class similarities, but may be sensitive to all of the different intra-class variations and inter-class similarities when combined. Furthermore, some of the individual similarity score calculation methods perform significantly worse than the highest performing one. When said similarity score values are combined with a geometric mean, both of these factors may result in a lower performance by the combined method when compared to the best-performing individual method. We therefore implement a fusion technique for each similarity score calculation method that *only* focuses on combining the similarity score values of comparisons that have a *high certainty* of being either a genuine or impostor comparison.

We now explain what is meant by a comparison with a high certainty of being either a genuine or an impostor comparison. When the distribution of similarity score values for genuine and impostor comparisons are plotted on the same graph, *three* different *score ranges* are often observed as illustrated in Figure 6.1. Range 1 mostly contains impostor comparisons, range 2 contains genuine and impostor comparisons, while range 3 mostly contains genuine comparisons. Comparisons within ranges 1 and 3 can therefore be labelled as impostor and genuine comparisons respectively, with a high certainty.

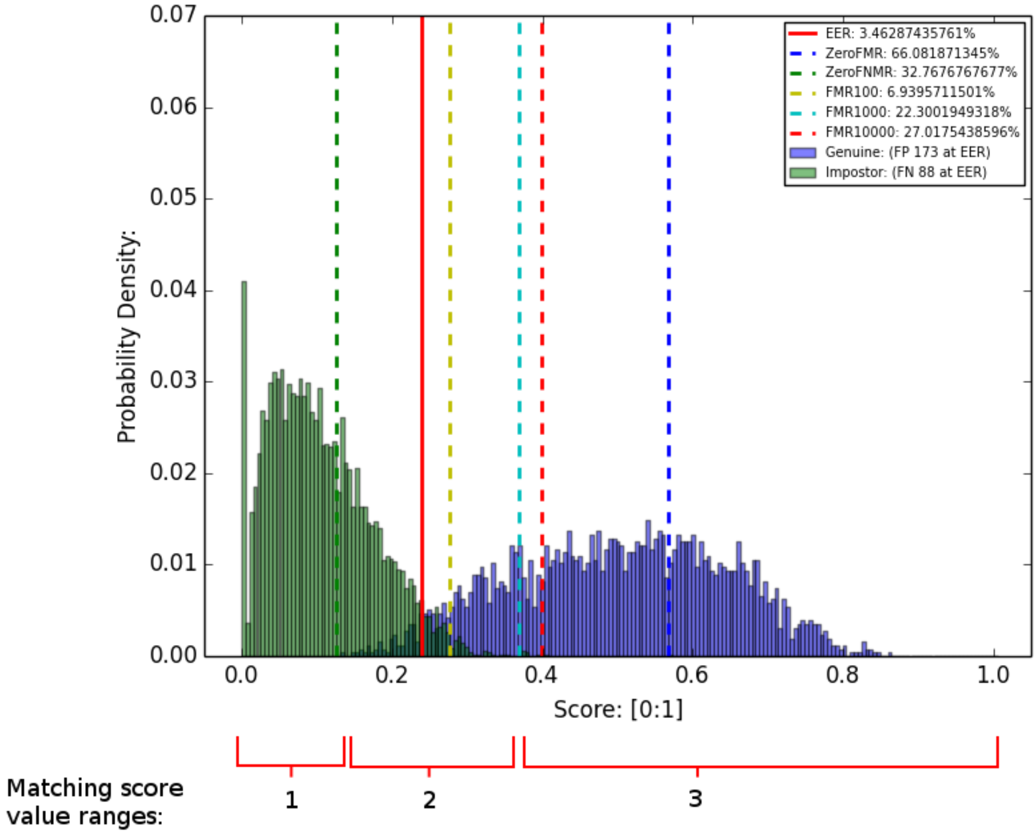


Figure 6.1: Illustration of the three similarity score ranges.

These ranges may however vary between different similarity score calculation methods. The reason for this is that the probability distributions depend on the features and normalization strategy utilised during similarity score calculation. Furthermore, these ranges may vary between different databases. When a database is associated with high levels of intra-class variation (of a specific type), range 2 (the overlapping, uncertain region) will spread out towards the *left*. Similarly, when a database is associated with high levels of inter-class similarity (of a specific type), the overlapping (uncertain) region will spread out towards the *right*. However, for a specific similarity score calculation method these ranges will not change significantly for a specific database as long as the preprocessing strategy is successful.

The fusion method depends on the ranges specified in Table 6.2 and involves *two* stages. The most accurate of the four similarity score calculation methods is first identified and the similarity score value of the comparisons is set to this best value. In this study we use similarity score calculation method S_{L1}. The other three similarity score values are then iteratively compared to each individual similarity score calculation method's certainty ranges, and should the similarity score values fall within their own specified range labelled "1" or "3" it is combined (using a geometric mean) with the current similarity score

Table 6.2: Certainty range limits for different similarity score calculation methods.

Similarity score calculation method	**Upper limit of range 1**	**Lower limit of range 3**
S_{L_1}	0.15	0.25
S_{L_2}	0.15	0.35
S_{L_3}	0.05	0.25
S_{L_4}	0.15	0.35

value between the comparisons. The utilised certainty ranges are specified in Table 6.2. On the other hand, should the similarity score value of a specific secondary calculation method fall within its own range labelled "2", then the current similarity score value is retained for the similarity score calculation method in question.

We now explain how this fusion approach may improve the performance of a similarity score calculation method. The idea is that, since these different individual similarity score calculation methods capture different similarities and are sensitive to different types of intra-class variations and inter-class similarities, the same comparisons may not necessarily fall within the same range for different similarity score calculation methods. It is therefore possible that the best similarity score calculation method may be uncertain about the legitimacy of a specific comparison, while one of the other similarity score calculation methods may be more certain about the outcome. In this case said fusion strategy may improve the certainty by increasing the similarity score of genuine comparisons or decreasing the similarity score for impostor comparisons, and therefore improve its accuracy. On the other hand, if both similarity score calculation methods have a high certainty that a comparison is either a genuine or impostor comparison, the fusion strategy will not decrease the accuracy. This approach is however only effective when the similarity score calculation methods being fused have similar ranges and even then it may still not produce optimal fusion results.

Density fusion approaches are based on the same principle, except for the fact that they consider all the different score values and not only the three specified ranges. This study is however limited to methods that do not require intensive training in order to determine the specific class-conditional density function for each of the different similarity scores. We therefore opted to obtain the similarity score values that have similar overlapping regions (through normalization techniques), and then use fixed thresholds for these regions in order to perform the fusion. This fusion strategy may not lead to a significant improvement in performance, but may still reduce the similarity score values for those inter-class similarities that specific similarity score calculation methods are sensitive to. These similarities include the local inter-class similarity, for which lower false match rates (FMRs) may be produced, which may in turn lead to an increase in accuracy.

6.4 Results and findings

The results are presented in Tables 6.3, 6.4, and 6.5. The main observations are as follows:

1. For five of the six databases the fused similarity score calculation method, S_C, has a lower EER than the seven *existing* similarity score calculation methods (S_1-S_7). The EER for S_C is marginally better than the EER for S_3, i.e. the method proposed by Cappelli *et al.* (2010*b*), when implemented on the above-mentioned five databases.

2. The $FNMR_{zero}$ for the proposed algorithm (S_C) is considerably lower than that for the *existing* seven similarity score calculation methods for all six databases.

3. S_C produces lower EERs than the individual *improved* similarity scores calculation methods (S_{L1}- S_{L4}) on five of the six databases.

4. S_C generally produces lower error rates (EER, FMR, and FNMR) for the FVC2002 fingerprint databases than is the case for the FVC2004 fingerprint databases.

5. S_{L1} and S_{L3} produce lower EERs than most of the *existing* similarity score calculation methods (S_1-S_7) when implemented on both the FVC2002 and FVC2004 fingerprint databases, while S_{L2} produces lower EERs than most of the *existing* similarity score calculation methods (S_1 - S_7) when implemented on the FVC2002 fingerprint databases.

6. The AUC indicates a similar level of performance by the different similarity score calculation methods across all the databases.

6.4.1 Statistical significance testing

The EER for S_C is better than that for the best *existing* similarity score (S_3) and also better than that for the best *improved* similarity score when implemented on a specific database. A statistical significance test is therefore conducted to investigate the significance of this improvement in proficiency. The same statistical significance test and protocol as explained in Section 5.5.1 are again implemented here in order to determine the significance of abovementioned improvements in proficiency, first for S_C and best *improved* similarity score (on each database), and then for S_C and S_3.

Table 6.3: The performance measures on the *FVC2002* fingerprint databases.

Score methods	**EER**	**FNMR$_{zero}$**	**FMR$_{zero}$**	**FMR$_{100}$**	**FMR$_{1000}$**	**FMR$_{10000}$**
		FVC2002	**DB1_A**			
S_C	0.18	0.93	**0.68**	**0.0**	**0.25**	**0.50**
S_{L_1}	**0.17**	**0.84**	7.05	**0.0**	0.32	0.76
S_{L_2}	0.36	2.67	10.33	0.14	0.97	10.22
S_{L_3}	0.30	1.84	18.89	0.07	1.58	3.56
S_1	0.47	9.88	3.99	0.36	1.30	2.95
S_2	0.73	65.96	2.20	0.65	1.44	2.09
S_3	0.47	38.26	1.76	0.40	0.86	1.12
S_4	2.23	41.70	13.57	3.09	8.71	10.80
S_5	0.43	7.52	9.86	0.32	2.66	9.46
S_6/S_{L_4}	1.92	41.27	12.49	2.91	7.30	9.32
S_7	0.76	14.95	52.21	0.61	13.06	40.55
		FVC2002	**DB2_A**			
S_C	**0.14**	**1.88**	**0.47**	**0.04**	**0.18**	**0.32**
S_{L_1}	0.22	2.63	7.19	**0.04**	0.50	3.02
S_{L_2}	0.41	26.42	5.58	0.18	1.12	4.17
S_{L_3}	0.36	2.79	3.67	0.11	0.72	1.51
S_1	0.64	17.64	2.66	0.50	1.37	1.69
S_2	0.74	76.55	10.61	0.68	2.01	4.71
S_3	0.36	11.11	1.12	0.18	0.68	0.94
S_4	1.64	33.43	12.45	2.01	4.64	6.22
S_5	0.53	16.85	3.78	0.32	1.73	2.88
S_6/S_{L_4}	1.39	32.73	11.22	1.51	3.96	5.40
S_7	0.83	22.73	13.78	0.68	3.52	5.40
		FVC2002	**DB3_A**			
S_C	**0.65**	14.71	47.62	0.59	**3.00**	16.48
S_{L_1}	0.96	13.76	86.13	0.94	7.23	43.44
S_{L_2}	1.45	24.36	31.99	1.80	5.59	11.45
S_{L_3}	0.76	**7.32**	64.38	**0.35**	6.05	13.36
S_1	1.45	25.08	13.32	2.19	7.81	12.66
S_2	2.63	96.76	13.71	3.83	6.99	11.52
S_3	1.36	38.21	**6.52**	1.48	3.87	**5.82**
S_4	4.62	60.32	24.26	8.79	18.32	22.23
S_5	1.61	16.16	29.84	2.54	16.67	20.11
S_6/S_{L_4}	4.12	60.00	21.41	8.20	16.33	21.33
S_7	2.15	21.52	88.63	6.05	49.92	73.24

Table 6.4: The performance measures on the *FVC2004* fingerprint databases.

Score methods	**EER**	**FNMR$_{zero}$**	**FMR$_{zero}$**	**FMR$_{100}$**	**FMR$_{1000}$**	**FMR$_{10000}$**
		FVC2004	**DB1_A**			
S_C	**1.85**	**17.90**	18.56	3.20	**7.45**	**8.38**
S_{L_1}	2.18	19.78	60.51	4.05	14.93	53.22
S_{L_2}	3.19	42.16	53.84	5.65	16.96	27.06
S_{L_3}	2.87	20.34	30.99	6.78	18.40	29.47
S_1	3.15	26.48	28.15	8.58	21.63	27.37
S_2	4.27	99.09	30.60	7.56	19.14	27.29
S_3	2.09	77.25	**14.78**	**2.77**	9.36	13.29
S_4	3.75	34.16	28.65	9.04	21.05	26.12
S_5	3.74	29.03	34.39	11.23	29.51	32.83
S_6/S_{L_4}	3.64	30.63	30.25	8.62	19.49	27.64
S_7	4.68	32.10	50.57	17.00	37.43	48.11
		FVC2004	**DB2_A**			
S_C	**1.39**	**21.58**	24.24	2.21	**7.36**	**15.72**
S_{L_1}	1.67	46.12	98.55	3.14	15.59	32.76
S_{L_2}	3.41	28.89	67.00	6.63	20.74	29.62
S_{L_3}	2.22	22.38	69.94	5.95	23.51	36.41
S_1	2.88	37.68	33.28	7.32	16.96	25.48
S_2	5.20	98.38	48.27	10.77	19.53	36.97
S_3	1.60	42.26	**19.45**	**2.17**	7.44	15.76
S_4	3.95	36.85	43.37	8.56	28.22	32.48
S_5	3.69	49.74	45.46	11.62	26.45	44.94
S_6/S_{L_4}	3.57	32.48	37.14	8.08	25.96	33.92
S_7	3.98	37.15	88.59	15.64	53.82	83.80
		FVC2004	**DB3_A**			
S_C	1.53	33.64	7.37	2.06	4.94	6.79
S_{L_1}	1.78	51.96	64.69	2.63	10.58	53.79
S_{L_2}	3.31	45.68	25.47	6.50	13.17	18.40
S_{L_3}	1.70	**13.09**	54.81	3.05	11.93	30.21
S_1	2.66	51.26	43.50	5.19	16.13	25.10
S_2	5.25	95.58	29.59	9.96	16.17	21.89
S_3	**1.16**	62.83	**7.04**	**0.86**	**4.20**	**6.38**
S_4	2.69	47.97	22.22	6.71	13.09	21.93
S_5	2.83	49.44	77.33	6.67	22.22	48.50
S_6/S_{L_4}	2.72	47.21	21.85	6.54	13.05	19.05
S_7	3.27	52.64	93.29	8.85	24.98	61.11

Table 6.5: The AUC measure.

Score methods	**DB1_A**	**DB2_A**	**DB3_A**
	FVC2002		
S_C	**0.999990**	**0.999987**	**0.999628**
S_{L_1}	**0.999974**	**0.999954**	**0.999144**
S_{L_2}	0.999902	0.999783	0.999061
S_{L_3}	**0.999914**	**0.999946**	**0.999562**
S_1	0.999808	0.999660	0.998924
S_2	0.998890	0.999098	0.994238
S_3	0.999687	0.999909	0.999050
S_4	0.998890	0.998733	0.992606
S_5	0.999788	0.999733	0.998714
S_6	0.997455	0.998961	0.993375
S_7	0.999289	0.999469	0.996586
	FVC2004		
S_C	**0.998842**	**0.999030**	**0.998139**
S_{L_1}	**0.998210**	**0.997891**	**0.998139**
S_{L_2}	0.995997	0.995404	0.995071
S_{L_3}	0.996938	0.997514	**0.998648**
S_1	0.996170	0.997026	0.997016
S_2	0.990353	0.984771	0.987327
S_3	**0.997855**	**0.998675**	**0.999038**
S_4	0.994859	0.994531	0.996772
S_5	0.994969	0.995332	0.996547
S_6	0.995239	0.995227	0.996825
S_7	0.992271	0.992973	0.995369

The t_{stat}-values between the EERs for S_C and the best *improved* similarity score are listed in Table 6.6. For the *FVC2002* fingerprint databases $t_{\text{stat}} < t_{\text{crit}}$ and therefore, at a significance level of 0.05, we cannot reject the null hypothesis and subsequently conclude that, based on statistical evidence, there is not a *significant* difference between the proficiency of these two methods. However, for the *FVC2004* fingerprint databases $t_{\text{stat}} > t_{\text{crit}}$, which enables us to reject the null hypothesis and conclude that S_C is *significantly* more proficient than the best *improved* similarity score with a confidence of 95%.

The t_{stat} values between the EERs for S_C and S_3 are listed in Table 6.7. For the *FVC2004* fingerprint databases $t_{\text{stat}} < t_{\text{crit}}$ and therefore, at a significance level of 0.05, we cannot reject the null hypothesis and subsequently conclude that, based on statistical evidence, there is not a *significant* difference between the proficiency of these two methods. However, for the *FVC2002* fingerprint

Table 6.6: The t_{stat}-values between S_C and the best *improved* individual similarity score when evaluated on the FVC2002 and FVC2004 fingerprint databases.

Fingerprint databases	t_{stat}
FVC2002	2.038064
FVC2004	6.738195

databases $t_{\text{stat}} > t_{\text{crit}}$, which enables us to reject the null hypothesis and conclude that S_C is *significantly* more proficient than S_3 with a confidence of 95%.

Table 6.7: The t_{stat}-values between S_C and S_3 when evaluated on the FVC2002 and FVC2004 fingerprint databases.

Fingerprint databases	t_{stat}
FVC2002	3.255176
FVC2004	0.164511

6.5 Discussion and conclusion

The first question posed in this chapter was whether combining the different types of similarity score calculation methods with the proposed fusion technique improves the performance of the individual similarity score calculation methods, since the combined method is expected to address more intra-class variations and inter-class similarities. The results indicate that the combination of the different similarity score calculation methods improve the accuracy of the individual methods *significantly* when implemented on the FVC2004 fingerprint databases, but the margin of improvement is not significant when implemented on the FVC2002 fingerprint databases. Although only a marginal improvement is observed for the EER (when implemented on the FVC2002 databases), the FMR and FNMR show a more substantial improvement which hints towards the fact that this approach may better deal with different types of intra-class variations and inter-class similarities.

We do however conclude that these types of similarity score calculation methods do not *adequately* address the problem of intra-class variations and inter-class similarities, which is evident from the fact that a lower performance is reported for the FVC2004 fingerprint databases than is the case for the FVC2002 fingerprint databases. The results clearly indicate that the performance is considerably lower when implemented on the FVC2004 fingerprint databases than is the case for the FVC2002 databases. This study does not provide an answer as to why this is the case, but it may be that these three types of similarity score calculation methods are not sufficiently robust to higher levels of nonlinear distortion and noise, even though they seem to better capture the overall accuracy than is the case for the individual similarity score calculation methods. Furthermore, the limitations of minutia extraction and point matching may also contribute to this.

The third research question was whether the proposed individual and fused similarity score calculation methods perform better than *existing* similarity score calculation methods. The results indicate that there may be an improvement for some of the databases considered, but at a 0.05 level of significance, the results for the FVC2004 fingerprint databases do not provide sufficient evidence to conclude that S_C performs significantly better than S_3. For the FVC2002 fingerprint databases said improvement is however significant at a 0.05 level of significance. We therefore conclude that the answer to this question depends on the intra-class variations and inter-class similarities within the specific database. For the FVC2002 fingerprint databases, S_C, S_{L1}, and S_{L2} perform better than S_3 based on the EER and the AUC. However, for the FVC2004 fingerprint databases, only S_C performs better than S_3 for two of the three databases (if only by a small margin).

One question that remains is why the individual similarity score calculation method S_{L1}, which is an adjusted version of S_3, performs worse than S_3 when implemented on the FVC2004 fingerprint databases. One reason for this may be (as discovered in Chapter 5), that the number of paired minutia points (stage 2) of S_3 is highly accurate in balancing robustness to noise against distinctiveness and addressing local inter-class similarity, whereas S_{L1} is sensitive to local inter-class similarity. It would therefore seem that penalizing the local inter-class similarity (in the same way that S_3 does), is more accurate than fusing S_6 as proposed by Khanyile *et al.* (2014). Therefore, for these types of similarity score calculation methods, there is still no effective way for dealing with local inter-class similarity, since S_3 and the penalization factor utilised by S_6 both penalize partially overlapping sets.

In summary, we conclude that fusing different similarity scores by employing the proposed fusion strategy only results in an improvement for five or the six databases and is only statistically significant for databases with high levels of noise and nonlinear distortion. This indicates that the proposed fusion process better deals with the effect of different intra-class variations and inter-class similarities across multiple databases when considering the $\mathrm{FNMR}_{\mathrm{zero}}$ and $\mathrm{FMR}_{\mathrm{zero}}$. Even though this fusion process may address the effect of intra-class variations and inter-class similarities to a certain extent, it does not solve the problems associated with high levels of noise and nonlinear distortion.

Chapter 7

Conclusion and future work

7.1 Conclusion

The research in this thesis investigated several similarity score calculation methods for two aligned minutia sets in the context of biometric fingerprint verification. Existing literature does not provide clear guidelines on how to calculate an accurate similarity score, nor does it demonstrate which existing similarity score is the most proficient, since the performance measures reported in existing studies include errors associated with the preprocessing procedures. The object of this study was therefore to group similar types of similarity scores together, identify the better existing similarity scores and from these results deduce guidelines for calculating accurate similarity scores. This study accomplished these objectives.

In Chapter 2 we provided an overview of existing advanced similarity score calculation methods. In Chapter 3 we explained the preprocessing procedures geared towards the removal of erroneous comparisons that may influence the similarity scores, and enabled us to sensibly compare different similarity scores in the subsequent chapters. In Chapter 4 we compared seven existing similarity scores and studied the problematic comparisons in order to identify what constitutes an accurate similarity score.

The results demonstrated that a similarity score has to address the two types of inter-class similarities without being sensitive to any intra-class variations. The study also suggested that global inter-class similarity can either be addressed by combining the local similarity method with distortion tolerant relaxation as proposed by Cappelli *et al.* (2010*b*), or by combining the local similarity method with the "percentage of paired minutia points". The aforementioned approach is slightly more sensitive to nonlinear distortion, while the latter approach is slightly more sensitive to noise. The local inter-class similarity can be addressed by including more paired minutia points for method S_3 or by adding a penalization factor for method S_1. When implemented on the six databases considered, methods S_1 and S_3, that both incorporate the above-mentioned features, generally produce the lowest EERs.

Since the local similarity forms an important part of the two strategies outlined in the above paragraph, Chapter 5 investigated different approaches to local similarity score calculation in order to establish guidelines for calculating the local similarity. The results demonstrated that the proposed local descriptor and similarity method that utilise all the neighbouring minutia pairs, perform the best across most of the databases when combined through different strategies during stage 2. This improvement in performance is significant when compared to the Minutia Cylinder Code (MCC) local similarity and implemented on the FVC2004 fingerprint databases. Said improvement is only marginal (and not statistically significant) for the other databases

Chapter 5 also indicated that it is best to consider an estimated 50% of the minutia pairs in order to calculate the final local similarity score through an assignment-based approach. This is due to the fact that when all the possible pairs are considered, the algorithm may be sensitive to noise, while the utilisation of only the best three pairs may not be distinctive enough. This approach is also more distinctive than the threshold-based approach for local similarity score calculation, and lead to a significant improvement in proficiency when implemented on databases with higher noise levels.

As mentioned above, S_1 and S_3 both address global inter-class similarity in different ways and are sensitive to different intra-class variations. These methods are however difficult to combine since both methods depend on different strategies for identifying the paired minutia points, which is vital for high distinctiveness within each method. However, not all comparisons are associated with the same intra-class variations or inter-class similarities, which points towards the possibility that using these methods in combination may improve the performance.

A method was therefore proposed in Chapter 6 that combines these different similarity scores in such a way that it results in a higher confidence. We concluded that combining the different types of similarity score calculation methods in this way marginally improves performance, since it addresses the different inter-class similarities, while also being slightly less affected by the different intra-class variations. However, said improvement is only significant when implemented on databases with high levels of noise and nonlinear distortion.

We finally concluded in Chapter 4 and 6 that a good similarity score has to address local inter-class similarity. Khanyile *et al.* (2014), Jain *et al.* (2008), and Cappelli *et al.* (2010*a*) propose ways for dealing with local inter-class similarity (Chapter 4). However, these methods are all sensitive to partially overlapping sets. This study therefore indicated that it is extremely difficult for similarity score calculation methods to deal with local inter-class similarity without being sensitive to partial overlap. A similar problem was encountered for nonlinear distortion and noise in the context of global inter-class similarity.

In summary, this study identified several of the best existing similarity score calculation methods for overcoming the adverse effects that preprocessing errors may have on the performance of these methods. The reported results provided guidelines for calculating accurate similarity scores, while certain problems that still exist, were identified. Better guidelines may slightly improve the accuracy, but do not solve the problems associated with high levels of intra-class variations and inter-class similarities. Even though we do not conclude that further improvements to similarity score calculation methods are impossible, we suggest that a significant increase in proficiency will more likely be achieved by addressing the problems (local inter-class similarity and nonlinear distortion) associated with the stages *preceding* the similarity score calculation stage.

7.2 Future work

This study identified three main problems that still complicate the development of accurate similarity score calculation methods, namely high levels of noise, nonlinear distortion and high levels of local inter-class similarity. We aim to address these problems in the future, however since the combination of the different similarity score calculation methods do not produce promising results, we will rather focus on improving the stages that *precede* the similarity score calculation stage. Future work will therefore investigate minutia point quality measures that reduce the impact of false minutia points *before* similarity score calculation. Furthermore, the thin plate spline model will be considered to account for nonlinear distortion *before* the similarity scores calculation stage is implemented. Finally we note that high levels of local inter-class similarity are the result of impostor comparisons that align only on the edge of a fingerprint. Future work may therefore consider core and delta points for alignment purposes, which may reduce the aforementioned problem without causing the similarity scores to be sensitive to partial overlap.

Appendices

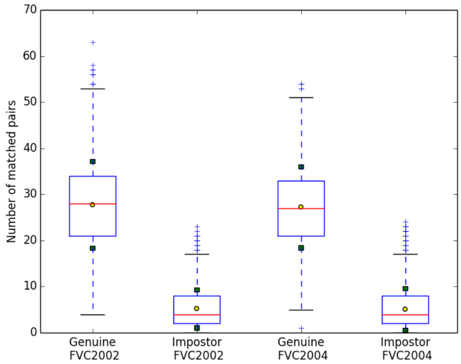
Appendix A

Determining the group thresholds for the problematic comparisons

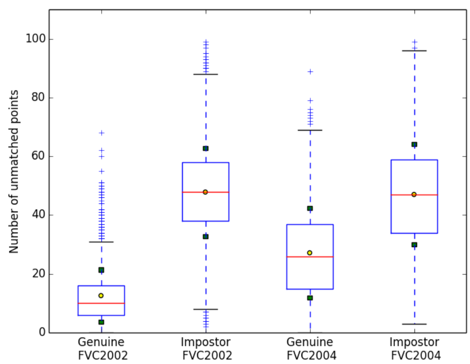
Defining the level of intra-class variations or inter-class similarities in fingerprint data is very challenging for several reasons. Firstly, these different types of variations and similarities are not equally represented in the same database or across different databases. The second reason is that although the impact of these variations and similarities depends on the specific score calculation methods, we aim to investigate how the different matching score calculation methods deal with these variations and similarities and can therefore not use the matching score calculation methods to determine the thresholds for the groups. In addition to different matching score calculation methods, we also have different numbers of problematic comparisons. Furthermore, the values of the different features used to measure the different intra-class variations and inter-class similarities have different ranges.

However, we know that a genuine comparison's estimated probability of being genuine is not only based on the different features, but also on how the differences in feature values compare to the same features for impostor comparisons. When we consider Figures A.1 (b), A.1 (c), A.1 (e), A.2 (a), and A.2 (b), it is clear that the impostor comparisons' distribution between the databases appears very similar even though large differences are present among the genuine comparisons. For these cases we used the impostor comparisons' distribution characteristics such as the standard deviation or different quartiles to determine the thresholds for the different groups.

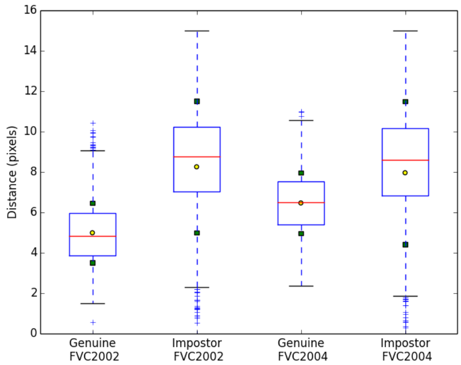
The different thresholds were determined as follows: The threshold for the matched ratio was specified at 0.5 since it is clear from Figure A.1 (e) that the maximum values for the whisker plots for impostor comparisons of both FVC2002 DB1_A and FVC2004 DB1_A are approximately 0.5. The threshold for the average distance between the minutia points forming pairs was taken to be the mean value of the lower quartile of the impostor comparisons for FVC2002 DB1_A and FVC2004 DB1_A as shown in Figure A.1 (c). The threshold for the local and global inter-class similarity features were taken to



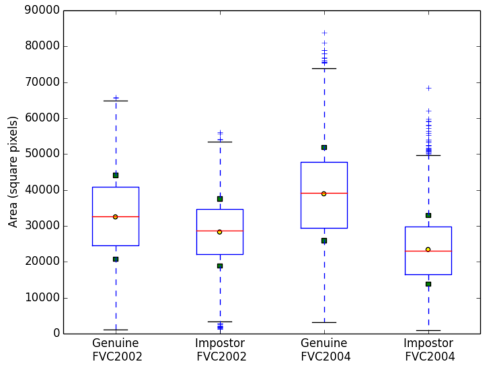
(a) Number of paired minutia points with the tolerance box approach.



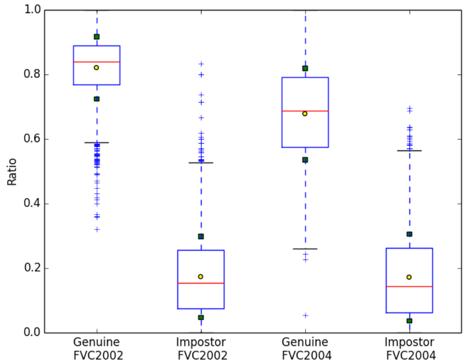
(b) Number of un-paired minutia points.



(c) Average distance between minutia pairs after alignment.



(d) Size of overlapping convex hull.

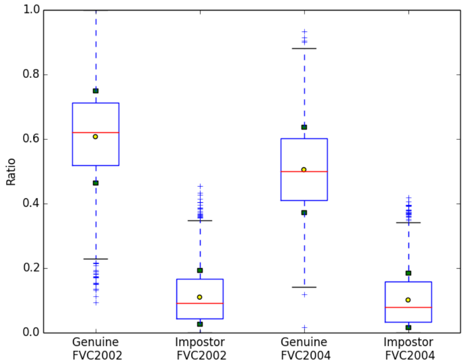


(e) Match ratio.

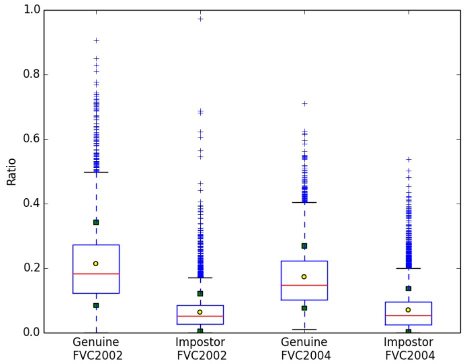
Figure A.1: Features measuring the levels of intra-class variations.

be the average value of the two upper limit standard deviations of the two databases' impostor comparisons.

Dealing with the partial overlap was more challenging since both the global



(a) Global inter-class similarity measure.



(b) Local inter-class similarity measure.

Figure A.2: Features measuring the levels of intra-class variations in the two databases.

and local inter-class similarities influence the impostor comparisons' overlapping size as is clear from Figure A.1 (d). In this case we calculated the average convex hull size of the sets in the different databases. FVC2002 DB1_A had a mean convex hull set size of 45203.3 square pixels with a standard deviation of 12381.6, while FVC2004 DB1_A had a mean set size of 53703.6 square pixels with a standard deviation of 15968.4. When the standard deviation is subtracted from both of these mean values the result is slightly larger than 30000 square pixels and the threshold was therefore set at 25000 square pixels.

The different thresholds were not calculated in exactly the same way, since the distributions differ too much and we only aimed to identify the worst problematic comparisons. In these scenarios the groups can not be directly compared as previously mentioned. Only the number of comparisons associated with a specific group may be compared for different score generators or databases. Therefore, the approach adopted in this section does not provide a perfect way of showing which score generators suffer from specific intra-class variations or inter-class similarities, but it still provides an indication of the proficiency of the score generators in question.

Appendix B

The type II erroneous comparisons

The type II erroneous comparisons are comparisons where the Local Greedy Similarity with Distortion Tolerant Relaxation (LGS_DTR) point matching algorithm failed to correctly identify the three most similar paired minutia points. As Chapter 3 explained, these comparisons were identified by superimposing the minutia points and three most similar paired minutia points onto the two fingerprint images, after which a manual study was performed to determine whether the point matching algorithm succeeded or failed. This is not the main focus of this study and the comparisons that were removed are therefore not included in Chapter 4, but we deemed it necessary to include it here. This protocol is followed since a manual study is subjective and may therefore be different when conducted by different individuals. In order to achieve the reported results, it is therefore important to know which type II erroneous comparisons were removed from each database. These comparisons are listed below, where a_b, c_d implies that impression b from individual a is being compared to impression d of individual c.

FVC2002 DB1_A

21_4,21_7	21_4,21_8	28_3,28_7	29_6,29_7
29_6,29_8	36_1,36_5	36_4,36_7	45_4,45_7
59_4,59_7	59_4,59_8	70_1,70_6	74_1,74_6
75_5,75_7	75_5,75_8	78_1,78_5	78_3,78_5
87_4,87_6	89_3,89_4	89_6,89_8	96_4,96_8
100_5,100_7			

FVC2002 DB2_A

4_4,4_6	4_4,4_7	5_1,5_5	8_5,8_8
32_6,32_7	33_1,33_6	36_4,36_5	38_2,38_3
60_3,60_8	60_4,60_8	68_2,68_5	68_5,68_7

73_2,73_8	77_2,77_4	77_2,77_6	77_2,77_7
78_1,78_5	89_1,89_6	94_5,94_8	97_7,97_8

FVC2002 DB3_A

3_4,3_8	11_3,11_8	16_1,16_2	16_1,16_4
17_1,17_5	17_3,17_4	17_4,17_5	17_4,17_7
19_1,19_4	19_1,19_5	19_1,19_6	19_1,19_7
19_1,19_8	19_4,19_5	24_2,24_7	24_2,24_8
24_4,24_6	24_4,24_7	24_4,24_8	26_5,26_6
30_2,30_6	30_5,30_6	30_6,30_7	31_1,31_8
36_1,36_4	36_2,36_4	36_3,36_4	40_2,40_5
40_2,40_6	40_3,40_6	40_3,40_7	40_4,40_5
40_4,40_6	41_3,41_4	41_5,41_6	41_6,41_8
42_1,42_5	42_1,42_6	42_2,42_6	42_3,42_6
42_3,42_7	42_5,42_6	42_6,42_8	47_1,47_6
47_2,47_6	47_4,47_6	47_5,47_6	47_6,47_7
48_5,48_7	52_4,52_7	53_3,53_5	53_5,53_6
53_5,53_7	53_5,53_8	55_6,55_8	64_1,64_8
71_2,71_6	71_5,71_6	71_6,71_7	71_6,71_8
72_3,72_7	72_4,72_7	72_5,72_7	74_4,74_5
82_3,82_8	82_4,82_8	86_1,86_5	86_2,86_5
86_3,86_5	86_4,86_5	93_1,93_6	93_1,93_8
93_2,93_6	93_3,93_6	93_3,93_8	93_4,93_6
94_1,94_2	94_1,94_6	94_2,94_4	94_2,94_5
94_2,94_6	94_3,94_6	95_4,95_5	96_2,96_3
96_2,96_5	96_2,96_6	96_2,96_7	97_2,97_5
97_2,97_6	97_2,97_7	97_2,97_8	98_1,98_2
98_2,98_3	98_3,98_5	98_7,98_8	100_2,100_6

FVC2004 DB1_A

1_4,1_6	1_4,1_8	2_2,2_4	2_4,2_5
2_4,2_6	2_4,2_8	2_1,2_4	2_3,2_4
2_4,2_7	3_1,3_4	4_2,4_4	5_4,5_7
6_2,6_3	6_3,6_7	6_3,6_8	7_2,7_4
7_3,7_8	7_4,7_6	7_4,7_7	7_1,7_4
8_3,8_7	9_3,9_7	9_3,9_8	11_2,11_7
13_2,13_3	13_3,13_6	14_1,14_3	14_3,14_5
14_3,14_6	14_3,14_7	14_3,14_8	14_7,14_8
14_2,14_8	14_2,14_3	14_5,14_8	15_1,15_4
15_4,15_5	16_1,16_4	16_4,16_8	16_5,16_6

17_1,17_3	17_2,17_3	18_2,18_8	18_3,18_8
19_1,19_4	19_4,19_8	20_2,20_3	20_3,20_6
20_3,20_7	20_5,20_8	20_3,20_8	21_7,21_8
22_1,22_3	23_1,23_5	26_1,26_3	26_2,26_3
26_3,26_4	26_3,26_5	26_3,26_8	26_6,26_8
28_3,28_4	28_3,28_5	28_3,28_7	28_3,28_6
29_3,29_8	29_3,29_6	32_3,32_6	34_1,34_3
34_2,34_8	34_3,34_8	36_7,36_8	37_3,37_7
38_1,38_3	39_6,39_8	39_3,39_7	39_6,39_7
46_1,46_2	46_3,46_8	46_1,46_3	49_3,49_4
49_3,49_8	50_2,50_3	50_3,50_5	50_3,50_8
52_3,52_4	52_3,52_5	52_3,52_6	52_3,52_7
52_3,52_8	52_1,52_3	54_1,54_3	54_2,54_3
54_3,54_8	56_4,56_8	56_3,56_8	56_1,56_8
57_2,57_4	57_2,57_5	57_2,57_6	57_2,57_7
57_2,57_8	57_1,57_2	59_4,59_7	59_2,59_3
59_2,59_4	60_3,60_7	61_3,61_5	61_4,61_6
62_1,62_8	62_3,62_8	62_4,62_7	62_5,62_8
62_1,62_4	62_2,62_8	63_1,63_2	63_1,63_4
63_2,63_4	63_4,63_5	63_4,63_6	63_4,63_7
63_4,63_8	63_3,63_4	64_1,64_3	64_2,64_8
64_3,64_4	64_3,64_6	64_3,64_7	64_3,64_8
64_2,64_4	66_2,66_4	66_4,66_5	66_4,66_6
66_4,66_7	66_1,66_4	68_1,68_3	68_1,68_8
68_3,68_6	68_3,68_8	69_1,69_4	70_3,70_4
70_3,70_7	70_3,70_6	72_1,72_6	73_1,73_3
73_3,73_7	73_3,73_8	74_1,74_3	75_3,75_7
75_1,75_3	75_3,75_8	78_2,78_6	78_2,78_8
82_2,82_6	83_3,83_4	83_2,83_7	83_3,83_7
84_1,84_4	84_3,84_4	84_4,84_7	84_1,84_3
84_6,84_8	85_1,85_8	85_5,85_8	85_6,85_8
85_2,85_8	85_3,85_8	86_2,86_8	86_3,86_8
86_4,86_8	86_7,86_8	86_1,86_8	87_4,87_8
88_1,88_8	89_3,89_8	90_4,90_5	90_4,90_8
90_4,90_5	92_1,92_8	92_3,92_8	92_5,92_8
92_7,92_8	96_1,96_4	96_2,96_4	96_3,96_4
96_3,96_8	96_4,96_6	96_4,96_7	96_4,96_8
96_5,96_8	96_5,96_7	98_2,98_5	98_2,98_6
98_3,98_5	98_3,98_6	98_4,98_8	

FVC2004 DB2_A

2_4,2_8	5_3,5_6	5_4,5_6	5_6,5_8

8_2,8_8	8_1,8_8	8_5,8_8	9_2,9_8
9_3,9_5	9_3,9_7	9_3,9_8	9_4,9_8
9_5,9_7	9_3,9_7	10_1,10_8	11_1,11_4
12_4,12_8	14_6,14_7	16_5,16_7	16_7,16_8
17_2,17_8	18_2,18_8	18_3,18_8	18_4,18_8
18_5,18_8	19_1,19_8	19_2,19_8	19_4,19_8
19_7,19_8	21_1,21_8	21_4,21_8	23_2,23_8
23_3,23_8	23_4,23_8	23_6,23_8	26_2,26_7
26_3,26_7	26_4,26_7	26_5,26_8	27_6,27_8
28_7,28_8	33_1,33_8	33_3,33_8	33_2,33_8
34_1,34_3	34_1,34_7	34_4,34_8	42_1,42_7
45_2,45_8	45_6,45_8	46_2,46_8	46_3,46_8
46_6,46_8	46_7,46_8	46_4,46_8	49_3,49_7
49_5,49_7	51_1,51_4	51_6,51_8	52_2,52_8
52_5,52_8	53_2,53_8	53_4,53_8	53_5,53_8
53_6,53_8	53_7,53_8	54_1,54_4	54_4,54_7
54_4,54_8	54_5,54_7	54_6,54_7	54_4,54_5
54_3,54_7	64_1,64_8	64_5,64_8	65_1,65_8
65_2,65_6	65_2,65_8	65_3,65_7	65_3,65_8
65_5,65_8	67_1,67_8	67_3,67_8	67_5,67_8
73_5,73_6	77_1,77_2	77_2,77_3	77_2,77_5
77_2,77_6	77_2,77_7	77_2,77_8	77_2,77_4
78_7,78_8	78_1,78_8	79_4,79_8	81_4,81_8
82_1,82_4	82_2,82_4	82_4,82_7	82_4,82_8
83_1,83_7	83_1,83_8	83_2,83_8	83_7,83_8
83_5,83_8	84_1,84_8	84_2,84_8	84_6,84_8
84_5,84_8	85_1,85_6	85_1,85_8	85_2,85_5
85_2,85_6	85_2,85_7	85_2,85_8	85_4,85_6
85_4,85_8	85_5,85_6	85_5,85_8	85_6,85_7
85_7,85_8	85_1,85_2	85_3,85_7	85_4,85_7
85_5,85_7	87_1,87_2	87_2,87_3	87_2,87_8
87_2,87_5	89_1,89_4	91_4,91_8	91_6,91_8
91_7,91_8	93_1,93_8	93_2,93_7	93_6,93_8
93_7,93_8	93_3,93_8	93_4,93_8	93_1,93_2
94_1,94_8	94_2,94_8	94_3,94_8	94_5,94_8
94_6,94_8	96_1,96_6	97_1,97_3	97_1,97_4
97_1,97_6	97_3,97_4	97_3,97_5	97_3,97_6
97_3,97_8	97_4,97_5	97_4,97_8	97_6,97_8
97_1,97_5	97_4,97_6	97_1,97_8	97_5,97_8
99_1,99_5	99_1,99_6	99_1,99_7	99_1,99_8
99_3,99_4	99_3,99_5	99_3,99_6	99_3,99_8
99_4,99_5	99_4,99_6	99_4,99_7	99_4,99_8
99_5,99_6	99_5,99_7	99_5,99_8	99_6,99_7
99_6,99_8	99_7,99_8	99_1,99_3	99_1,99_4

99_3,99_7	100_1,100_4	100_1,100_5	100_3,100_4
100_4,100_5	100_4,100_6	100_4,100_7	100_1,100_7
100_1,100_8			

FVC2004 DB3_A

1_2,1_4	6_1,6_6	10_2,10_4	10_4,10_8
19_1,19_2	19_3,19_8	20_3,20_7	21_2,21_8
21_5,21_8	22_1,22_8	27_2,27_8	29_2,29_4
29_2,29_5	29_2,29_7	29_3,29_5	29_3,29_7
29_3,29_8	29_5,29_7	29_6,29_7	29_2,29_8
30_3,30_7	31_6,31_7	32_4,32_8	34_3,34_8
34_4,34_7	34_4,34_8	34_6,34_8	34_1,34_8
34_2,34_8	34_5,34_8	37_5,37_8	41_2,41_7
41_3,41_4	41_3,41_6	42_7,42_8	42_1,42_8
42_3,42_8	43_1,43_5	43_2,43_5	43_5,43_6
43_5,43_8	43_3,43_5	44_3,44_8	44_1,44_4
46_2,46_8	46_3,46_8	47_4,47_8	48_2,48_7
48_3,48_7	48_4,48_7	48_1,48_4	48_2,48_4
49_1,49_7	49_1,49_8	50_1,50_8	50_2,50_7
50_2,50_8	50_4,50_7	50_6,50_7	61_1,61_7
61_1,61_8	61_6,61_7	61_6,61_8	61_5,61_7
61_2,61_7	62_1,62_8	62_2,62_7	62_3,62_7
62_3,62_8	62_4,62_7	62_5,62_7	62_6,62_8
62_6,62_7	64_2,64_8	64_5,64_8	64_6,64_8
65_2,65_8	65_3,65_8	65_5,65_8	66_5,66_8
67_2,67_7	69_1,69_8	69_3,69_8	69_4,69_8
69_5,69_8	70_1,70_5	70_1,70_7	70_2,70_7
71_1,71_8	71_6,71_8	72_1,72_4	72_1,72_5
72_1,72_6	72_1,72_7	72_1,72_8	72_2,72_5
72_2,72_6	72_2,72_8	72_3,72_5	72_3,72_6
72_3,72_7	72_3,72_8	72_4,72_8	72_5,72_8
73_1,73_7	75_1,75_6	75_6,75_8	75_1,75_5
75_5,75_6	75_5,75_7	75_5,75_8	76_1,76_4
76_4,76_8	76_5,76_8	76_3,76_4	77_3,77_7
78_1,78_8	78_2,78_8	78_3,78_6	78_3,78_8
78_4,78_8	78_5,78_8	78_7,78_8	80_3,80_7
80_2,80_7	82_4,82_5	88_1,88_3	89_1,89_7
89_4,89_7	89_6,89_7	89_5,89_7	90_4,90_7
90_4,90_8	90_5,90_7	92_4,92_7	92_5,92_8
93_2,93_5	94_1,94_4	94_1,94_8	94_2,94_4
94_2,94_6	94_2,94_7	94_2,94_8	94_2,94_5

96_1,96_4	96_1,96_6	96_1,96_8	96_2,96_8
96_3,96_8	96_4,96_8	96_5,96_8	96_6,96_8
96_7,96_8	97_4,97_8	98_2,98_3	100_5,100_7

List of References

Bazen, A.M. and Gerez, S.H. (2003). Fingerprint matching by thin-plate spline modelling of elastic deformations. *Pattern Recognition*, vol. 36, no. 8, pp. 1859–1867.

Bringer, J. and Despiegel, V. (2010). Binary feature vector fingerprint representation from minutiae vicinities. In: *Biometrics: Theory Applications and Systems (BTAS), 2010 Fourth IEEE International Conference on*, pp. 1–6. IEEE.

Cappelli, R., Ferrara, M. and Maltoni, D. (2010*a*). Minutia cylinder-code: A new representation and matching technique for fingerprint recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 12, pp. 2128–2141.

Cappelli, R., Ferrara, M., Maltoni, D. and Tistarelli, M. (2010*b*). MCC: A baseline algorithm for fingerprint verification in FVC-onGoing. In: *Control Automation Robotics & Vision (ICARCV), 2010 11th International Conference on*, pp. 19–23. IEEE.

Conti, V., Vitello, G., Sorbello, F. and Vitabile, S. (2013). An advanced technique for user identification using partial fingerprint. In: *Complex, Intelligent, and Software Intensive Systems (CISIS), 2013 Seventh International Conference on*, pp. 236–242. IEEE.

Dowdy, S., Wearden, S. and Chilko, D. (2011). *Statistics for research*, vol. 512. John Wiley & Sons.

Feng, J. (2008). Combining minutiae descriptors for fingerprint matching. *Pattern Recognition*, vol. 41, no. 1, pp. 342–352.

Feng, J. and Zhou, J. (2011). A performance evaluation of fingerprint minutia descriptors. In: *Hand-Based Biometrics (ICHB), 2011 International Conference on*, pp. 1–6. IEEE.

Feng, Y., Feng, J., Chen, X. and Song, Z. (2006). A novel fingerprint matching scheme based on local structure compatibility. In: *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 4, pp. 374–377. IEEE.

Fu, X. and Feng, J. (2015). Minutia tensor matrix: A new strategy for fingerprint matching. *PloS one*, vol. 10, no. 3, p. e0118910.

Fu, X., Liu, C., Bian, J., Feng, J., Wang, H. and Mao, Z. (2013). Extended clique models: A new matching strategy for fingerprint recognition. In: *Biometrics (ICB), 2013 International Conference on*, pp. 1–6. IEEE.

He, M., Horng, S.-J., Fan, P., Run, R.-S., Chen, R.-J., Lai, J.-L., Khan, M.K. and Sentosa, K.O. (2010). Performance evaluation of score level fusion in multimodal biometric systems. *Pattern Recognition*, vol. 43, no. 5, pp. 1789–1800.

Jain, A., Nandakumar, K. and Ross, A. (2005). Score normalization in multimodal biometric systems. *Pattern Recognition*, vol. 38, no. 12, pp. 2270–2285.

Jain, A.K. and Feng, J. (2011). Latent fingerprint matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 1, pp. 88–100.

Jain, A.K., Feng, J., Nagar, A. and Nandakumar, K. (2008). On matching latent fingerprints. In: *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pp. 1–8. IEEE.

Jain, A.K., Prabhakar, S. and Pankanti, S. (2002). On the similarity of identical twin fingerprints. *Pattern Recognition*, vol. 35, no. 11, pp. 2653–2663.

Jea, T.-Y. and Govindaraju, V. (2005). A minutia-based partial fingerprint recognition system. *Pattern Recognition*, vol. 38, no. 10, pp. 1672–1684.

Jiang, X. and Yau, W.-Y. (2000). Fingerprint minutiae matching based on the local and global structures. In: *Pattern recognition, 2000. Proceedings. 15th international conference on*, vol. 2, pp. 1038–1041. IEEE.

Kabsch, W. (1976). A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, vol. 32, no. 5, pp. 922–923.

Khanyile, N.P., de Kock, A. and Mathekga, M.E. (2014). Similarity score computation for minutiae-based fingerprint recognition. In: *Biometrics (IJCB), 2014 IEEE International Joint Conference on*, pp. 1–8. IEEE.

Liu, C. and Mago, V.K. (2012). *Cross disciplinary biometric systems*. Springer.

Maio, D., Maltoni, D., Cappelli, R., Wayman, J.L. and Jain, A.K. (2002). FVC2002: Second fingerprint verification competition. In: *Pattern recognition, 2002. Proceedings. 16th international conference on*, vol. 3, pp. 811–814. IEEE.

Maio, D., Maltoni, D., Cappelli, R., Wayman, J.L. and Jain, A.K. (2004). FVC2004: Third fingerprint verification competition. In: *Biometric Authentication*, pp. 1–7. Springer.

Maltoni, D., Maio, D., Jain, A.K. and Prabhakar, S. (2009). *Handbook of fingerprint recognition*. Springer Science & Business Media.

Medina-Pérez, M.A., García-Borroto, M., Gutierrez-Rodríguez, A.E. and Altamirano-Robles, L. (2012). Improving fingerprint verification using minutiae triplets. *Sensors*, vol. 12, no. 3, pp. 3418–3437.

Nandakumar, K., Chen, Y., Dass, S.C. and Jain, A.K. (2008). Likelihood ratio-based biometric score fusion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 2, pp. 342–347.

Nanni, L., Lumini, A., Ferrara, M. and Cappelli, R. (2012). Learning for combining fingerprint matchers: a case study FVC-OnGoing. *Neurocomputing: Learning, Architectures and Modeling, Nova Publishers, New York.*

Nanni, L., Lumini, A., Ferrara, M. and Cappelli, R. (2014). Combining biometric matchers by means of machine learning and statistical approaches. *Neurocomputing*, vol. 149, pp. 526–535.

Neumann, C., Evett, I. and Skerrett, J. (2012). Quantifying the weight of evidence from a forensic fingerprint comparison: a new paradigm. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, vol. 175, no. 2, pp. 371–415.

Peralta, D., Galar, M., Triguero, I., Paternain, D., García, S., Barrenechea, E., Benítez, J.M., Bustince, H. and Herrera, F. (2015). A survey on fingerprint minutiae-based local matching for verification and identification: Taxonomy and experimental evaluation. *Information Sciences*, vol. 315, pp. 67–87.

Tao, Q. and Veldhuis, R. (2013). Robust biometric score fusion by naive likelihood ratio via receiver operating characteristics. *Information Forensics and Security, IEEE Transactions on*, vol. 8, no. 2, pp. 305–313.

Wen, W., Qi, Z., Li, Z., Zhang, J., Gong, Y. and Cao, P. (2013). A robust and efficient minutia-based fingerprint matching algorithm. In: *Pattern Recognition (IAPR), 2013 Asian Conference on*, pp. 201–205.

Zhu, E., Yin, J. and Zhang, G. (2005). Fingerprint matching based on global alignment of multiple reference minutiae. *Pattern Recognition*, vol. 38, no. 10, pp. 1685–1694.