



Comparing performance of different fingerprint matchers by using StirMark distorted images

Masterarbeit

zur Erlangung des Diplomgrades an der
Naturwissenschaftlichen Fakultät der Paris-Lodron
Universität Salzburg

vorgelegt von
Michael Pober
mpober@cosy.sbg.ac.at

Betreuer:
Ao.Univ.-Prof. Dr. Andreas Uhl

Department of Computer Sciences
University of Salzburg
Jakob Haringer Str. 2
5020 Salzburg, AUSTRIA

Salzburg, im November 2010

Abstract

Present thesis ¹ aims to investigate the influences of “natural perturbations” in fingerprint images on the matching performance of different fingerprint matchers and in particular different types of fingerprint matchers (minutiae-based as well as non-minutiae-based). The term “natural” hereby denotes those kinds of perturbations, that frequently occur in real-life fingerprint application scenarios, mostly caused by the particular manner in which a finger is presented to the contact surface, when leaving the imprint. As fundamental test data for the studies in present work, three predetermined sets of fingerprint images from the Fingerprint Verification Contest 2004 are used. Each set has been acquired with a different type of fingerprint scanner, under varying finger- and positioning-related conditions. In order to simulate diverse types of natural perturbations then and to deliberately introduce them into the given fingerprint images, in this thesis a benchmark tool from the field of digital image watermarking robustness analysis, the *StirMark Benchmark*, is employed. In theory any collection of well-defined suitable image altering operations could serve the purpose, yet the *StirMark Benchmark* is in so far outstanding, as it gives great attention to standardization of the image manipulation process, as well as to reproducibility of experimental results. These aspects are also part of the inspiration for present work, as besides the pure investigation of the effects of natural perturbations on individual matching results, a further goal is, to evaluate, if the testing procedures applied in the experiments of this thesis, are suitable and furthermore beneficial as benchmark architecture for extensive performance evaluation of fingerprint matchers.

¹This work was funded by the Austrian Science Fund (FWF).

Contents

1	Introduction	1
1.1	Motivation	2
1.2	Goals	4
1.3	Terminology	7
2	StirMark Benchmark	12
2.1	Functionality	12
2.2	StirMark Tests included in the Benchmark	14
2.3	StirMark Benchmark in Present Work	15
2.3.1	Comments on the Selection of StirMark Tests	17
2.3.2	StirMark perturbations only in Probe Images, not in Gallery Images	18
3	Fingerprint Verification Contest and Test Data	20
3.1	Test Data	21
3.1.1	Notes on FVC2004 Test Data Usage in Present Work	24
3.2	Test Procedure	25
4	Fingerprint Matchers	27
4.1	Types of Fingerprint Matchers	27
4.2	Matchers in Present Work	29
4.3	Griaule Biometrics – Fingerprint SDK 2009	30
4.3.1	GrFinger Algorithm	31
4.3.2	Further Notes on Application in Present Work	31
4.4	VeriFinger	32
4.4.1	VeriFinger Algorithm	32
4.4.2	Further Notes on Application in Present Work	32
4.5	bozorth3	33

4.5.1	bozorth3 Algorithm	33
4.5.2	Further Notes on Application in Present Work	34
4.6	Phase Only Correlation Matcher	35
4.6.1	Properties of the Phase-Only Correlation Function	36
4.6.2	Band-limited Phase-Only Correlation Function	38
4.6.3	POCmatcher Algorithm	39
4.6.4	POCmatcher Implementation	40
4.6.5	FVC2004-Database Specific Differences	49
4.7	FingerCode Matcher	52
4.7.1	FingerCode Algorithm - Enhancement and Segmentation	54
4.7.2	FingerCode Algorithm	57
4.7.3	Enhancer and Segmentation Implementation	59
4.7.4	FingerCode Matcher Implementation	64
4.7.5	FVC2004-Database Specific Differences	75
5	StirMark Experiments	79
5.1	Original Images - Basis of Comparison	80
5.2	General StirMark Influence	85
5.2.1	Consequence	89
5.3	Additive Noise	89
5.4	Median Cut Filtering	96
5.5	Convolution Filtering – Mean Filtering	105
5.6	Convolution Filtering – Modified Gaussian Filtering	112
5.7	Remove Lines	122
5.8	Rotations	127
5.9	Affine Transformations – General Notes	130
5.10	Affine Transformations – Stretching in X-Direction	134
5.11	Affine Transformations – Stretching in Y-Direction	142
5.12	Affine Transformations – Shearing in Y-Direction	150
5.13	Affine Transformations – Shearing in X- and Y-Direction	161
5.14	Small Random Distortions	171
5.15	Latest Small Random Distortions	184
6	Summary	200
	Bibliography	212

Chapter 1

Introduction

In recent years biometric systems have become more and more important. Not just because, according to some, the world has become a more dangerous place, but far more significantly, because biometric systems – foremost fingerprint recognition systems – have become affordable and are meanwhile being employed in a broad variety of application scenarios: from immigration screening at airports, via the often-cited access-control systems, regulating the entry to restricted areas (including private homes), to the fingerprint sweep-scanner, that nowadays we commonly find built into consumer notebooks, to replace the necessity to enter (and more troublesome: remember) passwords.

As mentioned, fingerprint recognition is by far the most prevalent biometric application in use today. Consequently we also have a broad selection of (commercially) available fingerprint recognition systems, occasionally also applying non-conventional (i.e. non-minutiae-based) algorithms for fingerprint matching. To get an overview over the matching performances of individual matchers, several attempts have been made to systematically and objectively compare different systems with each other – one such renowned attempt is for example the Fingerprint Verification Contest (described in chapter 3). Often the sample data, used in these competitions, is compiled with the special intention, to include “difficult” fingerprint images – i.e. images containing perturbations, as they might occur in real-life fingerprint application scenarios (for instance images, where the finger imprint is only partially visible, or images, where the ridge-lines of the imprint are not clearly perceivable, or images, which contain certain types of noise).

One potentially interesting aspect, that would extend said comparative studies and that so far, to the best of my knowledge, has not been explored, is the reaction of the individual fingerprint matchers to particular types of perturbations within fingerprint images. Such an investigation is basically the topic of present thesis.

1.1 Motivation

The StirMark Benchmark is a generic benchmark test for evaluating the robustness of digital image watermarking methods. It was developed by Fabien A. P. Petitcolas et al. [27, 20] in response to the problem that, by the time afore-referenced papers were published, comparison of different digital watermarking techniques was difficult, due to the lack of commonly

used and accepted evaluation procedures. Hence Petitcolas et al. had set out, to establish a quasi-standard for robustness evaluation in digital watermarking, by providing a well-defined benchmark-architecture.

The basic idea behind the StirMark Benchmark is, to embed a watermark in a sample image and then to attack it, by introducing a variety of small, ideally imperceptible perturbations into this image. Finally an attempt is made, to extract the watermark again, thereby the respective certainty of extraction is measured. The attacks themselves are structured in a given set of tests, each corresponding to an individual, well-defined type of perturbation, whose characteristics can be adjusted via a small set of parameters. (For more details on the StirMark Benchmark, its functionality and the various types of image manipulation operations provided, please refer to chapter 2).

The pursuit of Petitcolas et al., to establish a common architecture for evaluation of digital watermarking techniques, is also part of the motivation of present work. The aim to use a single, openly available and well-defined benchmark to simplify and unify the testing and comparison of watermarking systems could potentially also be adopted for fingerprint matcher analysis and comparison.

The Fingerprint Verification Contest (FVC) for instance, introduced in detail in chapter 3, is basically already that. It represents a sophisticated infrastructure for comparing various fingerprint matchers: specific test data is provided, the matching procedure and preconditions to be met, are well-defined and so are the criteria and methodology for performance evaluation (to just name some of the aspects that make the FVC very well suited for objective matcher comparison). As the large number of participants suggests, the FVC also finds/found high response and acceptance in the fingerprint matching community.

The aspect, in this regard, that I would like to investigate in present work, is following: On one hand we have a comprehensive tool – in our case the StirMark Benchmark – that enables us, to easily introduce a broad variety of perturbation types into images. The perturbation types themselves are predefined and their intensity is adjustable via a small set of given parameters. On the other hand we have the Fingerprint Verification Contest with aforementioned assets. My question is now: Is it feasible and more important, beneficial, to combine both evaluation systems into a, let’s say, “extended” benchmark?

What would this *extended benchmark* look like and what could be its benefits? The testing procedures, as well as the methodology for analysis and comparison of the results, could, for example, be adopted from the Fingerprint Verification Contest. As test data we would use a certain predetermined set of fingerprint images – ideally, as is the case with the test images of the FVC, separate sets of samples, acquired with a representative array of diverse fingerprint scanners under varying finger- and positioning-related conditions (please refer to section 3.1 for details on the FVC’s sample images).

Still no different to the FVC, this basic set of test data is then used to evaluate the fingerprint matchers. Regarding the entire matching results generated in this first “test-run”, the matchers are compared to each other and further a ranking is determined, based on their individual matching performances.

While the analysis so far gives us an extensive overview over the matching performances of the various fingerprint matchers, its scope is still limited to the basic set of predetermined sample images. Concerning this point now, my proposition is to introduce the StirMark benchmark (or any other collection of well-defined suitable image altering operations for that matter). We use the available test data and exploit the diverse StirMark benchmark tests, by

applying their respective image manipulation operations to the original fingerprint images. This gives us the possibility to create a user-defined variety of additional, “new” sets of sample images, each with a specific type of perturbation, whose properties are adjustable via a given set of parameters. Thereby we not only considerably extend the available test data, but even more important, we have the possibility to simulate designated perturbations within the fingerprint images, the way they commonly appear in real-life application scenarios.

When we consider the matching results and the ranking, established beforehand by using the original, unmodified fingerprint images in the first test-run, quasi as our “ground truth”, then we can observe, how the fingerprint matchers react to certain types of perturbations – taken by themselves as well as in comparison with each other. Put differently, by comparing the results of the first test-run with the results generated by using a set of perturbed images, we can investigate questions, like in the following examples:

- Regarding a single matcher, in how far does a specific type of perturbation influence the matching results? Is the matching performance strongly deteriorated or can the respective matcher cope well with it?
- How do the results of one matcher change in relation to the results other matchers? Are all matchers equally strong influenced by the perturbations, or are some showing a more distinct response than others?
- Do perturbations, or certain types thereof, cause changes in the overall ranking of the fingerprint matchers? For example, is the “best” matcher exhibiting a superior overall performance in any case?
- Are we able to identify general differences in the responses to perturbed images, depending on the particular *types* of fingerprint matchers? (See section 4.1 for details on various matcher types).

Bearing the idea of combining FVC-like fingerprint matching procedures with StirMark-like image manipulation into an “extended benchmark”, further the discussed outlines for according experiments, as well as above posed questions in mind, we can formulate the goals of present work:

1.2 Goals

The goals of present work can be stated as follows:

- *Investigate the influence of different kinds of natural perturbations in fingerprint images on the matching performance of diverse types of fingerprint matchers.*
- *Determine the applicability of the well-known digital image watermarking robustness-testing tool StirMark Benchmark for simulation and induction of said natural perturbations.*
- *Evaluate in general, if the experiment procedures applied, could form the basis for a common fingerprint matcher comparison and analysis benchmark.*

Following now an explanation of several terms, appearing in previous definition:

Fingerprint images In my experiments in present work, technicalities of fingerprint image acquisition are not of explicit concern, consequently I have decided to use readily available test data. I chose the sample image databases from the Fingerprint Verification Contest 2004. Please refer to chapter 3 for details.

Furthermore the images are processed "as is", in other words, no quality based selection is made. Cases of "Failure to enroll" will not be treated specially either, but will simply lead to fingerprint templates producing a matching score of 0.

Matching performance The fingerprint matchers are tested on aforementioned sample data, following the procedure specified in the Fingerprint Verification Contest. Details on the testing procedure can be found in section 3.2 on page 25. For performance evaluation and comparison, the matching results are analyzed and the following indicators inspected:

- Score Distribution of the genuine and impostor scores
- False Matching Rate (FMR) and False Non Matching Rate (FNMR)
- Equal Error Rate (EER)
- Receiver Operating Curve (ROC)

Further information on these performance indicators can be found in section 1.3 on page 7.

Types of fingerprint matchers Three types of fingerprint matchers, differing in the particular properties regarded for determination of similarity between two fingerprints, are compared in present work:

- 3 Minutiae-based matchers
- 1 Correlation-based matcher
- 1 Ridge Feature-based matcher

Concerning the latter two: In lack of readily available realizations of purely non-minutiae-based fingerprint matchers, own implementations of already existing algorithms were created. Chapter 4 on page 27 presents detailed information on the employed fingerprint matchers.

Perturbations In present work the term "perturbation" is used to denote any kind of property of a fingerprint image, that impedes the matching process and (if not remediable) consequently leads to a "non-ideal" matching score. Those perturbations can be rotations, shearing or warping of the finger imprints, noise or blur within the image, etc.

Simulation and Induction of Perturbations This refers to the act of deliberately introducing a certain type of perturbation into a fingerprint sample image. The perturbation itself is thereby generated artificially, according to exact user-specifications. As stated on several occasions above, in course of the experiments of present work, the StirMark Benchmark will be applied for the purpose generating and introducing the perturbations. Detailed information on the StirMark Benchmark and the various perturbation types it provides, is given in chapter 2 on page 12.

Natural Perturbations By “natural” we refer to those kinds of perturbations that frequently appear under normal usage-related circumstances in real-life fingerprint application scenarios. This constraint applies to the specific type of perturbation as well as to the respective intensity.

For example a mirroring of the image is not considered as possible perturbation, as this effect will most likely not appear “naturally” in real-life applications. On a side note, also none of the regarded fingerprint matchers innately is prepared to deal with mirrored images.

Another example: While for instance a rotation of up to $\pm 180^\circ$ of one finger imprint in relation to another might very well appear, a stretching-like deformation in horizontal direction by more than, say, 50% is very unlikely. For illustration, Figure 1.1 shows an original fingerprint and the same fingerprint stretched in horizontal direction by “only” 35%.

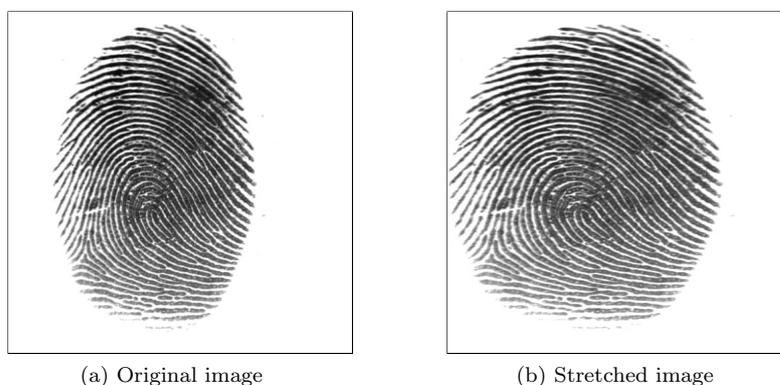


Figure 1.1: A sample fingerprint image and the same image stretched in horizontal direction by 35%

I also want to use the generated data and findings to get to a concluding verdict about the suitability of my experimental approach for a common generic benchmark of fingerprint matcher performances. (Referred to as “extended benchmark” in section 1.1).

1.3 Terminology

Real-life Fingerprint Application Scenario As such I consider any kind of application scenario, where automatic fingerprint processing is actually being used in nowadays life, ranging from simple access control at security-gates, via personality verification in combination with a biometric passport, to forensic analysis. However these application scenarios might look like, the one determinant that – from my point of view – they have in common, is, that the initial process of fingerprint acquisition does not happen under precisely set and controlled conditions. (To be accurate, in case of forensic analysis, the acquisition of a fingerprint is most certainly a very precise process, conducted by highly trained specialists – so in this context, henceforth please consider the incident, that led to the latent fingerprint in first place). In other words, when acquiring several imprints of a single, identical finger in a real-life application scenario, it is most likely,

that no two imprints will be totally equal. This aspect is due to the broad variety of conditions that can influence the appearance of the resulting fingerprint. Following I will only name very few:

Conditions with respect to the finger The moisture of the fingertip; the angle, the finger is presented to the contact surface in (relative to the axis normal to the surface, as well as relative to the longitudinal axis of the finger); the amount of pressure applied, furthermore its direction.

Conditions with respect to the acquisition-environment The structural properties of the contact surface; if the surface is clean or dusty, or if residual fingerprint details from previous scans are present; scan- or transmission errors that might occur; the type of scanner.

Perturbation In regard to the previous explanation of the *real-life fingerprint application scenario* and the mentioned conditions influencing the actual appearance of a fingerprint, we can get to a supplementary definition of the term *perturbation* as used in present work (please compare to the original definition given in section 1.2): We can say, it denotes the difference between a theoretical ideal imprint of a certain finger and the actual acquired one.

Fingerprint Template This is an application dependent representation of a processed fingerprint. It contains the specific features extracted from the imprint, that are supposed to make it distinct and identifiable. For many applications it is favorable to have small sized templates, in order to reduce processing time and memory requirement. For security reasons, an additional design-criteria might be, that given a template, it should be impossible to reconstruct the original fingerprint from it.

Gallery Image In many application scenarios we have a database, containing enrolled fingerprints, i.e. fingerprints that have been (voluntarily or involuntarily) registered in combination with information which identifies the corresponding person. In present work, I will denote such a database as *gallery*. As it is application dependent, in what form a fingerprint is stored in the gallery – e.g. as original acquired image or as preprocessed image or as specific template – I will generically use the term *gallery images* to refer to the according database entries.

Probe Image With this term I will refer to an acquired fingerprint image, that is not meant for enrollment, but to be tested against one or many gallery image(s).

Fingerprint Verification In *verification systems* a claim is made about the identity of a provided fingerprint. The system then compares this probe image with the gallery image(s) previously enrolled for this particular identity and determines, if the claim is correct. Hence (in theory) verification systems conduct *1 to 1* comparisons.

Fingerprint Identification An *identification systems* is presented with a plain fingerprint and its purpose is then, to determine the corresponding identity. This is done by comparing the probe image to the entire set of gallery images. In other words, identification systems conduct *1 to many* comparisons.

Matching Score When presented with two fingerprint images to be tested, fingerprint matchers in general will return a corresponding score, representing either the similarity or the difference between the two imprints. Regarding the fingerprint matchers applied in present work, four matchers return a similarity score and one returns a measure of distance. To facilitate comparability, the matching scores are treated as follows:

1. The distance score s_D is transformed into a similarity score s_S by calculating $s_S = \max_D - s_D$, where \max_D represents the maximum possible distance and its value has been empirically determined, inspecting the matching results for all available fingerprint images, including StirMark-perturbed ones. (Different methods to transfer a distance value into a similarity score can be found for example in [1] and [6]).
2. All scores have been normalized to a range of [0,1].

Genuine Scores Scores generated for those cases, where the presented probe image corresponds to the gallery image enrolled for claimed identity.

Impostor Scores Scores generated for those cases, where the presented probe image does not correspond to the gallery image of claimed identity.

Match, Non-Match The decision, if two fingerprint images are declared as *match* or as *non-match* is made depending on a certain *threshold* t . If the calculated matching score is less than t , the two fingerprints are declared as *non-match*, if the score is greater than or equal t , the pair is declared as *match*.

False Match Rate (FMR) A *false match* denotes the error, when fingerprints of two different fingers are mistakenly declared to be from the same finger. The *false match rate* is the number of false matches per total number of matching-tests performed.

False Non Match Rate (FNMR) A *false non match* denotes the error, when two fingerprints acquired from the same finger are declared to be from two different fingers. The *false non match rate* is the number of false matches per total number of matching-tests performed.

Equal Error Rate (EER) Analyzing a set of matching results, the *equal error rate* denotes the point, where corresponding FMR and FNMR have an equal value.

Zero False Match Rate (ZeroFMR) This measure denotes the lowest FNMR value established in a set of matching results, while at the same time the FMR is still 0%.

Score Distribution (SD) This graph portrays the probability density function of the impostor scores and of the genuine scores. See Figure 1.2 for an example.

Receiver Operating Curve (ROC) A diagram very well suited for analyzing the performance of a verification system. In present work we plot the FNMR against the FMR. This allows to analyze the inevitable trade-off between the two error rates. See Figure 1.3 for an example.

Notes on the ROC plots in present work:

- With the threshold running from 0 to the maximum matching score (here: 1), a ROC plot naturally covers the complete range of values, from FMR = 100% and FNMR = 0% to FMR = 0% and FNMR = 100%. In certain StirMark related experiments though, the quality of the fingerprint images is derogated so strong, that numerous match-pairings result to a matching score of 0. In the subsequent analysis therefore, already the first threshold-step leads to a large amount of false matches and false non-matches. In the corresponding ROC plot then, where the single data points are connected by lines for better perceptibility, the large gap between the first data point of FMR = 100% and FNMR = 0% and the

second data point representing said high error rates, would lead to an irritating long, straight connection line. As this segment might give the false impression of several result values lying on its path, I decided to exclude the first data point from the plots. Consequently a ROC curve then not terminates, as natural, on point (1,0) but on the following calculated pair of error rates.

For an illustrative example, take the results produced by the fingerprint matcher VeriFinger in the *Additive Noise* test for perturbed images of database DB2 (see page 91 for test-related details). Inspecting the results of, for instance, noise level 15, we see, that already 46% of all fingerprint pairings cause a matching score of 0. In the analysis then, when the threshold is incremented for the first time, we immediately obtain a FMR of 44% and a FNMR of 28%. The ROC plot in Figure 5.8c on page 94 shows, that if the natural first rate pair of FMR = 100% and FNMR = 0% would be included and connected in the plot, we would obtain a long straight line from (0.44,0.28) to (1,0), falsely suggesting the presence of actual data values.

- As a consequence of the large amount of data, the plots tend to become overloaded, if every single data value is represented by an individual point/mark. Hence to improve the perspicuity of the plots, in most cases, only every second or third data value will be indicated.

Test-Run The term *test-run* will be used in connection with the testing activities, based on the test-protocol specified in the *Fingerprint Verification Contest* (please refer to chapter 3 for details): Following the testing procedures defined in the protocol and using the pre-determined set of test fingerprint data, a series of all-together 7750 matches will be conducted. For simplicity I will henceforth refer to such a complete matching-series as single *test-run*.

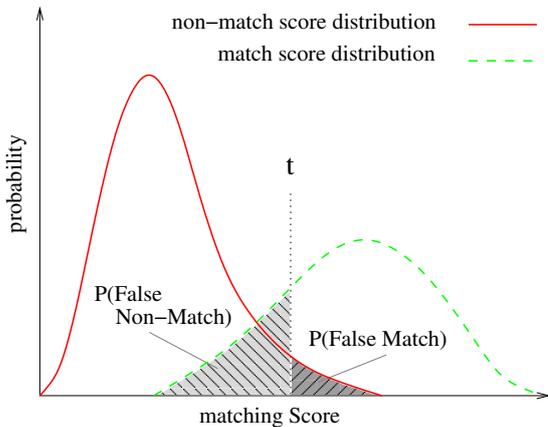


Figure 1.2: Sample of genuine and impostor *score distributions*.

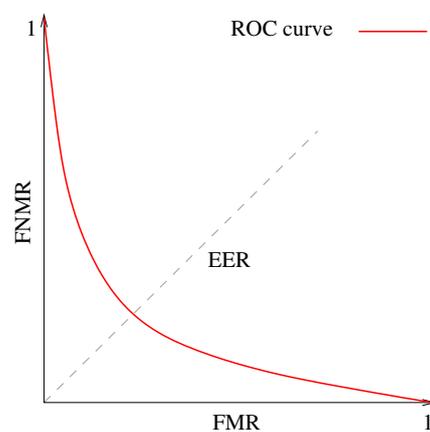


Figure 1.3: Sample of a *ROC plot*. The gray dashed line marks the location of the EER.

Chapter 2

StirMark Benchmark

The StirMark Benchmark is a generic benchmark test for evaluating the robustness of digital image watermarking methods, developed by Fabien A. P. Petitcolas et al. [27, 20].

Petitcolas et al. had observed, that around the time aforementioned papers were published, most studies on digital image watermarking were using own test data, own methodology and an own set of criteria to determine and report the robustness of the proposed methods. Comparison of different watermarking systems was therefore not easily possible. This was one of the reasons that motivated Petitcolas et al. to develop the StirMark Benchmark, aiming to establish a quasi-standard for robustness evaluation by providing a well-defined benchmark-architecture, that enables fair and independent comparison of various digital watermarking techniques. With such a common tool authors would be provided the means to easily test their own developments and subsequent reporting could be simplified to merely publishing a table of benchmark results (supplemented, if necessary by a listing of specific parameter settings).

2.1 Functionality

The basic idea behind the robustness tests in the StirMark benchmark is, that a digital watermark within an image can be attacked and possibly rendered useless, by introducing small, ideally imperceptible perturbations into the marked image. To be suitable for application in a common generic benchmark, the specific types of perturbations are pre-defined and the respective intensity is adjustable via a given set of parameters.¹ Typical attacks are rotation, scaling, color quantization, various random geometric distortions, etc.

The procedure of a single robustness test for a watermarking scheme to be evaluated, is thereby as follows (based on the definition given in [20]):

1. Take a sample image and embed a watermark with the greatest strength which does

¹The same principle also applies to digital watermarking methods for other types of multimedia objects, like audio signals, but as in present work we are only concerned with images, I will limit the discussion accordingly. For more information on watermarking robustness evaluation for other multimedia object types, as well as on the corresponding extension of the StirMark Benchmark, please refer to [26] and [28]. In mentioned papers Petitcolas et al. also describe their idea for a public fully automated evaluation service based on the StirMark Benchmark.

not introduce annoying effects (i.e embed as such, that the image quality – for a given quality metric – remains greater than a given minimum).

2. Attack the watermark by introducing a specific, pre-defined perturbation into the marked image.
3. Try to extract the watermark from the perturbed image and measure the certainty of extraction.

Petitcolas et al. note in [27], that while many of the contemporary watermarking schemes could survive mentioned basic image manipulations, they could not cope with combinations of them. Hence they developed the StirMark attack, which in itself combines a pre-defined set of differing basic ones. (For a detailed description of the StirMark attack, please refer to section 5.14 on page 171).

In order to provide previously mentioned extensive benchmark-architecture for robustness evaluation of digital watermarking techniques, Petitcolas et al. then bundled their own StirMark attack together with a large variety of basic attacks into the encompassing StirMark Benchmark.² Currently this is available in form of the “StirMark Benchmark 4.0” on the website of Fabien A. P. Petitcolas [25].

2.2 StirMark Tests included in the Benchmark

Following, a quick overview over the individual tests provided in the StirMark Benchmark. More detailed information can be found for example in [20], or in the corresponding sections of chapter 5 of present document.

PSNR This test measures the PSNR of an image after the watermark to be tested has been embedded. The expected strength of the embedding is adjustable and the test reports a failure, if the PSNR does not meet this value ($\pm 10\%$ of tolerance).

Embedding Time The watermark to be tested is embedded a user-specified number of times and the embedding time is measured.

Additive Noise Additive random noise is introduced to the input image. The amount of noise is adjustable and can range from “none” to “completely random image”.

JPEG The image is JPEG compressed. The level of quality is thereby adjustable.

Median Cut Filtering This test applies a median cut filter to the input image. The size of the filter mask can be set.

Convolution Filtering A generic test for any kind of convolution filtering. The specific filter mask has to be defined by the user.

Self-Similarities A not otherwise specified test by Eurécom, dealing with self-similarities. The parameters allow for a selection of the color-space and the respective channel(s) to be attacked. Further the user can choose between two test-modes: spatial and exchange.

²Henceforth I will denote all attacks/tests, that are part of the StirMark Benchmark, as “StirMark attacks/tests” and to avoid ambiguity I will use the term “*the* StirMark attack/test”, with emphasis on *the*, when referring to the one particular attack/test, developed by Petitcolas et al., which comprises a combination of several specific attacks.

Remove Lines This tests removes rows and columns from a given image at the specified frequency k – “remove 1 line in every k lines.”

Cropping While maintaining the image center, this tests crops the image to a user-defined percentage of its original size.

Rescale Rescales the image to a user-defined percentage of its original size.

Rotation Rotates the image by a given angle.

Rotation and Cropping Rotation by a given angle followed by cropping. Thereby the output image is cropped so much, that it only contains parts of the original input image and no black corners, as the normal rotation introduces for areas without corresponding input image data.

Rotation and Rescaling Same procedure as for *Rotation and Cropping*, followed by scaling to the original input image size.

Affine Transformations Generic test for arbitrary affine image transformations. The user specifies the parameters a, \dots, f of the inverse transformation matrix of the form:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix}$$

Small Random Distortions *The* StirMark test. Being a combination of several basic manipulations (see aforementioned ones), this test aims to simulate a resampling process, i.e. the errors introduced when printing an image and then scanning it again.

Latest Small Random Distortions Introduced in version 4.0 of the StirMark Benchmark, this test represents a modification of the *Small Random Distortions* test. It replaces two original, sine-function-based processing steps, responsible for generating global and local image-distortion, with a single procedure, employing randomized “Distortion Maps” instead. (See sections 5.14 and 5.15 for details).

2.3 StirMark Benchmark in Present Work

As the focus of present work is exclusively on fingerprint matching performance and has no connection to any digital watermarking aspects, I am interested in the StirMark Benchmark solemnly for its following properties:

- It provides a large collection of various image manipulation operators, suited to introduce perturbations into a specified set of images.
- The available image manipulations include basic ones, like rotation, shearing, convolution filtering, etc., as well as two different versions of *the* StirMark test, which combines several perturbation types in itself (see previous section 2.2 for details).

By repeated application of StirMark tests – i.e. by applying a test on already perturbed images – the perturbations can also be arbitrarily combined (with limitations though – see section 5.2 for details).

- The available set of perturbation types is predefined and the respective intensities, as well as certain properties, can be adjusted via a given, small set of parameters.

- Said set of parameters also allows for a single test to be run repeatedly, while automatically changing the initialization values. E.g. for the *Rotation* test it is possible to either specify a list of angular values or alternatively, to set a start value, an end value and a step width. The test is then executed for each angle, specified by these settings.
- The setup of the StirMark Benchmark, as well as the specific settings for the individual tests, is comfortably done via a configuration file, specified at benchmark start-up.
- The StirMark Benchmark is freely available and source code is included in the downloadable package [25].

For the experiments of present work a subset of the complete range of StirMark tests is used. This subset consists of only those tests, that (to my estimation) simulate “natural” perturbations – in other words, tests, whose influence on fingerprint images creates perturbed versions thereof, that resemble cases appearing in real-life fingerprint application scenarios – *rotated* fingerprint images being a very common example. (Also compare to the definition of the term “natural perturbation” given on page 6). The selection of tests for the experiments is as follows:

- *Additive Noise*
- *Median Cut Filtering*
- *Convolution Filtering* . . . in configurations for:
 - Mean Filtering
 - Modified Gaussian Filtering
- *Remove Lines*
- *Rotation*
- *Affine Transformations* . . . in configurations for:
 - Stretching in X-direction
 - Stretching in Y-direction
 - Shearing in Y-direction
 - Shearing in X- and Y-direction
- *Small Random Distortions*
- *Latest Small Random Distortions*

More detailed information on each individual test, on why it was chosen, as well as on the particular range of examined parameter configurations is given in the respective sections of chapter 5.

2.3.1 Comments on the Selection of StirMark Tests

No Combinations of StirMark Tests

As already stated on several occasions, the aim of the experiments in present work is to establish the differences in matching performance among the various types of fingerprint matchers, in regard to specific types of perturbations (rotation, noise, etc.). For that matter, the StirMark manipulations will be applied to the test data images only separately, one at the time. In other words, the fingerprint images will either be rotated or sheared or convolution-filtered etc., but no combination of StirMark tests will be considered. This has two reasons:

- First, as mentioned before, the StirMark tests – *Small Random Distortions* and *Latest Small Random Distortions* – themselves already constitute a combination of basic manipulations. So (to a certain degree) combinations of perturbation types will be studied here implicitly.
- Second and more important: Only by regarding the StirMark manipulations separately, the influences of their respective perturbations on matching performance can be determined, because only then the observed changes in the matching results are distinctively attributable.

Dealing with Changed Image Sizes

Another aspect that had to be dealt with: For certain StirMark tests the dimensions of the perturbed output images are likely to differ from the dimensions of the original input images. To preclude any influences that a difference in sizes of a gallery and a probe fingerprint images might have on the matching performance, the original image dimensions have to be reestablished:

- The tests *Affine Transformation* and *Rotation* produce output images with larger dimensions, than that of the input images. (Except, of course, for rotations by multiples of 90°). In these cases the output images are being cropped to the original dimensions, while maintaining the image center.
- The test *Remove Lines* per definitionem reduces the size of the output images. To restore the original image dimensions, the perturbed images are extended at the borders with monochrome pixels of a user-defined background color. Again the image center is maintained during the operation.

As the necessary additional image manipulations for cropping and boarder-extending are not part of the StirMark Benchmark, corresponding operators of the ImageMagick suite [41] are applied for that purpose.

Reduced Number of Tests for Non-Minutiae-Based Fingerprint Matchers

As mentioned in the introduction, in section 1.2, five different fingerprint matchers will be used for the experiments of present work. Two of these matchers are own implementations of existing non-minutiae-based algorithms. (Please refer to chapter 4 for details). Unfortunately the two self-implemented matchers are extremely slow (which, to my opinion,

is mostly due to certain very time-consuming characteristics of the underlying feature extraction and matching algorithms). Consequently, running these matchers on the complete projected collection of sets of StirMark-perturbed images would have taken several months of processing time. Therefore, while the minutiae-based matchers will still be tested on the entire collection, the two non-minutiae-based fingerprint matchers will only be applied to a limited number of image sets. Details can be found in the respective sections of chapter 5.

2.3.2 StirMark perturbations only in Probe Images, not in Gallery Images

An important aspect that has to be noted in regard to the application of the StirMark benchmark in course of the experiments of present work, is, that in each pair of fingerprint images to be matched, only the probe image will be StirMark-perturbed, while the image, representing the enrolled gallery fingerprint, will be taken from the original, unperturbed set of sample images. There are mainly two reasons for choosing this modus operandi:

- For one, as stated above, I aim to be able to distinctively attribute a change in a matcher’s performance to the type and intensity of the respective StirMark-perturbation in question. However, matching two perturbed fingerprint images might lead to an interaction of the perturbations, obfuscating the potentially observable influence, a certain perturbation type would originally have on the matching performance.

For illustration, we can regard the simple perturbation-type *Rotations*: Matching two images, that have been turned by 20° will obviously have less impact on a fingerprint matcher’s performance, than when matching a probe image rotated by 20° and a gallery image in its original orientation.

On the other hand, regarding for example the StirMark test *Modified Gaussian Filtering*: As further explicated in section 5.6, the application of this test causes certain types of fingerprint images to “thin-out” and details are gradually being lost. When matching two such perturbed images, the result would very much depend on which particular areas of the finger imprints are being lost and which ones remain – if, per coincidence, corresponding areas of the fingerprint are still present in both, the probe and the gallery image, the matching result might be good, however, if only differing areas remain, the result will undoubtedly be bad, even though the intensity level of the perturbation would be the same.

- Second, in most real-life fingerprint application scenarios one can assume, that during the enrollment phase, certain means of quality control will be implemented, assuring a minimum level of clarity, distinctiveness, etc. for those fingerprint images, that will be registered in the gallery. (For example, this might be a trained person, monitoring the enrollment process and visually checking the image quality, or it might be the fingerprint application itself, evaluating the acquired imprint and, if necessary, requesting the user to present the finger once more to the acquisition-device). As such it is my belief, that the experiments in present work correspond better to the actual characteristics of real-life fingerprint application scenarios, when in the matching process, a fingerprint image, representing the respective gallery image, is employed without any additionally introduced perturbations.

Chapter 3

Fingerprint Verification Contest and Test Data

The Fingerprint Verification Competition (FVC) was a contest held in the years 2000, 2002, 2004 and 2006, comparing fingerprint matching algorithms of up to 53 (in 2006) different developing parties. The FVC was organized by the Biometric System Laboratory of the University of Bologna [35], the Pattern Recognition and Image Processing Laboratory of the Michigan State University [22], the Biometric Test Center of San Jose State University [32] and of late the Biometrics Research Lab - ATVS of the Universidad Autonoma de Madrid [34].

The aim of the FVC was to provide developers a “common benchmark” to compare their matching algorithm’s performance to that of others, as well as to gain an overview of the “state-of-the-art” in contemporary fingerprint recognition. The matching algorithms entered in the competition were coming from the academic field, from commercial organizations and from several independent developers. Further information on the Fingerprint Verification Contests can be found on the respective websites [7, 8] and in the detailed analysis of the results of the FVC2004 [21].

The characteristic of the FVC of being a means for unambiguously comparing different fingerprint matching algorithms, fits very well with the aim of present work, to study the impact of perturbations in fingerprint images, artificially introduced via the StirMark suite, on the matching results of various types of fingerprint matchers. Hence I chose to adopt test data and test procedure of the FVC for the experiments in present work (see following sections for details).

3.1 Test Data

For the experiments in present work we will be using the example fingerprint images from the FVC2004, as by the time the experiments were conducted, the “latest” fingerprint images, i.e. those of the FVC2006 were not freely available.

The complete test data of the FVC2004 consists of four separated fingerprint image databases, depending on the way, the images were acquired: three databases contain orig-

inal finger imprints scanned using different sensors and one database contains artificially generated fingerprint images (see Table 3.1 for details). Example images of each of the four databases can be seen in Figures 3.1a to 3.1d. All images are provided in uncompressed TIFF format.

	Sensor Type	Model	Image Size	Resolution
DB1	Optical	CrossMatch <i>V300</i>	640×480	500 dpi
DB2	Optical	Digital Persona <i>U.are.U 400</i>	328×364	500 dpi
DB3	Thermal Sweeping	Atmel <i>FingerChip</i>	300×480	512 dpi
DB4	Synthetic Generator	<i>SFinGe v3.0</i> [35]	288×384	500 dpi

Table 3.1: Details on the fingerprint images in the four different databases of the FVC2004 test data.

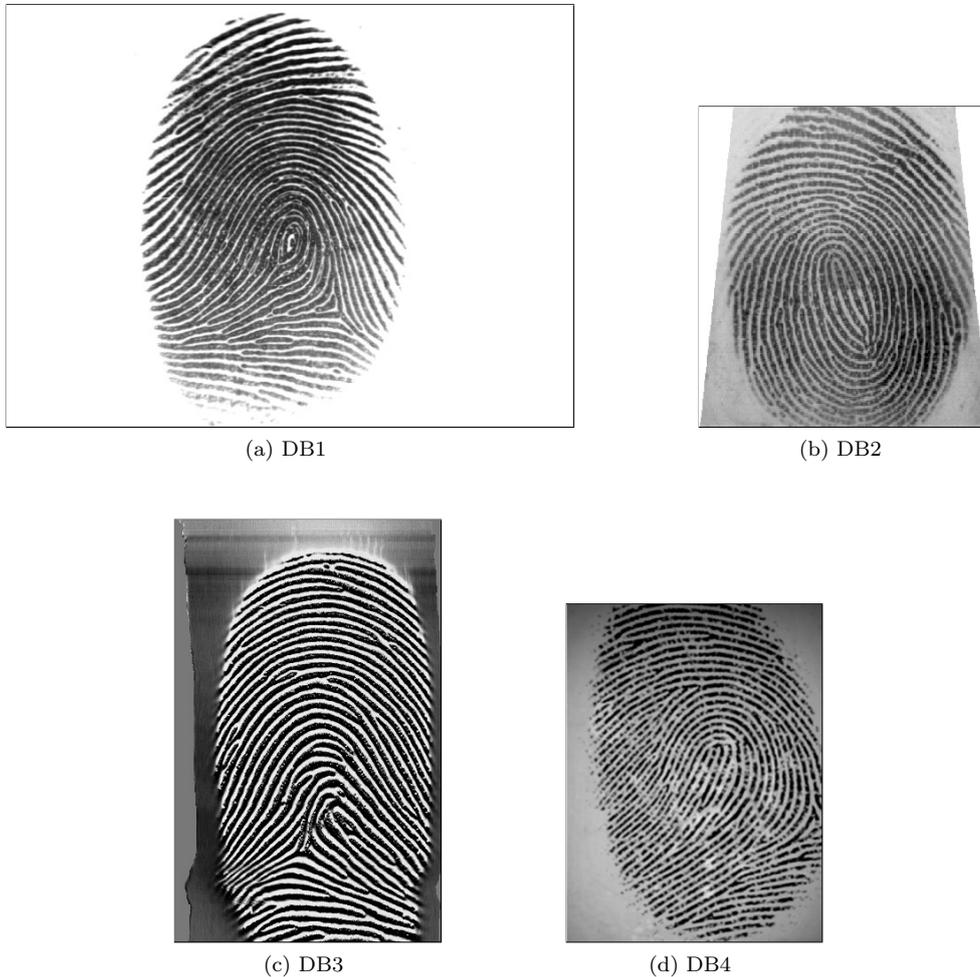


Figure 3.1: Examples fingerprint images from the databases DB1-DB4 of the FVC2004 test data set (scaled to $\frac{1}{3}$ of original size).

One explicitly stated criteria for the fingerprint images within each database was, for them

to be “difficult”. Therefore, in regard to the three databases containing scanned original fingerprints, the acquiring process was structured, in outlines, as follows:

- Per sensor (database) a group of 30 randomly chosen volunteers was assigned.
- Three scanning sessions were held, a minimum of two weeks between them. Each time the volunteers were to presented alternating forefinger and middle finger of both hands (ergo 12 imprints in total).
- Following instructions were given during image acquisition (the images show examples of especially difficult cases. Images were scaled to $\frac{1}{4}$ of original size):
 - Imprints 1, 2: Slightly different vertical position. (Fig. 3.2)

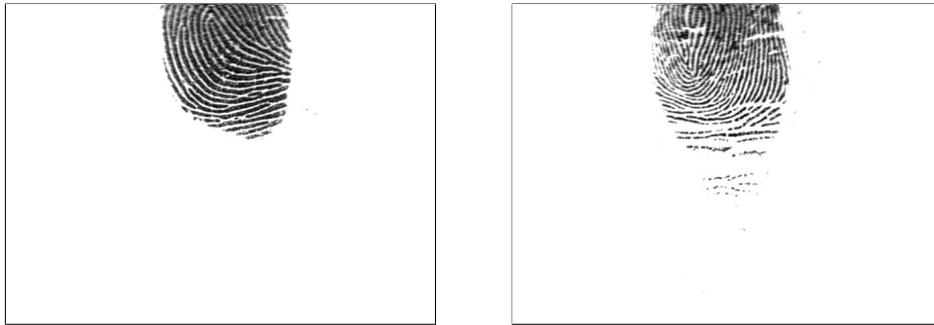


Figure 3.2: Criteria: Different vertical position.

- Imprints 3, 4: Alternate high and low pressure. (Fig. 3.3)

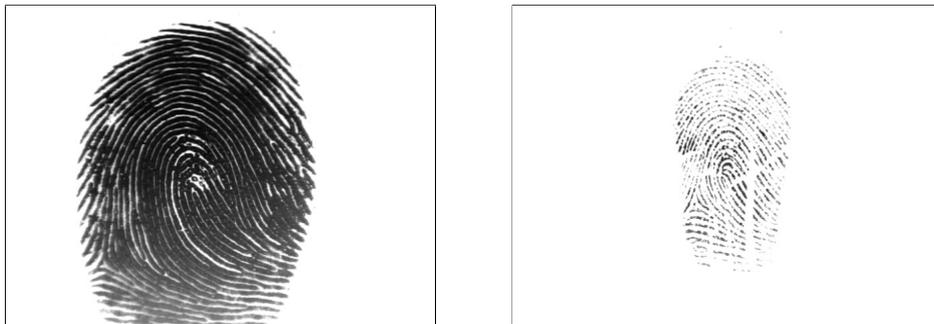


Figure 3.3: Criteria: High and low pressure.

- Imprints 5, 6: Exaggerate skin distortion. (Fig. 3.4)
 - Imprints 7, 8: Exaggerate rotation. (Fig. 3.5)
 - Imprints 9, 10: Dried finger. (Fig. 3.6)
 - Imprints 11, 12: Moistened finger. (Fig. 3.7)
- No image quality control was made, nor the sensor platens systematically cleaned.

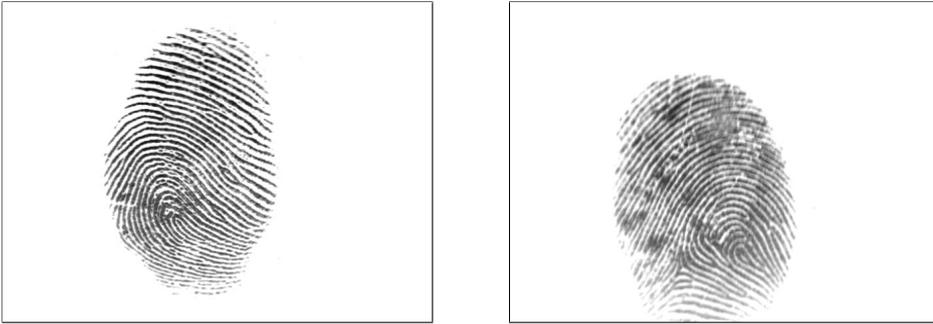


Figure 3.4: Criteria: Exaggerate skin distortion.

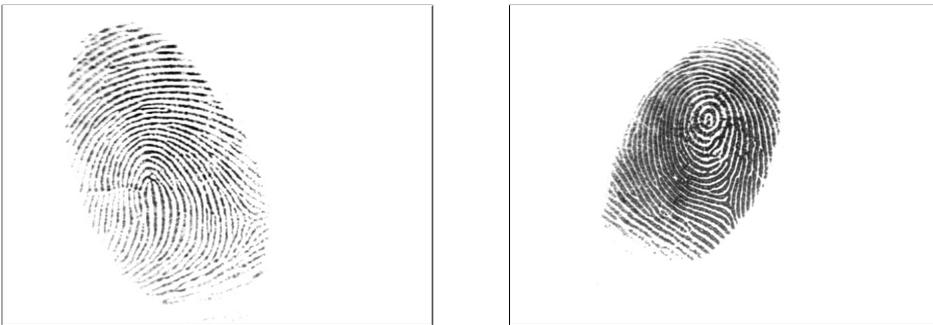


Figure 3.5: Criteria: Exaggerate rotation.

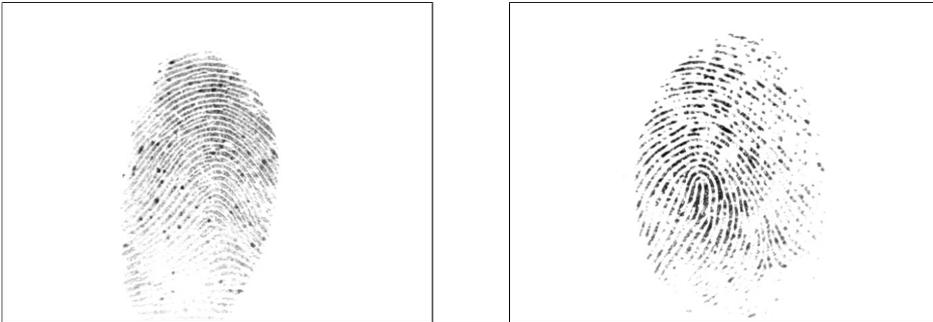


Figure 3.6: Criteria: Dried finger.

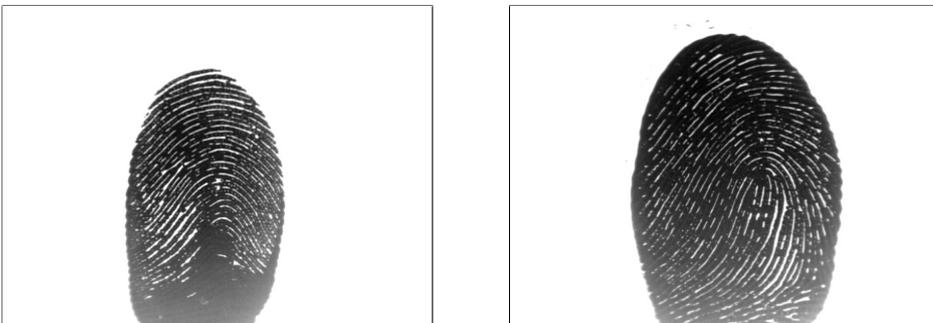


Figure 3.7: Criteria: Moistened finger.

This amounts to 120 fingers, à 12 impressions per database. Out of these 1440 fingerprint images, 880 were then selected for the final test database, which itself was split into sets A and B.

- Set A consists of 100 fingers (IDs 1 to 100) with 8 imprints each. It is used exclusively in the final experiments.
- Set B consists of 10 fingers (IDs 101 to 110) with 8 imprints each. This set is used for preliminary tests, aiming to determine an optimal parameter set-up for each database.

This separation into sets A and B is done to avoid parameter over-fitting.

3.1.1 Notes on FVC2004 Test Data Usage in Present Work

Limitation to DB1, DB2 and DB3

As a consequence of the large amount of different test runs that have to be conducted per fingerprint image database in course of the experiments of present work – Number of StirMark perturbation types \times Number of intensity levels – and the fact, that my implementations of not-minutiae-based fingerprint matching algorithms (see section 4.2 for further details) have an extremely bad performance in terms of processing speed, I decided to only use 3 of the 4 databases included in the FVC2004 test data. Given the fact, that databases DB1–DB3 contain fingerprint images scanned from real fingers, I chose those over the synthetically generated ones in DB4.

DB1 with Image Size of 512×512

Another alteration that I had to introduce to the matching process in consequence of the poor processing speed of the not-minutiae-based fingerprint matching algorithms is, that images of DB1 are not used in their original image size of 640×480 , but instead in a size of 512×512 . This is a concession to the fact that both not-minutiae-based algorithms employed in present work, use a two-dimensional Fast Fourier Transform at certain points. As the 2D FFT algorithm applied only processes images having dimensions of a power of 2, fingerprint images of DB1 would have automatically been extended to a size of 1024×512 for the transform. Consequently the transformation results would likewise be of this largely extended size, causing the subsequent, computationally expensive processing steps of the not-minutiae-based matching algorithms to take enormous amounts of additional processing time. Hence I settled on adjusting the size of the original input fingerprint images from 640×480 to 512×512 . This operation is performed in two steps:

1. On the left and on the right side of the image a stripe of 64 pixels width is removed, leaving an area of 512×480 , cut out from the center of the original image.
2. Top and bottom are extended by 16 pixels with the original image's background gray-level of 255 (white).

Cropping a fingerprint image, as in step 1, holds the risk of losing valuable image information. A scan through the complete database DB1 showed, that this was the case in only 4 images and there only very small parts of insignificant boarder sections got cut off.

3.2 Test Procedure

The procedure for performance evaluation is basically the same in all Fingerprint Verification Contests, from 2000 to 2006. Following I will briefly reflect the two types of tests, conducted in the FVC:

Genuine Tests These tests are carried out to establish the *False Non Match Rate FNMR*. Each sample is matched against the remaining samples of the same finger. Symmetric matches are not executed to avoid correlation in the scores (i.e., if fingerprint image g is matched against h, then h will not be matched against g). Using the test data of the FVC2004 [7], per database with a width of 100 fingers and a depth of 8 imprints per finger, the total number of genuine tests is:

$$\frac{8 \times 7}{2} \times 100 = 2800.$$

Impostor Tests These tests are carried out to establish the *False Match Rate FMR*. The first imprint in each finger is matched against the first imprints of all remaining fingers. Here likewise, symmetric matches are not executed to avoid correlation effects. Using the test data of the FVC2004, the total number of impostor tests is:

$$\frac{100 \times 99}{2} = 4950.$$

Another point, that is adopted from the test procedure of the FVC in present work: Together with an executable version of their fingerprint matching algorithms, contestants are given the possibility to provide one configuration file per database, containing database-specific parameter settings. As the not-minutiae-based fingerprint matching algorithms employed in present work are strongly dependent on the actual gray-level representation of the ridge and furrow structure within a fingerprint (see section 4.1 for details), adjusting certain matching parameters according to the type of fingerprint images to be tested, is a means of clearly improving the respective matching performance. Details on the database-specific settings of the applied not-minutiae-based matchers can be found in Sections “FVC2004-Database Specific Differences” on pages 49 and 75.

Chapter 4

Fingerprint Matchers

As already mentioned in the introduction in the course of present work I want to study the influences of various types of perturbations (distortions, noises, etc.) commonly appearing in fingerprint images, on the matching results. In this regard it is especially interesting to compare the performance of different fingerprint matching techniques – does a certain method have particular strong points or weaknesses when dealing with a specific type of perturbation? Does it maybe have advantages over other methods? These are some of the questions, that I will examine in the course of my experiments.

4.1 Types of Fingerprint Matchers

The different approaches to fingerprint matching can best be described, when first looking at several characteristics, that can be determined in a fingerprint and that can be utilized by a matching algorithm to establish distinctive discriminative features, representing a single fingerprint (please also see Figure 4.1 for reference):

- Looking at a fingerprint on a global level, the overall ridge flow structure can be perceived, as well as the so called *singular points* [13, 17] therein. Generally three categories of singular points can be found: *loops* and *whorls*, characterized by areas of high curvature and *deltas*, representing a specific triangle-shaped ridge pattern common to most fingerprints. (Occasionally, like in [17], in the analysis a whorl is substituted by two facing loops).
- Going into more detail, looking at the ridge and furrow structure in the fingerprint from a more localized point of view, then characteristics like orientation and frequency of the local ridge flow can be observed. The frequency being derived from the inter-ridge spacing in the ridge pattern of the respective area, while the orientation represents its general direction.
- On a local level then, details of the ridge shapes themselves can be determined. The most important ones and to my knowledge the only ones used for fingerprint matching, are *ridge endings* and *ridge bifurcations* which are subsumed in the term *minutiae*. A ridge ending is a point, where a ridge simply runs out, ends. A ridge bifurcation is a formation, comparable to a Y-junction, where a single ridge splits into two separate

ridges. Further ridge details concerning the shape are for example islands (very short, almost point-shaped ridges), lakes (oblong holes in the interior of a ridge) and small holes.

- Looking even a step closer at a fingerprint, then also diminutive intra-ridge features are detectable – the finger’s sweat pores. Yet despite being a highly distinctive characteristic, individual to each finger, the pattern of sweat pores is not well suited for most applications, as it can only be determined in sufficiently good quality in high-resolution fingerprint images of for example 1000dpi and above. As in present work we exclusively deal with fingerprint images of 500dpi resolution, sweat pore related algorithms are not further considered.



Figure 4.1: Overview over different features detectable in a fingerprint image.

Now depending on what type of features, what aspects of a fingerprint a matcher takes into account when establishing the resemblance of two fingerprint images, its approach will basically be one of the following classes [13]:

Correlation-Based Matcher These approaches use the fingerprint images in their entirety. Here the global ridge and furrow structure of each fingerprint is decisive, when the images are correlated at different rotational and translational alignments. Optionally different image transform techniques may be utilized for that purpose. The final matching score is then related to the correlation values.

Ridge Feature-Based Matcher Matching algorithms in this category likewise deal with the overall ridge and furrow structure in the fingerprint, yet in a localized manner. Characteristics like local ridge orientation or local ridge frequency are used to generate a set of appropriate features representing the individual fingerprint. Matching two fingerprints then merely consists of comparing their corresponding feature sets.

Minutiae-Based Matcher The set of minutiae within each fingerprint is determined and stored as list, each minutia being represented (minimum) by its location and direction. The matching process then basically tries to establish an optimal alignment between the minutiae sets of two fingerprints to be matched, resulting in a maximum number of

pairings between minutiae from one set with compatible ones from the other set. The higher the number of these pairings, the higher the similarity between the respective fingerprints.

The minutiae-based approach is the most common and most widely used method for fingerprint matching.

Hybrid Techniques There also exist fingerprint matching algorithms, that combine some or all of the above listed techniques, aiming to combine the particular strong points of each individual approach into a single, more precise matcher. For example [30] combine a minutiae-based rotational and translational fingerprint alignment with a ridge feature-based matching method.

4.2 Matchers in Present Work

In the experiments of present work, I will use an array of five different fingerprint matchers to generate matching results for the StirMark altered fingerprint images. Three of the matchers are minutiae-based, one is correlation-based and one ridge-feature based.

From the set of minutiae-based matchers, two are commercial software products (see sections 4.3 and 4.4) and one is freely available as open source program (see section 4.5). As for the not-minutiae-based fingerprint matching methods, I was not able to find adequate existing realizations, that were freely or at least easily available. For that reason I chose to create own implementations based on already proposed algorithms, covering the other two minutiae-independent fingerprint matching approaches (see sections 4.6 and 4.7).

In the following three sections I will briefly introduce the minutiae-based fingerprint matchers. As far as possible I will also mention some details on their respective matching procedures, as it is my believe, that a deeper insight in the operating mode of a matcher will also give a better understanding as to how and why it will react to certain perturbation types in fingerprint images.

In sections 4.6 and 4.7 I will then turn to the not-minutiae-based matchers and my realizations in more detail: First I will give a basic overview over the respective matching algorithm and its specific matching method. Then I will cover several aspects of my implementation, including the changes compared to the original algorithm, I introduced. Concluding each of the two sections, I will state the matcher setup for the experiments, especially in regard to settings depending on the particular FVC2004 fingerprint image database in use.

Finally, I want to make a note, concerning my realizations of the not-minutiae-based matching algorithms: The implementations of the proposed algorithms were created to the best of my knowledge and abilities, following the information given in the according papers as closely as possible. My added changes and improvements are based on experiments conducted with my implemented versions during and after the creation process. Yet despite all these efforts, it can still very well be, that the original implementations, created by the authors of the algorithms, exhibit superior matching performances than mine. Hence I do not claim, that my matching results represent the best achievable results using the proposed matching algorithms.

4.3 Griaule Biometrics – Fingerprint SDK 2009

The first in the list of minutiae-based fingerprint matchers, that I use in the experiments of present work, has been developed by *Griaule Biometrics* [9] and is applied in form of the *Fingerprint SDK 2009* (in particular the Java version of the SDK). For the sake of simplicity, I will henceforth, throughout this document, refer to this matcher as *GrFinger*.

Griaule Biometrics is based in San Jose, California, USA, with a research and development headquarter in Campinas, São Paulo, Brazil and links to the State University of Campinas.

Researchers associated with Griaule Biometrics, that are named on [9]: Pascual Figueroa Rivero, Luis Mariano Del Val Cura, Gustavo Sá. Research advisers: Roberto de Alencar Lotufo, Neucimar Jerônimo Leite.

The Griaule fingerprint recognition algorithms won in the “Open Category”, section “Average results over all databases” of the Fingerprint Verification Contest 2006 [8], with having the best Average EER.

4.3.1 GrFinger Algorithm

Regarding the fingerprint matching algorithm, I quote the information as it is given on [9]:

- The fingerprint is acquired from a fingerprint scanner.
- Image is improved (better contrast and distinctness).
- Noise and defects are eliminated.
- Fingerprint features are detected and analyzed.
- Minutiae are identified.
- Fingerprint search on database is made based on some measures; so polygons are determined connecting 3 minutiae. Thus, internal angles, sides and each minutia angle are computed. These measures are invariant to rotation and translation.

This method allows that a desired fingerprint can be localized on database even with position variation (displacement and rotation) in relation to the found fingerprint.

Concerning the robustness of the matching algorithm, Griaule Biometrics claims: “Our mature technology provides for the best recognition rates overcoming the real-life challenges such as rotation, scratches, low-contrast, distortion, dirt, different resolutions, different image sizes, ink-and-paper, digitized fingerprints.”

4.3.2 Further Notes on Application in Present Work

In course of the experiments of present work, the GrFinger matcher is used with a rotation alignment range of $\pm 180^\circ$ and a range of $\pm 20^\circ$. Latter setting provides for the results to be better comparable with the results of the two not-minutiae-based fingerprint matching algorithms that I created an implementation for, whose rotation alignment is also set to $\pm 20^\circ$. (Please refer to sections 4.6 and 4.7 for details).

4.4 VeriFinger

The second minutiae-based fingerprint matcher, that is applied in present work, is called *VeriFinger* and is developed by *Neurotechnology* [24]. For integration in PC- and web-based application this matcher is currently available in form of the *VeriFinger SDK 6.2*.

Neurotechnology, until 2008 known under its former name Neurotechnologija, is based in Vilnius, Lithuania and released the first version of VeriFinger in 1998.

In the Fingerprint Verification Contest 2006 [8], in the “Open Category”, section “Average results over all databases” VeriFinger ranked 7th and was additionally having the best Average Zero FMR results among the contestants.

4.4.1 VeriFinger Algorithm

As for the algorithm, Neurotechnology gives the following information in [24]: “The VeriFinger algorithm follows the commonly accepted fingerprint identification scheme, which uses a set of specific fingerprint points (minutiae) along with a number of proprietary algorithmic solutions that enhance system performance and reliability”

Concerning the robustness of the matching algorithm, Neurotechnology states, that the VeriFinger algorithm is tolerant to fingerprint translation, rotation and deformation. Noises, ridge ruptures and stuck ridges are being eliminated to make minutiae extraction reliable. Identification is said to be possible even if the gallery template and the query fingerprint only have 5 -7 similar minutiae.

4.4.2 Further Notes on Application in Present Work

The template extraction procedure, as well as the fingerprint verification engine of VeriFinger both provide an individual threshold that allows to regulate the minimal quality required of the generated template and the matching result respectively. To allow for a fair comparison between the results of VeriFinger and those of the other fingerprint matchers regarded, both threshold are set to the minimum level of 0 in course of the experiments of present work.

Further, cases where templates could not be created by the extraction procedure, due to problems like “too few minutiae” in the respective fingerprint image, are regarded equal to empty templates. Hence, when referenced during the matching process, the matching score will automatically be 0.

4.5 bozorth3

The third minutiae-based matcher in use is *bozorth3*. It is part of the *NIST Biometric Image Software* (NBIS) package [23], developed by the *National Institute of Standards and Technology* (NIST) for the Federal Bureau of Investigation (FBI) and Department of Homeland Security (DHS). *bozorth3* is a “modified” version of a matcher originally written by Allan S. Bozorth.

4.5.1 bozorth3 Algorithm

Basically the bozorth3 program reads two minutiae files and computes their matching score. The input files each contain a list of minutia points found in the corresponding fingerprint, with a point being represented by location (x-, y-coordinates), direction and quality information.

Minutiae detection with mindtct

In order to detect minutiae in a fingerprint image and to generating a format compliant minutiae file, the program *mindtct* can be used, which is likewise part of the NBIS package. This is also the way chosen in present work.

The mindtct algorithm itself is comprised of the following processing steps (for further details please refer to [37]):

- Generate Image Quality Maps
 - Direction Map, representing areas in the fingerprint image with sufficient ridge structure; recording general ridge orientation.
 - Low Contrast Map, to localize areas for which no reliable minutiae information can be determined.
 - Low Flow Map, marking blocks that could not initially be assigned a dominant ridge flow, due to low quality, etc.
 - High Curve Map, marking high-curvature areas, like delta and core regions as unreliable for minutiae detection.
 - Quality Map, integrating the information of the previous maps.
- Binarize Image, using a rotated window scheme.
- Detect Minutiae, scanning the binary image, identifying specific localized pixel patterns.
- Remove False Minutiae, like islands, lakes, in regions of poor quality, etc.
- Count Neighbor Ridges, reporting up to 5 nearest neighbors.
- Assess Minutiae Quality, a consolidation of quality map information and pixel intensity statistics like mean and standard deviation.
- Output Minutiae File

Matching with bozorth3

According to [36], the bozorth3 algorithm is designed to be rotation and translation invariant and consists of three major steps:

Construct Intra-Fingerprint Minutia Comparison Tables Created for each fingerprint to be matched. Per minutia within the fingerprint the relative measurements to all others in the same fingerprint are listed.

Construct an Inter-Fingerprint Compatibility Table The comparison tables of two fingerprints to be matched are examined to find “compatible” entries.

Traverse the Inter-Fingerprint Compatibility Table The set of compatible entries is regarded as tree and is traversed, starting at various points. Clusters of compatible entries are built and themselves combined further. At the end the match score is derived from the number of linked compatibility entries. The higher, the better the match.

4.5.2 Further Notes on Application in Present Work

According to [37], mindtct can take the following file types as input: standard compliant ANSI/NIST-ITL 1-2007, WSQ, JPEGB, JPEGL and IHEAD. In present work preliminary tests were made with fingerprint images in WSQ and JPEGL (lossless JPEG) format. As expected, bozorth3 produced clearly better matching results, when working with minutiae files generated from lossless JPEG images. Furthermore considering that the other fingerprint matchers applied, take uncompressed TIFF images as input, it was decided to only apply fingerprint images in lossless JPEG format for bozorth3 tests in course of the final experiments.

4.6 Phase Only Correlation Matcher

The first non-minutiae based fingerprint matching algorithm that I created an implementation for, to include it in the array of fingerprint matchers used in the experiments of present work, is based on [19]. Therein Ito et al. refer to the the known fact that there are a considerable number of people who can not be identified by means of feature-based fingerprint matching algorithms, due to various special skin conditions on their fingertips. The magnitude of this group of people varies over race, sex, age, job groupings, etc. but it is said to be up to 5% of the respective population. To address this problem Ito et al. propose a matching algorithm using a Phase-Only Correlation function. A technique that has been used in image registration tasks for computer vision applications, as for example in [2] or [18], but can also be effectively used in fingerprint matching scenarios, as is for example discussed in [10].

Consider two $N_1 \times N_2$ images $f(n_1, n_2)$ and $g(n_1, n_2)$ with the index ranges (as defined by Ito et al.) being $n_1 \in \{-M_1, \dots, M_1\}$ and $n_2 \in \{-M_2, \dots, M_2\}$, with $M_1, M_2 > 0$ and further $N_1 = 2M_1 + 1$ and $N_2 = 2M_2 + 1$.

The 2D Discrete Fourier Transform (DFT) of image $f(n_1, n_2)$ is defined in [19] by

$$F(k_1, k_2) = \sum_{n_1, n_2} f(n_1, n_2) W_{N_1}^{k_1 n_1} W_{N_2}^{k_2 n_2} \quad (4.1)$$

with $k_1 = -M_1, \dots, M_1$ and $k_2 = -M_2, \dots, M_2$, further $W_{N_1} = e^{-j \frac{2\pi}{N_1}}$ and $W_{N_2} = e^{-j \frac{2\pi}{N_2}}$, and the operator \sum_{k_1, k_2} denoting $\sum_{k_1=-M_1}^{M_1} \sum_{k_2=-M_2}^{M_2}$. $G(k_1, k_2)$, representing the 2D DFT of image $g(n_1, n_2)$, is likewise defined.

Using the Fourier transforms of the images, the normalized cross spectrum (or cross-phase

spectrum) is calculated:

$$\hat{R}_{FG}(k_1, k_2) = \frac{F(k_1, k_2)\overline{G(k_1, k_2)}}{|F(k_1, k_2)\overline{G(k_1, k_2)}|} \quad (4.2)$$

The phase-only correlation function is then defined as the 2D Inverse Discrete Fourier Transform (2D IDFT) of the normalized cross spectrum:

$$\hat{r}_{fg}(n_1, n_2) = \frac{1}{N_1 N_2} \sum_{k_1, k_2} \hat{R}_{FG}(k_1, k_2) W_{N_1}^{-k_1 n_1} W_{N_2}^{-k_2 n_2}. \quad (4.3)$$

4.6.1 Properties of the Phase-Only Correlation Function

Ito et al. state, that the following properties inherent to the POC function make it very well suited for the use in fingerprint matching:

High discrimination capability As opposed to the normal correlation function, the phase-only correlation produces a sharp and distinct peak, when two images are mostly similar, however if the images are different, the correlation peak drops significantly. In the special case that the two images are identical, the POC function $\hat{r}(n_1, n_2)$ even turns into the Kronecker's delta function $\delta(n_1, n_2)$ with a value of 1, if $n_1 = n_2$ and 0 else.

Shift invariance This property follows directly from the shift invariance of the underlying Fourier transform. Ito et al. show this as follows:

Let $g_1(n_1, n_2)$ be the shifted version of the original image $g(n_1, n_2)$ defined as

$$g_1(n_1, n_2) = g(n_1 + \tau_1, n_2 + \tau_2), \quad (4.4)$$

with (τ_1, τ_2) being the displacement. The POC function $\hat{r}_{fg_1}(n_1, n_2)$ between $f(n_1, n_2)$ and $g_1(n_1, n_2)$ will then be given by

$$\hat{r}_{fg_1}(n_1, n_2) = \frac{1}{N_1 N_2} \sum_{k_1, k_2} e^{j\{\theta(k_1, k_2) - \frac{2\pi\tau_1 n_1}{N_1} - \frac{2\pi\tau_2 n_2}{N_2}\}} W_{N_1}^{-k_1 n_1} W_{N_2}^{-k_2 n_2} \quad (4.5)$$

$$\cong \hat{r}_{fg}(n_1 + \tau_1, n_2 + \tau_2). \quad (4.6)$$

(The same definitions as for Equation 4.1 apply.)

The above equations show, that for an image shifted by (τ_1, τ_2) the correlation peak will be shifted likewise, while on the other hand the value of the peak is not being influenced by the translation.

This property of shift invariance of the POC function is important for the fingerprint matching algorithm of Ito et al. mainly for two reasons:

First of all the value of the correlation peak is directly related to the matching score, therefore it must not be influenced in any way by the relative position of the two fingerprint images that are being matched.

Second, the fact that a displacement of one of the two regarded images relative to the other leads to a displacement of the correlation peak by the same amount will be used to establish the translational alignment for the images. For more details on the translational alignment see Section "Displacement Alignment" on page 40

Brightness invariance This property of the phase-only correlation function can likewise be deduced from the properties of the underlying Fourier transform:

Let $g_2(n_1, n_2)$ be the brightness-scaled image of $g(n_1, n_2)$, given by

$$g_2(n_1, n_2) = \alpha g(n_1, n_2), \quad (4.7)$$

with the scaling factor $\alpha \in \mathbb{R}, \alpha > 0$. The 2D DFT $G_2(k_1, k_2)$ of $g_2(n_1, n_2)$ is then given by

$$\begin{aligned} G_2(k_1, k_2) &= \sum_{n_1, n_2} \alpha g(n_1, n_2) W_{N_1}^{k_1 n_1} W_{N_2}^{k_2 n_2} \\ &= \alpha G(k_1, k_2). \end{aligned} \quad (4.8)$$

Now calculating the normalized cross spectrum \hat{R}_{FG_2} we get:

$$\begin{aligned} \hat{R}_{FG_2} &= \frac{F(k_1, k_2) \overline{G_2(k_1, k_2)}}{|F(k_1, k_2) G_2(k_1, k_2)|} = \frac{F(k_1, k_2) \overline{\alpha G(k_1, k_2)}}{|F(k_1, k_2) \alpha G(k_1, k_2)|} \\ &= \frac{\alpha F(k_1, k_2) \overline{G(k_1, k_2)}}{\alpha |F(k_1, k_2) G(k_1, k_2)|} = \hat{R}_{FG} \end{aligned} \quad (4.9)$$

So as the normalized cross spectra of the the brightness-scaled image and of the original image, \hat{R}_{FG_2} and \hat{R}_{FG} , are shown to be equal, it follows that also their 2D IDFTs, i.e. their phase-only correlations, are equal. This implies that the POC function is not influenced by brightness change.

At this point I would like to add a comment to the above stated property of brightness invariance about an observation, that is also of influence when dealing with the StirMark perturbations of fingerprint images, discussed in chapter 2 and more specific in sections 5.2 and in 5.6:

As shown above, it is obviously true, that on a theoretical level the POC function is not influenced by a brightness-scaling of one or even of both of the input images. This situation changes though, when dealing with an application scenario, where the range of brightness values that image data can assume is limited – most common in fingerprint applications is the usage of 8bit gray-scale images with gray-levels ranging from 0 to 255 as input-images, for storage purposes, etc. In this case the assumption, that the brightness-scaled image $g_2(n_1, n_2)$ can be represented as $\alpha g(n_1, n_2)$ with a uniform positive scaling factor α , will only be true under the precondition that $\forall (n_1, n_2) \in N_1 \times N_2 : \alpha g(n_1, n_2) \leq \text{maxColorlevel}$, with maxColorlevel representing the maximum possible color-level in the image domain (i.e. the value 255 in images with a color-depth of 8 bits per channel). Otherwise, image data elements of $g(n_1, n_2)$ that, when brightness-scaled by α exceed maxColorlevel , can not be represented in the brightness-range of $g_2(n_1, n_2)$ as is, but will be dealt with in an application-dependent manner – in most cases values $\geq \text{maxColorlevel}$ will simply be mapped to maxColorlevel itself. This obviously leads to a change (in this case: loss) of the overall image information. Therefore the assumption $g_2(n_1, n_2) = \alpha g(n_1, n_2)$ is no longer true and consequently the phase-only correlation functions of $g(n_1, n_2)$ and $g_2(n_1, n_2)$ will lead to differing results.

High immunity to noise To demonstrate this property, Ito et al. present a series of diagrams depicting the behavior of the phase-only correlation function when calculated

between an original image and the same images with additive white noise. The figures show, that in the presented cases – ranging from S/N = 20dB to S/N = 0dB – the correlation-peak can be clearly identified, even though its value decreases with increasing noise.

Fingerprint matching performance under noisy conditions is also one of the topics of my StirMark-related experiments. The setup and the results can be found in section 5.3

4.6.2 Band-limited Phase-Only Correlation Function

Based on the experiences and observations made in [10], Ito et al. found that in a 2D DFT of a fingerprint image the most significant information is concentrated in an elliptical frequency band, which is the representation of the respective ridge pattern of the fingerprint, whereas the high frequency areas in the 2D DFT usually have low power and only contain rather meaningless and highly unreliable information. Yet the calculation of the normalized cross spectrum, $\hat{R}_{FG}(k_1, k_2)$, emphasizes those high frequency components, whereby the unreliable parts of the 2D DFT representation strongly effect the outcome of the POC function, leading to reduced distinctiveness of the correlation peak.

In order to improve the matching performance, Ito et al. introduce the notion of *Band-limited Phase-Only Correlation Function*. The idea behind it being, to limit the frequency spectrum regarded for the calculation of the normalized cross spectrum to only those areas, that are strongly related with the actual fingerprint information – especially the inherent elliptical frequency band originating from the ridge pattern – thereby excluding the interfering components in the high frequency areas.

One problem though, that occurs when applying the band-limited POC function for fingerprint matching, is, that it might produce not just one, but multiple correlation peaks. This effect, Ito et al. explain, is caused by elastic deformations in a fingerprint, that can for example occur, when the fingerprint touches the sensor during image acquisition. Each peak will then represent one sub-division of the fingerprint that has been distorted independently from the other areas. So to take this effect into account and thereby also make the matching algorithm robust against elastic deformations, Ito et al. use the sum of the highest P peaks of the band-limited POC function as final matching score.

4.6.3 POCmatcher Algorithm

The fingerprint matching algorithm proposed by Ito et al. in [19] consists basically of four consecutive steps: *rotation alignment*, *displacement alignment*, *common region extraction* and finally *fingerprint matching*. The output of the algorithm is a score value from the interval [0,1], expressing how well the presented probe image matches the specified gallery image. A score of 1 constitutes a perfect match (i.e. two identical images) and a score of 0 stands for no match at all.

Following I will briefly describe each of the four stages of the algorithm. Details concerning my implementation will be discussed in the next section, 4.6.4.

Rotation Alignment

The phase-only correlation function is very sensitive to rotation, therefore it is important to establish a rotational alignment of the two images to be matched, before continuing with subsequent processing steps. Ito et al. even suggest to align the images with an angular resolution of less than $\pm 1^\circ$ to achieve “high-accuracy fingerprint matching”.

To obtain the rotational estimation the algorithm adopts a “straightforward approach”, i.e. it starts by generating a set of rotated versions of the gallery image, with rotations in a predefined angular range and predefined step-width – Ito et al. suggest a range of $\pm 20^\circ$ with a step-width of 1° , employing bi-cubic interpolation. In the next step the POC function is put to use to establish the correlation between each of the rotations of the gallery image and the probe image. The one rotated version of the gallery image that produced the highest correlation peak is then regarded as the rotation-normalized image and will be used for the subsequent matching process.

Displacement Alignment

As already shown on page 36, the phase-only correlation function is shift invariant. Still, in regard to the third processing step of the matching algorithm it is highly beneficial to establish a translational alignment between the probe image and the rotation-normalized version of the gallery image.

As denoted when explaining the shift invariance property of the POC function, we can directly obtain the displacement of one fingerprint image to the other from the location of the correlation peak. Using the amount of displacement in horizontal and vertical directions and the respective sign, both images are then border-extended accordingly (for example the gallery image with an amount of τ_1 on the right and τ_2 on the bottom border, while the probe image with τ_1 on the left and τ_2 on the top border) to normalize the position of the finger imprints in the images.

Common Region Extraction

This step establishes the regions of actual fingerprint information in the normalized gallery image and probe image respectively and then crops both images to the common area of intersection. As the images got rotation- and translation aligned beforehand, now only those image-sections remain, where both images contain fingerprint data. The non-overlapping regions which would lead to uncorrelated noise in the POC function, got cut away.

Fingerprint Matching

This final step follows the procedure outlined in section 4.6.2 to calculate the band-limited phase-only correlation function between the common regions of the gallery and the probe image. For establishing the “inherent frequency band”, used for setting the borders of the band-limitation, only the 2D DFT of the gallery image is examined.

The conclusive matching score is then established by summing up the P highest peaks (Ito et al. suggest $P = 2$) of the band-limited POC function.

4.6.4 POCmatcher Implementation

In this section I would like to give some details on my implementation of the Phase-Only Correlation fingerprint matching algorithm presented by Ito et al. in [19]. Starting with some general information, I will then go through the separate stages of the algorithm, like was done in section 4.6.3 and briefly describe how I implemented it, how I interpreted those details left unexpressed in [19] and where I introduced changes and modifications to the original algorithm, in order to further improve the matching performance, based on the findings in my own preliminary tests and experiments during the implementation phase.

Additional information on the actual matcher-configurations and -settings for application in the experiments of present work can be found in section 4.6.5

General Information

The POCmatcher code was written in Java and last tested with the java-6-openjdk. For image handling I have chosen to use the Java Advanced Imaging (JAI) API [38] (currently in its version 1.1.3). I chose this image processing API mainly for three reasons:

- First, because by the time I started my project, the more commonly used Java Image I/O API – as far as I could find out – did not readily provide methods and interfaces for dealing with images in the TIFF format, yet the fingerprint images in the FVC2004 databases are all TIFF-images.
- Second, because besides the essential methods for image I/O, access and manipulation of the image data as well as the image properties, the JAI API offers an extensive set of interfaces for high-level image processing. For example a large set of point operators for real- and complex-valued images, a very useful set of area operators including an interface for *convolve* operations, geometric operators like *rotate* and *translate*, file-type specific operators and last but not least frequency operators, especially for Cosine- and Fourier transforms, which I heavily applied in the implementations of the POCmatcher and the FCmatcher.
- Third, because aside from the easily accessible and comprehensive API documentation, there exist also a number of sources with explanations and sample code for the usage of the JAI framework and for its specific operations. In particular I would like to mention two online resources, that helped me a great deal in getting acquainted to the framework and understanding how to use it (efficiently):
 - The extensive programmer’s guide *Programming in Java Advanced Imaging* [39]
 - R. Santos’ website *Java Advanced Imaging Stuff* [33] which features a good variety of source code examples.

Despite the fact, that great efforts were made to speed up the overall processing time – on several occasions in the following sections I will refer to this aspect in detail – the Java implementation of the POC fingerprint matching algorithm is very slow, therefore extremely limiting its applicability in real-life scenarios. For example using the current speed-optimized implementation for matching two 512x512 pixels 8bit gray-scale images on an Intel Core 2 Duo processor with 2.53GHz, the whole algorithm cycle – including loading the files with the appropriate JAI operators, creating the set of rotations for the gallery image, etc. . . until

the output of the final matching score – takes about 9.5 seconds. Applying the POCmatcher for a complete test run according to the FVC-regulations (see section 3.2 for details), the overall processing time therefore amounts to more than 12 hours.¹

Rotation Alignment

My implementation of the rotational alignment processing step follows the specifications outlined by Ito et al. as close as possible. The only changes that I introduced here, was to add a cache for the rotated versions of the gallery image and some other minor tweaks to speed up the overall processing time, especially when the matcher is used for batch processing.

When the rotation alignment subroutine is called with the filename of the gallery image used in the current matching process, it first checks, if this filename equals the filename of the last call to the subroutine (which means the last matching process).

If so, then the whole set of gallery-image rotations (i.e. a set of rotated versions of the gallery image, with rotations within a preset angular range, spaced by a preset step-width in degrees) is loaded from cache.

If not so, then the gallery image will be loaded into the system using the JAI *fileload* operator, the set of rotations will be newly created and afterwards also stored in the cache for subsequent calls to the matching process.

While creating the rotated versions of the gallery image, two aspects have to be paid attention to:

- When rotating an image using the JAI operator *rotate* those areas for which there is no original image data to fill them with (i.e. the image corners), will instead be filled with a gray-level of 0, ergo black (see the example in Figure 4.2a). For fingerprint images that do not have black as background color, this added image information (especially the added edges) has a negative impact, when applying the POC function, resulting in a degraded performance in establishing the rotational alignment.

As the JAI framework does not provide the possibility to set the default background color of an image and as other suggested solutions² to this problem did not work without introducing a new set of problems, I had to subject to the “straight forward” version of extending the image by half its respective dimension on each side, using its original background color, rotate this image and then crop it back to its original size (see the example in Figure 4.2b) This, of course, goes not without a minor penalty in processing speed and memory usage. Additionally it also raises the question of how to establish the original background color of a fingerprint image.

Analyzing the images of the FVC2004 databases I observed, that in most cases the four corner pixels hold the original background color. In some cases one of the corners shows foreground data and in very few cases two or more corners are non-background. Looking for a fast solution I decided on following: Insert-sorting the four corner pixels and taking the average of the 2nd and 3rd largest values. This way I eliminate at least

¹Taking 9.5 seconds \times 7750 matches as occur in a FVC-conform test run, the processing time would turn out to be \approx 20.5 hours. The difference to the actual time needed, is the effect of various caching mechanisms, included in the current POCmatcher implementation, that work in favor of processing speed, when using the matcher for batch- processing.

²for example: <http://archives.java.sun.com/cgi-bin/wa?A2=ind0204&L=jai-interest&P=R165926&I=-3>

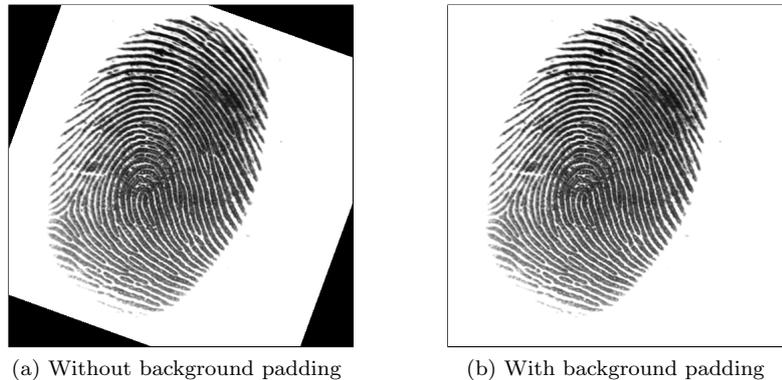


Figure 4.2: Example of rotation by 20° without (a) and with (b) background padding.

one outlier (1st and/or 4th largest value) and in case of 2 or more non-background corner pixels, I limit their influence by averaging.

- Another effect that has to be taken care of, comes into play, when the images being matched are pre-processed using the image enhancer based on Hong et al. [11]. (Please refer to Section “Application of Image Enhancement” on page 47 for motivation and details on the application of said image enhancement in the POC matcher). The image returned by the enhancer is binarized to amplify the distinctiveness of the ridge-structure, with the fingerprint ridge-information being displayed as 1 and the background being displayed as 0 (or any other high-low combination for that matter). Yet when rotating a gallery image, the bi-cubic interpolation being applied by the rotation operator causes especially strong blurring at the edges of the ridge lines. As the edge information of the ridges is vital to the outcome of the POC function, this leads to an impaired accuracy of the rotational alignment function. Therefore, when image enhancement is used together with the POC matcher, I introduce an additional adjusted binarization step at this point, to re-establish clear edges at the fingerprint-ridges.

After having loaded the set of rotated versions of the gallery image, the rotation alignment process enters the second stage, which establishes the “rotation-normalized gallery image”. This is done by looping over the set of rotations and each time calculating the phase-only correlation function between the current entry and the probe image. The one pairing that leads to the highest correlation peak, is regarded as rotation aligned and the corresponding rotated version of the gallery image will then be used in all the subsequent matching steps.

Displacement Alignment

Using the probe image and the rotation-normalized version of the gallery image, the phase-only correlation function is once more calculated, yet this time not the height of the correlation peak is regarded, but its location. Theoretically I could have also memorized the peak location already during the second phase of the rotation alignment step, when repeatedly applying the POC function for establishing the optimal rotated version of the gallery image, but tests during the implementation phase showed, that the overall processing time can be considerably reduced, when substituting the repeated location-determination by this single additional calculation of the POC function.

The rest of the displacement alignment step closely follows the proposal of Ito et al.: The dislocation of the correlation peak from the origin at the image-center is regarded as displacement vector (τ_1, τ_2) . The gallery- and probe images are then border-extended in a suitable way, to achieve their translational alignment: τ_1 is used to extend the images in horizontal dimension, one image on the left, the other on the right border. τ_2 is used for the vertical dimension, here again extending the images on opposite borders respectively.

Common Region Extraction

In a first step the area of actual fingerprint-data in the gallery-image and probe-image are separately established. This is done by initially creating the projections of the image in the horizontal and vertical dimension. Next these projections are scanned, starting from both ends, for those indices, where a difference between the current value and the last value greater than an heuristically determined threshold occurs. These indices mark those respective rows or columns in the original image, where background data ends and foreground data begins.

Special care has to be taken for those cases, where the foreground data already starts at an image border or where no foreground data can be found at all, which is occasionally the case in some experiments of present project (see especially section 5.6).

Having established the fingerprint-data areas of the gallery- and probe image, next both images are cropped to their “common area” – i.e. the intersection of both data areas. In theory both images now only consist of common rotational- and translational aligned fingerprint data, while the non-correlated, result-derogating sections are limited to a minimum. An example for the limitation to the common region can be seen in Figure 4.3.

As a small note: The implementation also takes care, that the common region has even-valued dimensions – if necessary adding a 1 pixel broad column of row in background color to the images – as otherwise the following DFT and IDFT operations would be disproportionately elongated.

Fingerprint Matching

After the gallery- and the probe image have been rotation aligned, displacement aligned and reduced to their common regions of actual fingerprint data, now follows the actual fingerprint matching and score generation.

First the prepared images have to be Fourier transformed. At this point, the image dimensions are in general not equal to a power of 2. The FFT-implementation of the JAI API though can only work with image dimensions being a power of 2, therefore it automatically extends the images accordingly with a value of 0 prior to transformation. To circumvent a possible influence on the matching result, caused by this automatic image extension, especially when executing the inverse transformation on the band-limited phase only correlation function (see further down for explanation) I chose to not apply the JAI FFT and -IFFT operators for this fingerprint matching step, but a FFT package by “teneighty.org”, [40]. The explanation in [40] states “*The main algorithms in use in this package are the Cooley-Tukey (including both the radix-2 and the mixed-radix flavors) algorithm, the Prime Factor algorithm, and Rader’s algorithm*”. The transform-methods in the teneighty.org package are therefore able to handle images of arbitrary dimensions. This freedom though comes with a noticeable penalty in processing speed, as compared to the JAI operators. This speed-issue is also the reason, why I limited the application of the teneighty.org methods to the fingerprint

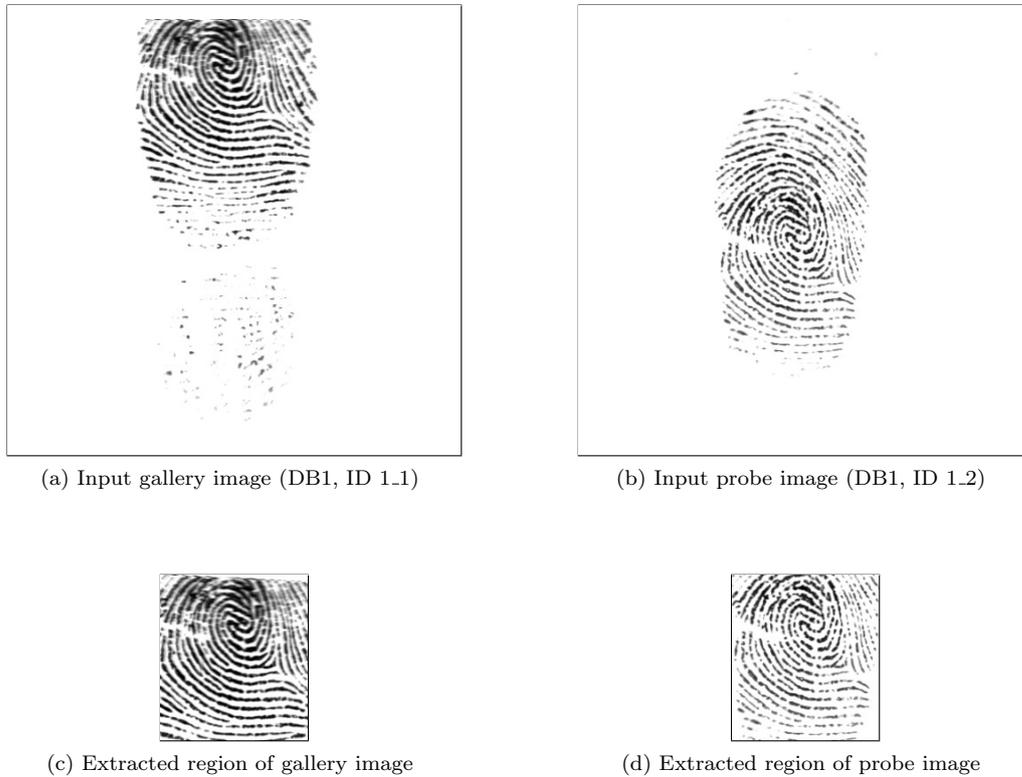


Figure 4.3: Results of the common region extraction processing step. Here common region is set to “intersection”.

matching step alone, instead of using it already in the phase only correlation related calculations, when establishing the rotational normalized version of the gallery image. There I chose to trade the possible slight inaccuracies, caused by the implicit image extension of the JAI transforms, for the minutes of additional calculation time, that the use of the teneighty.org transforms would have brought per matching run.

When the gallery- and probe image have been Fourier transformed, their normalized cross spectrum is created. Contrary to the rotation alignment step, now the matching score is not directly derived from the inverse Fourier transform of the normalized cross spectrum, but at this point the aspect of band-limitation is introduced:

First the limits are established. Therefore the projections in horizontal and vertical dimension of the amplitude spectrum of the gallery image are created and their respective mean values calculated. Each projection is then scanned to find those indices where the accumulated values (first) rise above the related mean value. (As the amplitude spectrum is symmetric, a scan starting from one end of a projection suffices). These indices are regarded as the band-limits.

To restrict the inverse Fourier transform of the normalized cross spectrum to only the frequencies within the established limits, in order to create the *band-limited* phase only correlation, I adopted the straight forward approach, to simply crop the normalized cross spectrum to the area within the limits. The inverse transform is then executed only on this cropped section. For the sake of precision, here again the transform-methods of the teneighty.org

package are applied.

The matching score is finally established by summing up the P highest peaks in the result of the band-limited phase only correlation function. The value P can be set via parameter on instantiation of the matcher object.

Application of Image Enhancement

The original POC fingerprint matching algorithm as proposed by Ito et al. in [19] does not include any image enhancement or any other form of image pre-processing. The fingerprint images are taken 'as is' and passed in their original form to the processing chain.

In the course of implementing the Finger Code fingerprint matcher, I also created a realization of a fingerprint enhancement- and segmentation algorithm based on a proposal by Hong et al. [11] (see section 4.7 for information on the Finger Code matcher and section 4.7.1 for details on the the enhancement- and segmentation algorithm). Thus having this fingerprint image enhancer ready at hand, I also tried to apply it as a pre-processing step to the POC matcher. My question was, if enhancement and segmentation of the fingerprint images prior to processing in the various matching-steps of the POC matcher would show any clear and evident effect on its matching performance. The answer to this question was not at all obvious to me, given that the band-limited phase-only correlation algorithm provides own strategies to cope with varying quality of fingerprint images – foremost the process of band-limitation itself and the high immunity to noise, inherent to the POC function.

Figure 4.4 shows, that application of prior fingerprint image enhancement can indeed have an obvious effect on the overall matching performance. In this example we see the ROC curves for the matching results for fingerprint images of FVC2004 DB2, when tested with the POC matcher once without and once with prior enhancement. As can be seen in the diagram, in this setup the test-run with enhancement clearly brings better results.

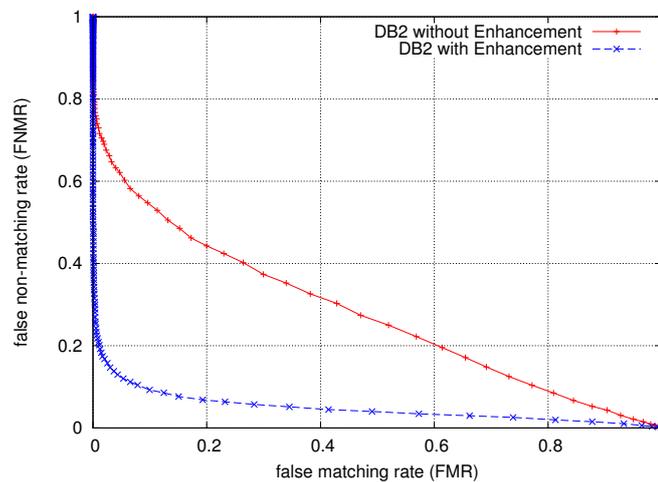


Figure 4.4: Comparison of Receiver Operating Characteristics of the Phase Only Correlation Matcher for fingerprint images of the FVC2004 DB2 data set: once the images are matched in their original form and once fingerprint image enhancement was applied prior to matching. Enhancement clearly improves the results.

Further observations regarding the application of image enhancement as pre-processing step to the POC matcher when dealing with fingerprint images of the FVC2004 databases can

be found in Section “FVC2004-Database Specific Differences” on page 49, under point “Enhancement”.

Batch Runner for FVC Test

In order to conduct the matching test runs following the regulations of the FVC2004 (see chapter 3 for details) using the POCmatcher, I created a dedicated batch runner. Put simple, its purpose is to:

- set up the POC matcher and if necessary the image enhancer used for pre-processing
- loop through the required pairings of gallery and probe images, each time calling the matcher with a respective pair.
- outputting the results into a csv-file.

As input the batch runner receives a configuration file and, if not specified within the configuration file, the location of the probe images for the current test. Currently the batch runner supports a set of 39 different parameters that can be supplied via the configuration file and are used to set and fine-tune the behavior of the batch runner, the POC matcher and the image enhancement in pre-processing. Among them the most important ones are parameters for setting up the location of the gallery and probe images, those for setting up the angular range and step-width of the rotation alignment process and a parameter that specifies the type of image database used (in case of the current work: FVC database 1, 2 or 3). With this last parameter all the other “behavioral” parameters can be set at once to the values I established during my numerous preliminary tests to be optimal for the specific types of images. (Further details on the database-specific settings are given in the following section)

The output of the batch runner is a csv-file containing a result-line for every gallery- and probe image pair matched. The format is *probe_image;gallery_image;matching_score;*

4.6.5 FVC2004-Database Specific Differences

Here a short discussion of those parameters set via the batch runner, that optimize the functionality of the fingerprint enhancer and matcher for working with the respective fingerprint image types in the FVC2004 reference databases.

Sum P Peaks

As mentioned in section 4.6.2 the band-limited phase-only correlation function tends to produce multiple peaks due to the elastic deformations in the fingerprints. Therefore in the proposed algorithm of Ito et al. the sum of the highest P peaks ($P = 2$ suggested) is being used as final matching score.

Regarding the fingerprint images of the FVC2004 databases and using the suggested value $P = 2$, I established in my experiments, that only for DB2 the summation of the highest peaks leads to an improved matching performance. For the databases DB1 and DB3 the opposite is true: The sum of P peaks clearly leads to worse results than when using just the highest peak alone as matching criterion.

Enhancement

As opposed to the application of the image enhancer as pre-processing step in the Finger Code matching algorithm (see section 4.7.1 for details), here, with the POC matcher, a database-dependent adjustment of the segmentation parameters does not lead to an improved matching performance. My rather conservative default settings for the segmentation, which are aimed to just segment the foreground from the background, but not care about low-quality areas in the finger-imprint, bring the best results.

An explanation for this effect seems to be, that when a rather aggressive segmentation also eliminates areas of low fingerprint-quality, this changes the overall shape and appearance of the imprint, also adds additional sharp edges and therefore strongly influences the fingerprints 2D DFT and in consequence also the phase only correlation function. This detraction can be witnessed especially clear, when the removed low-quality areas are internal to the fingerprint. In other words, for the POC matching algorithm it is of advantage, when the segmentation rather works a bit “sloppy”, but therefore leaves the general shape of the imprint untouched.

Still it can not be said, that all databases can be treated equal in respect to image enhancement. As stated in section 4.6.4 the introduction of image enhancement in the processing cycle can lead to a big improvement in matching performance. Regarding the databases of the FVC2004, the positive effect can very clearly be witnessed for DB2 (see also Figure 4.4). Also in DB1 the improvement is obvious, though not as strong as in DB2. (The respective EER values can be found in Table 4.1). DB3 however shows a different behavior: As can be seen in Figure 4.5 for DB3 the matching results are clearly better without prior enhancement of the fingerprint images.

	EER (%) without enhancement	EER (%) with enhancement
DB1	29.52	21.91
DB2	34.75	9.51
DB3	15.13	18.43

Table 4.1: Equal error rates (EER) for Phase Only Correlation Matcher, when matching is performed without or with preceding fingerprint image enhancement

Common Region Extraction Area

As explained in Section “Common Region Extraction” on page 40 the third processing step of the band-limited phase-only correlation algorithm, first establishes the areas of fingerprint-data in the gallery- and the probe image respectively. Then it reduces the images to only the intersection of the data areas, thereby chopping off all uncorrelated sections in the original images.

On one hand out of curiosity, on the other because of noticing that the intersection of the data areas sometimes leads to very small resulting images, I also regarded the application of the union of both data-areas instead.

Using the database specific settings I figured out so far, regarding the sum of P peaks and the application of enhancement, I was trying out both possible common region configurations

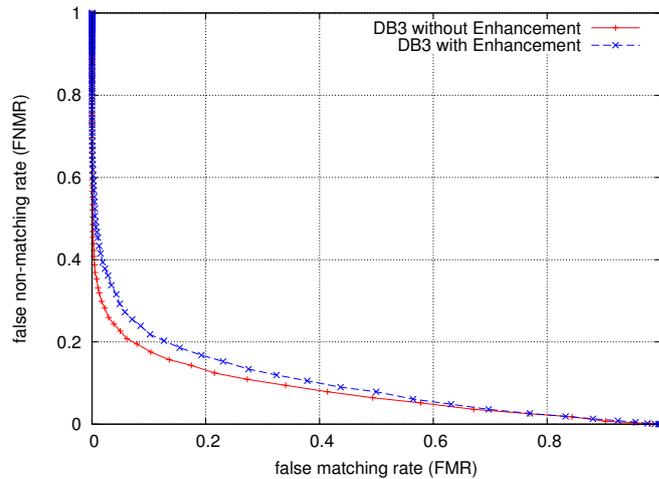


Figure 4.5: Comparison of Receiver Operating Characteristics of the Phase Only Correlation Matcher for fingerprint images of the FVC2004 DB3 data set: once the images are matched in their original form and once fingerprint image enhancement was applied prior to matching. Here enhancement does not improve the results.

– intersection versus union of the respective fingerprint data areas – for each of the three databases. This lead to following, quite interesting observations: For matching the images of DB2 the original common region extraction version (i.e. intersection of the data areas) showed a slightly better performance. On the other hand for databases DB1 and DB3 it was the union-version that outperformed the intersection-version. For DB3 the difference is again rather small, but regarding the results of DB1, the improvement by using the union as common region is quite clearly observable. Figure 4.6 show the two different outcomes: The original version having better results than the union-version for DB2 (4.6b) and vice versa for DB1 (4.6a).

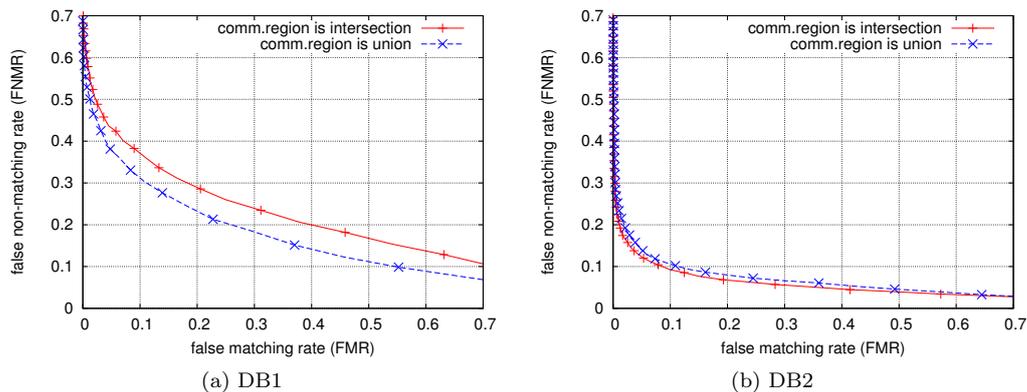


Figure 4.6: Comparison of Receiver Operating Characteristics of the Phase Only Correlation Matcher when defining either the intersection or the union of the actual fingerprint data areas withing the images as “common region”.

4.7 FingerCode Matcher

The second explicit not-minutiae-based fingerprint matcher that I implemented, is basically a realization of the algorithm presented by Ross et al. in [31]. Additionally I also took some inspirations from a later paper by the same authors, [30], and a paper by de Sá and Latufo [5], who propose several improvements to the matching stage of the algorithm in [30].

The common ground that all aforementioned papers relate to, in one way or another, is a method to generate a fixed-length feature vector representing the individual ridge and furrow structure of a fingerprint, called *FingerCode*, introduced by Jain et al. in [14]. Originally developed as basis for a fingerprint classification algorithm, Jain et al. later also put the FingerCode to use in fingerprint matching [15].

The idea behind FingerCode is to apply a bank of 8 Gabor filters to capture the orientation and frequency of the ridges in the fingerprint. The resulting filtered images are then tessellated in a specific way and per cell a distinctive feature representation is created. Thus each feature value captures the local ridge information of its cell, while the global ridge-configuration is recorded by the enumeration of the cells, induced directly by the tessellation.

The algorithm in [14] applies a circular tessellation centered around a previously established core point. The FingerCode is then created by calculating the standard deviation of gray-scales within each cell. This feature vector looks very similar to the IrisCode introduced by Daugman in [4], hence its name.

In [15] instead of the standard deviation, the average absolute deviation from the mean is used to create the FingerCode. For fingerprint matching then the Euclidean Distance between the FingerCodes of the gallery and the query image is established.

In theory the feature vectors are inherently translation invariant, as the tessellation is created around the core point, which is supposed to be identical in imprints of the same finger. Rotation invariance on the other hand is approximated by cyclically rotating the FingerCode features, a rotational step therefore being equivalent to a rotation of 22.5° of the underlying fingerprint image.

This algorithm by Jain et al. generally has some drawbacks though:

- It is dependent on a single core point, which acts as center for the circular tessellation and indirectly as reference point for the alignment. Detecting the core point is not an easy problem [31]. Additionally it might happen, that the actual core point of a fingertip is not even present in the imprint or that it is so close to the image border that large parts of the tessellation would be lying outside the image boundaries. (This might be the case for example, when the imprint was acquired using a small solid-state sensor, or when dealing with latent fingerprints.)
- It only takes into account the circular tessellation, which in general does not cover the entire fingerprint, thus forfeiting potentially valuable information outside the circular area.
- The feature extraction is done on the original input image without any prior enhancement. Therefore the matching performance is highly sensitive to perturbations like noise or brightness variance.

To circumvent these problems, subsequent algorithms substitute the circular with a square

tessellation, covering the whole fingerprint image. Further they present different strategies for alignment of the gallery and probe images, to avoid the dependence on the single core point. [16, 30, 5] for example use additionally gathered minutiae information to establish the translational as well as the rotational alignment. Thereby of course heavily depending on the accurate localization and successful matching of the minutiae information of the respective fingerprints.

As stated in section 1.2 my goal is to find and implement an algorithm that is explicitly not using minutiae data in any stage of the fingerprint matching process, to compare the matching performance of the minutiae matchers listed in sections 4.3, 4.4 and 4.5 to that of essentially different approaches. Therefore I selected an algorithm presented by Ross et al. in [31] for my implementation.

In the following sections, 4.7.1 and 4.7.2, I will give a brief overview over the individual processing steps of the algorithm. At first about the fingerprint enhancement and segmentation method applied and then about the further steps of the matching algorithm itself. In the next sections then I will go into more detail on my implementation of the enhancer (section 4.7.3) and of the matcher itself (4.7.4). My explanation of the FingerCode algorithm and realization will then conclude in stating the differences in individual matcher set-up for the three FVC2004 image databases used in the experiments of present work.

4.7.1 FingerCode Algorithm - Enhancement and Segmentation

As first step the FingerCode algorithm presented by Ross et al. [31] starts by applying image enhancement and subsequently a separate fingerprint segmentation to the supplied fingerprint image. The enhancement method is specialized on fingerprints and was developed by Hong et al. [11]. Its purpose is to reduce the negative influences of noise and brightness variations in fingerprint images on the matching result, as well as to improve the clarity of the ridge-furrow structure.

The enhancement algorithm basically consists of the following five sequential steps: Normalization of the fingerprint image, estimation of the local ridge orientations, estimation of the local ridge frequencies, generation of a region mask and finally filtering the normalized image with a bank of Gabor filters, utilizing all of the information gathered in the previous steps.

Normalization

As quasi pre-processing to the subsequent algorithm steps, this normalization aims to reduce the variations in gray-level values of the original fingerprint image. This is done by pixel-wise adjusting the gray-levels to obtain an image with pre-specified mean and variance. A procedure which does not have any influence on the clarity of the ridges and furrows though. In [11] the mean and variance are set to 100.

Orientation Image Estimation

The orientation image represents the local ridge orientation in the normalized fingerprint image, which is specified for blocks of size $w \times w$ (instead of per pixel). Hong et al. suggest $w = 16$. The according processing steps are as follows:

The gradients in horizontal and vertical direction are calculated for each pixel in the normalized image. For that purpose an arbitrary gradient operator can be applied. (Hong et al. name a Sobel and a Marr-Hildreth operator as examples).

Using the gradient information, the least square estimate of the local ridge orientation in each block is established. The resulting orientation image already corresponds quite well to the actual ridge orientations in the original fingerprint, yet still includes a number of incorrectly estimated blocks. These errors are due to noise or other disturbances in the corresponding blocks of the fingerprint image or happen occasionally, if minutiae or other singular points were present in the respective area.

To deal with the incorrectly estimated orientations, Hong et al. exploit the fact, that local ridge orientation varies slowly in a neighborhood without singular points. They transform the orientation image into a continuous vector field and apply a 2-dimensional low-pass filter with unit integral to smoothen it. A filter size of 5×5 is suggested. An accordingly smooth orientation image is then obtained from this vector field.

Frequency Image Estimation

The normalized fingerprint image and the orientation image are used to create an image representing the local ridge frequency, which is defined in [11] as “*the frequency of the ridge and furrow structures in a local neighborhood [without minutiae or singular points] along a direction normal to the local ridge orientation*”. To calculate the local frequencies, we proceed as follows:

Per block an *oriented window* with size $l \times w$ is defined. Hong et al. propose $l = 32$ and $w = 16$. This rectangular window is centered at the center of the block and its length l lies in a direction orthogonal to the ridge orientation of the block. Within this window the *x-signature* is calculated, which is figuratively speaking the projection of the respective gray-level values of the normalized image onto the length l .

Except for blocks, where minutiae, singular points or disturbances are present in the fingerprint, the x-signature forms a discrete sinusoidal-shape wave with the same frequency as the ridge and furrows in the oriented window. The frequency can therefore be determined, by taking the reciprocal of the average distance between peaks in the x-signature.

Those blocks, where the x-signature did not form a discrete sinusoidal-shape wave, can be identified by inspecting the calculated frequencies. If the value does not lie within a certain range, specific for an average ridge furrow structure, the respective block is marked as invalid. The frequency of an invalid block then has to be interpolated from the frequencies of the neighboring blocks. For this interpolation Hong et al. apply a discrete Gaussian kernel with a mean of 0, a variance of 9 and a size of 7.

As last step of the frequency image generation, outliers are removed, by filtering the image with a 2-dimensional low-pass filter with unit integral and a filter size of 7.

Region Mask Generation

This section deals with the segmentation of a fingerprint image into foreground and background regions, or as Hong et al. put it, recoverable and unrecoverable regions. For that purpose Hong et al. apply a one-nearest neighbor classifier working with block-wise feature vectors containing amplitude, frequency and variance of the respective x-signature. In [31]

though, Ross et al. use a separate segmentation algorithm, that is applied to the image returned by the enhancer. Yet no hint can be found in [31], which algorithm is referenced or how it would function. Then again in [30], where the enhancement algorithm by Hong et al. is used as well, Ross et al. write about their segmentation method: “Segmentation is done by observing the local variation of intensity on the original gray-scale image.” and reference [12], that again references [29], which states: “We compute the variance of gray-levels in a direction orthogonal to the orientation field in each block [...] the background has low variance in all the directions”.

Filtering

The idea of the filtering step is, to apply a band-pass filter, tuned to the local orientations and frequencies determined in the previous steps, to the normalized fingerprint image, in order to remove noises and clarify the ridge and furrow structure itself. “Gabor filters are band-pass filters which have both orientation-selective and frequency-selective properties and have optimal joint resolution in both spatial and frequency domains” [14, 3], therefore they are very well suited for this purpose.

The real part of an even symmetric Gabor filter has the following form:

$$G_{\theta,f}(x,y) = \exp\left\{\frac{-1}{2}\left[\frac{x_{\theta}^2}{\sigma_x^2} + \frac{y_{\theta}^2}{\sigma_y^2}\right]\right\} \cos(2\pi f x_{\theta}), \quad (4.10)$$

$$x_{\theta} = x \sin \theta + y \cos \theta, y_{\theta} = x \cos \theta + y \sin \theta,$$

where θ is the orientation of the Gabor filter, f the frequency of the sinusoidal plane wave and σ_x, σ_y being the standard deviations of the Gaussian envelope along the x and y axis respectively.

When applying Gabor filters to the normalized fingerprint image, per block the parameter values for frequency and orientation are determined by the corresponding values in the frequency image and orientation image created earlier. The parameters σ_x and σ_y are set to 4, based on empirical data, not specified in more detail. The size of the Gabor filters is set to 11.

4.7.2 FingerCode Algorithm

After a supplied fingerprint image has been enhanced and segmented as described in the previous section, the FingerCode algorithm by Ross et al. [31] next aims to determine localized information on the strength and on the orientation of the ridge-structure in the imprint. As already noted when explaining the enhancement algorithm, Gabor filters are very well suited to extract this type of features, therefore they are also applied by Ross et al. in present matching algorithm.

The definition of the Gabor filter here, is basically equal to the one given in [11] and presented in the previous section, Equation (4.10). Ross et al. also refer to the empirical data by Hong et al. when setting σ_x and σ_y , the standard deviations of the Gaussian envelope along the x and y axes, to a value of 4.

A difference then lies in the definition of the parameter f , the frequency of the sinusoidal plane wave at an angle θ with the x-axis. In [11] the value for f is set locally, directly obtained

from the corresponding block in the frequency image of the current fingerprint. Here, in the matching algorithm on the other hand, the value of f is pre-set in advance to a constant value that corresponds to the inter-ridge spacing in an average fingerprint image. (In [31] $f = \frac{1}{8}$, while in my implementation f is set to a separate specific value for each FVC2004 fingerprint database in use. See more information in Section “Gabor Filter Frequency” on page 76).

The most significant contrast to the application of Gabor filters during the enhancement phase is, that in the matching algorithm the value for θ , the direction of the filter, is not determined by a local orientation field, but a preset filter bank is used. This filter bank consists of eight separate Gabor filters, each oriented at a different constant angle. The reason behind that being, that the matching algorithm is not trying to perfectly fit the orientation of the filter to that of the underlying ridges in the fingerprint, but on the contrary examines the varying responses of the ridges and furrow structure to the differently oriented filters and bases its discriminating features there upon. The angles chosen by Ross et al. for the parameter θ are 0° , 22.5° , 45° , 67.5° , 90° , 112.5° , 135° and 157.5° .

The Gabor filter bank is then applied on the enhanced and segmented fingerprint, resulting in eight distinct filtered images. For each of these a standard deviation image is created by calculating per pixel the standard deviation of its 16×16 neighborhood. The union of these eight standard deviation images constitutes the so called *Standard Deviation Map*.

When the algorithm operates in enrollment mode, i.e. the fingerprint image has been intended for registration in the database (in present work called the “gallery”), the fingerprint’s standard deviation map is further processed in order to obtain a small, fixed length template. Each of the eight standard deviation images is sampled at regular intervals – Ross et al. suggest every 16^{th} pixel – in horizontal and vertical direction, creating the so called ridge feature images. The union of which then constitutes the *Ridge Feature Map*.

Matching

Suppose one or more fingerprint images have been enrolled, their ridge feature maps stored in the gallery. When a probe image is presented to the system, its standard deviation map is created as described above. This standard deviation map is then matched against each ridge feature map in the gallery as follows:

The eight ridge feature images within a map are expanded to the size of the probe fingerprint’s standard deviation images by interpolating with 0s. In other words, the ridge feature values “return” to the positions they have been sampled from, re-obtaining their original layout.

The translational alignment of the probe and gallery fingerprint is then achieved by aligning the two maps: each expanded ridge feature image is 2D correlated with the corresponding entry in the standard deviation map of the probe fingerprint. The correlation is done in the Fourier domain and the resulting matrices are summed. Next the values in that matrix sum (i.e. the potential translation offsets) are c . The translation vector then corresponds to the location of the maximum value in the weighted matrix sum.

Having established the alignment, the matching score is created by calculating the Euclidean Distance between the non-zero elements in the expanded ridge feature map and the corresponding non-zero values in the probe images’ standard deviation map.

The matching algorithm per se does not include a procedure to establish the rotation align-

ment of the probe and gallery fingerprint. Ross et al. suggest though, that this aspect could be accounted for by applying various rotated versions of the ridge feature map for matching with a probe's standard deviation map.

4.7.3 Enhancer and Segmentation Implementation

The implementation of the fingerprint image enhancement procedure based on the algorithm by Hong et al. [11] as well as the implementation of the FingerCode fingerprint matcher based on the algorithm by Ross et al. [31] were both written in Java. For specialized image processing I chose to apply the Java Advanced Imaging (JAI) API. Further details on this API as well as the main reasons, why I decided to use it, can be found on page 41, in Section "General Information".

Simply put, the fingerprint image enhancer receives a fingerprint image and returns an enhanced version of it, together with a so called region mask. The region mask specifies on a block-wise level, which areas in the enhanced image show ridge and furrow data and which areas belong to the background or represent fingerprint areas that are too heavily damaged to obtain reliable structural information. The enhancement process itself is modeled by and large on the algorithm steps described briefly in section 4.7.1.

Normalization

This step is basically implemented word for word according to the specification in [11]. However the normalized images tended to show considerable differences in range of values – some areas within an image were set to gray-levels far below that of other areas, thereby impeding the performance of subsequent algorithmic steps. Large improvement could be achieved by further normalizing the calculated image values to a range of [0,255] and even better to a range of [-255,255]. Tests during implementation showed that this additional range-normalization leads to improved enhancement results, ultimately leading to a better separability between true matches and true non matches in the FingerCode matcher algorithm. An example for the differing enhancement results when applying the additional range normalization or not can be found in Figures 4.9b (with range normalization) and 4.10 (without).

Orientation Image Estimation

The orientation image estimation starts by calculating the horizontal and vertical gradients of the normalized fingerprint image. I tried out a Sobel-, a Prewitt- and a Frei-Chen-operator and all three lead to nearly the same results in terms of the final enhanced image. For the current program version I settled on the Sobel-operator.

In the remaining steps of the orientation image generation, again I basically followed the formulas and specifications given in [11]: The normalized fingerprint image is tiled into blocks of size 16×16 . Per block the least square estimate of the local ridge orientation is calculated. Figure 4.7b shows an example of a set of least square estimates overlaid on the original fingerprint image. Next this first version of the orientation image is smoothed by initially converting it into a continuous vector field, then filtering this field with a 2-dimensional low-pass filter with unit integral and finally re-obtaining the orientation estimates. An example for the resulting smoothed orientation image can be seen in Figure 4.7c.

From my side, only minor adjustments were necessary in the implementation. For example I had to deal with problems occurring in angular calculations due to the fact, that a ridge orientation of α and one of $\alpha + \pi$ can not be differentiated from each other.

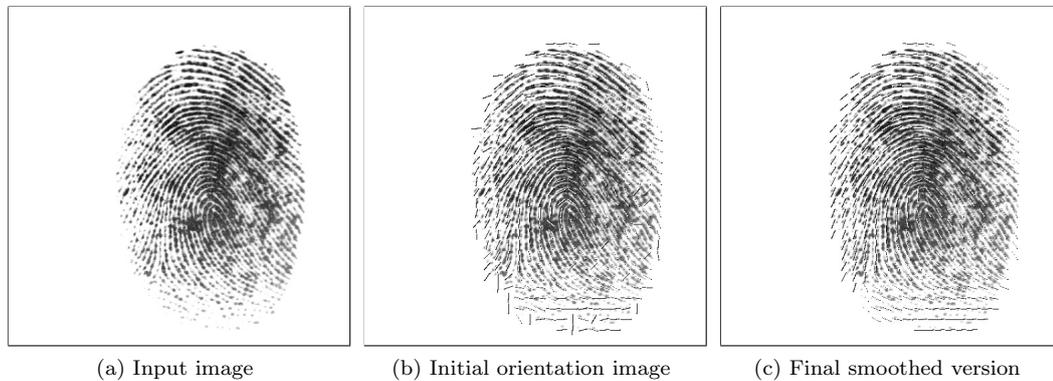


Figure 4.7: Input fingerprint image and the estimates for the ridge orientation field (overlaid). a) shows the original input image (DB1, ID 78.3), b) is the initial set of least square estimates for the orientation field and c) shows the final version, smoothed using a 2-dimensional low-pass filter.

Frequency Image Estimation

As input to this procedure we receive the normalized fingerprint image and the orientation image, created in the previous step. An oriented window is defined for every block in the normalized image and therein the x-signature is calculated, precisely following the formulas given in [11].

Next the frequency of the local ridge and furrow structure is determined by establishing the average distance between peak-values within the x-signature. I found it interesting, that in this regard even the treatment of the boundary values of the 1-dimensional x-signature array has a considerable influence on the detection of the actual ridge frequency, evident in the correct alignment of the Gabor filter in the final filtering step, ultimately leading to an effective image enhancement. The best configuration I found in my tests is, when the first array value is regarded as peak, given the second value is lower, while on the other hand the last array value is never considered as peak. Another topic that is influential, when establishing the ridge frequency from the x-signature, is how to deal with plateaus, in other words a consecutive number of equal values in the array. In these cases I choose the center of the plateau as peak location.

When the frequency value has been calculated, it is inspected. If the frequency does not lie within a certain range characteristic for the present fingerprint image type, the respective block is deemed invalid, i.e. the ridge and furrow structure has been “corrupted” by minutiae, singular points, noise, etc., and the block is marked with a value of -1. Hong et al. suggest to test for a frequency range of [0.33,0.04]. For the fingerprint images used in the experiments of present work, I found to achieve better results, when testing for a narrower range of [0.21,0.08]. An example fingerprint image and this first version of its frequency image can be seen in Figures 4.8a and 4.8b.

The interpolation of the invalid blocks with the frequencies in the neighborhood applies a Gauss filter. This step again is implemented following the specifications word for word. For

comparison, the interpolated frequency image for the example image can be seen in Figure 4.8c.

For the final step in frequency image estimation, the removal of outliers, Hong et al. propose the use of a 2-dimensional low-pass filter with unit integral and a kernel size of 7. When in my implementation I worked with a mean filter of size 7, the frequency image appeared to be smoothed quite strong, so I also experimented with mean filters of sizes 5 and 3, as well as with Gaussian filters of the sizes 7, 5, and 3. In regard to the final enhancement image and lastly the matching performance, the best results were achieved, using a simple mean filter of size 3. The final frequency image for the example fingerprint image can be seen in Figure 4.8d.

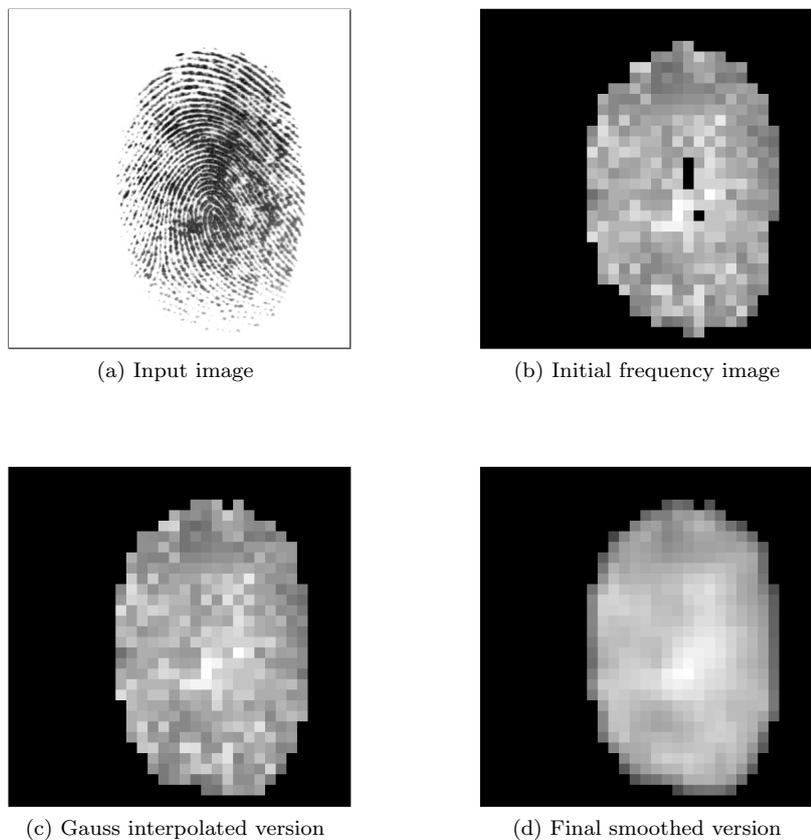


Figure 4.8: Input fingerprint image with the estimates for its ridge frequencies. a) For comparison the input image is shown again. b) Represents the initial frequency image, derived from the block-wise x-signatures. Black (empty) blocks mark those areas, where the established frequency is not within pre-set limits. c) Shows the frequency image interpolated using a Gauss filter and c) is the final version, smoothed using a 2-dimensional low-pass filter.

Filtering and Segmentation

As for the filtering step in my implementation, I closely followed the specifications and formulas provided in [11]: per block in the normalized fingerprint image a Gabor filter is created, with its parameters being set to the corresponding values in the frequency and

orientation images. Each pixel within the block is then filtered and the value stored in the output image. From my part, no additions worth mentioning were integrated.

For the creation of the region map, i.e. a map indicating foreground and background blocks in the fingerprint image, Hong et al. apply a one-nearest neighbor classifier. Aiming to avoid the complexities and complications of implementing and training such a classifier, I adopted the approach of Ross et al. instead, to use a different and simpler method to establish a segmentation.

As mentioned in Section “Region Map Generation” on page 56, Ross et al. essentially seem to be using “the variance of gray-levels in a direction orthogonal to the orientation field in each block” as criterion to establish a segmentation into foreground and background blocks on a fingerprint image. “Orthogonal to the orientation field”, to my understanding, is an area of pixels, similar to the one defined by the oriented window, used when creating the x-signature for ridge frequency estimation. Yet despite my best efforts to follow this approach by Ross et al, I was not able to find any accordant ways to calculate a criterion with sufficient distinctiveness, to reliably distinguish between foreground and background blocks – at least not in a way that would outperform the solution I finally settled to. Attempts to limit the pixels within the oriented window to only those inside the current block, or to use different measures than the variance, did not show any better results either.

Now to the method of segmentation that I included in my current implementation: A measure that, to my surprise, has already quite a fair distinctive character is the ridge frequency. Taken the frequency values by themselves, it is already possible to determine a threshold that creates a segmentation, roughly fitting the results expected from an effective algorithm. And this is true for fingerprint images from all three of the FVC2004 databases in use for present project. Reasons that might explain the unexpectedly good performance of the ridge frequency as decision criterion are: The frequency value is calculated from the x-signature, which is orthogonal to the respective ridge orientation and has a length of even twice the block-length. Therefore, if the orientation fits that of the actual ridges in the corresponding fingerprint image block, then also the ridge frequency has a good chance to be close to the actual value. Additionally, as mentioned on page 61, the frequency value has been certified to be within a specific range, representing a clear ridge and furrow structure in an average fingerprint image.

Still, a number of blocks remain, that would actually belong to the background, but due to specific noise or other particular conditions in the original fingerprint image, possess frequency values similar to those of genuine foreground blocks. A simple frequency threshold alone would therefore not be sufficient to correctly categorize these areas. When inspecting the corresponding segments in the enhanced image, it can be found, that the majority of them has one thing in common: The distance between the highest and the lowest image value within each block is clearly smaller, than that observed in blocks displaying clear ridge and furrow structures. (Henceforth I will call the distance between the highest and the lowest image value within a block its *dynamic range*). For that reason, I use the dynamic range of a block in the enhanced fingerprint image as my second criterion in my segmentation method. This also has an additional benefit: The dynamic range can likewise be used to identify sections in the original image, that are part of the fingerprint itself, but show an indistinct ridge and furrow structure, because those also tend to produce smaller values (in terms of the absolute value) in the enhanced image. Therefore by adjusting the threshold for the dynamic range, it is also possible to determine the degree of clarity a local ridge and furrow structure must exhibit, for its corresponding block to be part of the foreground.

So to summarize the segmentation method being used in the current implementation of the fingerprint enhancer: A block in the enhanced fingerprint image is declared a foreground block, if its ridge frequency and its dynamic range are both above the respective thresholds. The dynamic range is better suited to identify blocks of indistinct ridge and furrow structure in the original fingerprint. Naturally both thresholds have to be optimized for the respective type of fingerprint image in use - in the case of current project, the threshold values differ for each of the FVC2004 databases. The classification of each block is recorded as boolean value in the region map.

Enhancer output

The final output of the enhancer is an object containing the enhanced fingerprint image and the corresponding region map. An example for the resulting enhanced fingerprint image can be seen in Figure 4.9. The segmentation is hereby already “included”, as in the example image, background blocks have been set to gray-level 0.

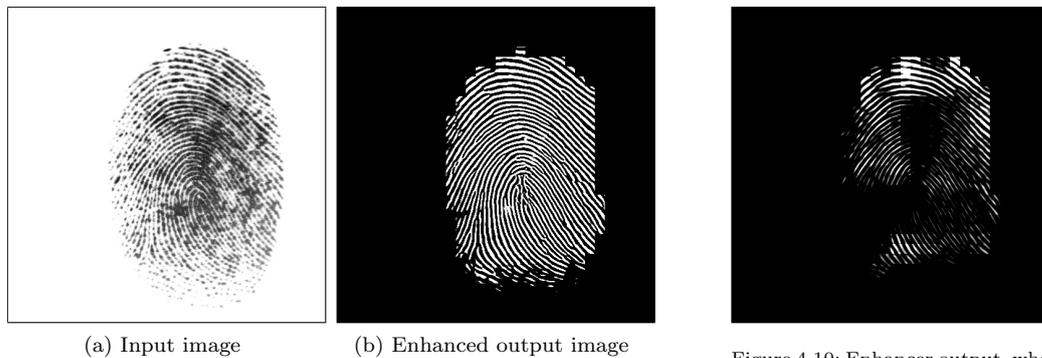


Figure 4.9: Comparison of enhancer input (DB1, ID 78_3) and output. (Additionally the output is segmented using the established region map).

Figure 4.10: Enhancer output, when range normalization (see 4.7.3 for details) is not applied.

So far the example images presented in this section have been showing the entire enhancement process for an image taken from the FVC2004 test data set DB1 (to be precise: my adapted set of DB1 images with an image size of 512×512 pixels). To give an example for the enhancement effects for (slightly difficult) images of databases DB2 and DB3, the essential intermediate stages and the final results can be found in Figures 4.11 and 4.12.

4.7.4 FingerCode Matcher Implementation

After the fingerprint image has been enhanced and the region mask, indicating the foreground and background blocks, established, the next step is to create the standard deviation map. In principle, following the algorithm description in [31], this process is the same for gallery images as for query images. Only thereafter the algorithm would distinguish, if the current fingerprint is destined for enrollment in the gallery or as query in a matching process and continue the course of action accordingly. In my implementation on the other hand, I have to make the distinction already at this point. The reason lies in the integration of rotational alignment into the matching process:

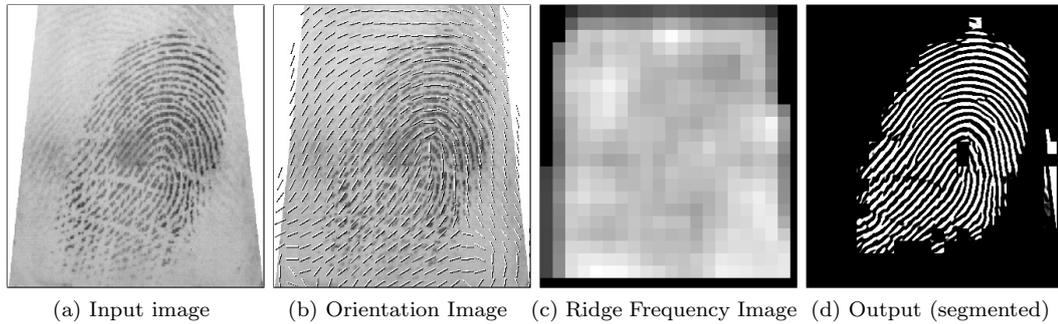


Figure 4.11: Essential enhancement steps for example fingerprint image of FVC2004 DB2 (ID 28.2). a) Is the input image. b) The input image overlaid with the established ridge orientation image. c) The smoothed ridge frequency image. d) The final output of the enhancer, segmented using the additionally created region map.

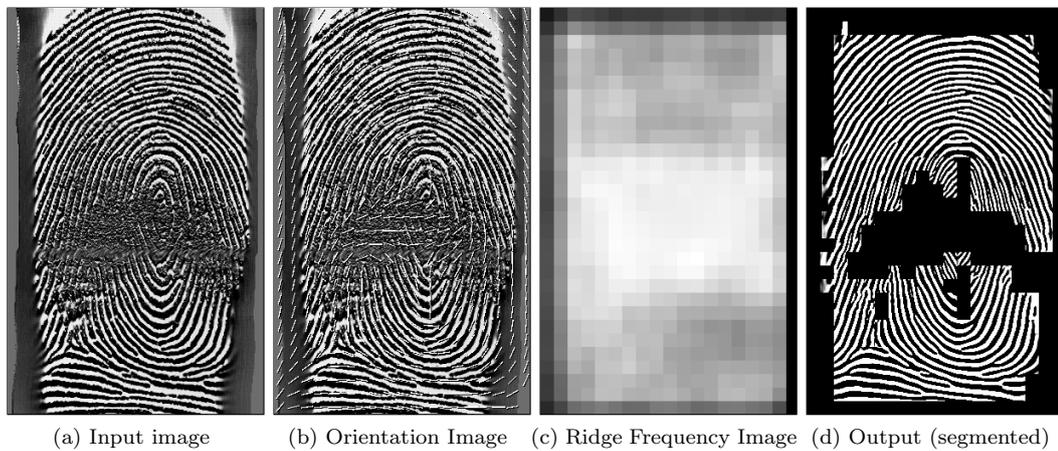


Figure 4.12: Essential enhancement steps for example fingerprint image of FVC2004 DB3 (ID 32.3). a) Is the input image. b) The input image overlaid with the established ridge orientation image. c) The smoothed ridge frequency image. d) The final output of the enhancer, segmented using the additionally created region map.

Rotational Alignment

Concerning the rotational alignment of a query fingerprint and the gallery template it is matched against, I follow the proposal by Ross et al. to use various rotated versions of the template ridge feature map and correlate them with the query feature map.

In this line of thought and also with an eye on keeping the processing speed within reasonable bounds, I tried out different ways of creating the “various rotated versions”:

Rotating the original supplied fingerprint image Rotated versions of the originally supplied fingerprint image are created and for each of the rotations the steps: enhancement, creating of the standard deviation map and sampling it for ridge feature map creation are performed. Following the separate ridge feature maps are assembled into a comprehensive one.

This method is the slowest of all, as enhancement has to be performed for each rotated fingerprint image.

Rotating the enhanced image The original fingerprint image is enhanced and then this enhanced image is rotated. Here again per rotated image a separate standard deviation map is created and sampled to obtain the ridge feature map. Finally the set of ridge feature maps is united into a comprehensive one.

In this rotational alignment method one topic has to be taken special care of: As was already noted on page 44, in the listing of Section “Rotation Alignment” of the POC-matcher, rotation of a binarized enhanced fingerprint image leads to strong blurring along the ridge lines, therefore a second adjusted binarization step has to be applied on each rotated image prior to standard deviation map generation, to re-establish the clarity of the ridge and furrow structure.

Rotation adjusted sampling In this method the original image is enhanced and the standard deviation map is created “as normal”. Rotation comes into play, when the ridge feature map is created from the standard deviation map, as now the sampling locations are rotation adjusted. In other words not the fingerprint image or one of its representations so far is rotated, but the sampling grid that produces the ridge feature map.

This method is fast, as no image rotations are necessary.

Rotation adjustment during matching My last rotation alignment candidate is somewhat related to the previous one, yet here the rotational alignment happens during the matching phase instead of during enrollment. Here every ridge feature map only contains the representation of the original image and no rotations. During the matching phase then, when the ridge feature map gets expanded and interpolated with 0s, the locations, where the ridge features get expanded to, are rotation adjusted.

This method is the fastest and uses the least memory, as the ridge feature maps only contain one rotation.

For my implementation I chose the second method, to rotate the enhanced image. In matching performance it is clearly better than methods 3 and 4 (with 4 showing the worst results) and only slightly inferior to the first method, rotation of the original image. Whereas the first method is rather unfit for actual application, due to the enormous additional time needed for the additional enhancement operations.

Standard Deviation Map - Query Image

In a first step, the image as returned by the fingerprint image enhancer is binarized to amplify the distinctiveness of the ridge and furrow structure, with the fingerprint ridge-information being displayed as 1 and the background being displayed as 0 (or any other high-low combination for that matter). In the course of my preliminary tests during the implementation phase, I found that this binarization has great positive influence on the matching performance.

Another step concerning the enhancer output, that can be activated via a parameter on program start, is to apply a single pass of erosion with a plus-sign shaped structuring element of size 3×3 on the region mask. This serves to impose an even stricter limitation of the foreground area to “pure” foreground blocks, eliminating bordering blocks that potentially contain just ends of ridge lines that fade into a background area. (Further notes concerning the application of erosion when dealing with fingerprint images from the FVC2004 databases can be found in Section “Erosion” on page 75.)

The next and very essential step is to filter the enhanced fingerprint image using the Gabor filter bank. As noted in the explanation of the matching algorithm in section 4.7.2, the orientations of the eight filters in the filter bank is set to constant pre-defined values. Also the filter frequency is determined a-priori, depending on the type of fingerprint images in use. Hence the Gabor filters can already be calculate on program start and the filter bank can be realized as simple look-up table, allowing for fast access.

The filters are generated using the same procedure as the image enhancer (see section 4.7.3 for details). The convolution is done using the JAI operator *convolve*. The result of this processing step are eight filtered images, showing the response of the ridge structure in the fingerprint to the Gabor filter of respective orientation. Corresponding examples can be seen in Figures 4.13 and 4.14 on page 70.

For each of the eight filtered images a separate standard deviation image is generated. Thereby also the region mask, established in the segmentation step, is taken into account. Every pixel related to a background block is automatically set to 0. For each pixel in the foreground area the standard deviation of its 16×16 neighborhood is calculated and stored in the output image. Special care has to be taken, to exclude those foreground pixels from the calculation, that are part of the boarder that was introduced by the convolution operation along the edges of the filtered image. This boarder has a size of half the Gabor filter kernel and contains unreliable data.

Taken together, the eight standard deviation images compose the standard deviation map of the originally supplied fingerprint image. Corresponding examples of standard deviation images can be seen in Figure 4.15 on page 71.

Standard Deviation Map - Gallery Image

This section explains the differences in processing, when the standard deviation map is created during enrollment, for a fingerprint that is intended to be registered in the gallery. The additional processing steps necessary originate from the introduction of rotation alignment capability to the fingerprint matching algorithm. (Present explanation assumes, that the alignment method titled “Rotating the enhanced image” on page 66 is in use.) As result of this processing step, a standard deviation map will be created, that contains eight standard deviation images – stemming from the 8 pre-set Gabor filter orientations – for each rotated version of the enhanced fingerprint image. As default values my implementation uses an angular range of $[-20^\circ, +20^\circ]$ with a step-width of 1° . (Corresponding to the settings of rotation alignment for the POC matcher).

The enhancement step is executed on the supplied gallery fingerprint the same way as for the query fingerprint. A difference then occurs in dealing with the region mask. Generated from the original image, it is not easily rotatable, due to its limitation to block-indices. For that reason it can not be used to impose a segmentation on the rotated fingerprint images. As alternative I already incorporate the segmentation directly into the enhanced image, by setting all those pixels to 0, that correspond to a background block. A drawback is though, that during creation of the standard deviation image I can not refer to the region mask to exclude entire background blocks from calculation already in advance, therefore substantially prolonging the overall processing time for this step.

After the segmentation has been incorporated directly into the enhanced fingerprint image, this image is rotated according to the rotation alignment parameters noted above.

As already explained on page 66, when the enhanced image is rotated, an additional adjusted binarization has to be applied, to mend the negative effects of the bi-cubic interpolation on the clarity of the ridge lines. This is done for each single rotated image and afterwards it is used to create the individual standard deviation map, following the same calculation procedure as was described for query images.

The output of this processing step is a comprehensive standard deviation map, that contains an individual map for each rotated version of the enhanced fingerprint image.

The examples in Figures 4.14 and 4.15 shows the Gauss-filtered images and the standard deviation map for the gallery image (Figure 4.13) in the “original” rotation of 0° .

Ridge Feature Map

In this final step of the enrollment procedure the FingerCode – the fixed-length template of the current fingerprint, to be stored in the gallery – is created. A ridge feature map can be seen as the structure containing the FingerCode. It is directly obtained from the standard deviation map created as described in the previous section.

Following the specification of the rotation alignment method “Rotating the enhanced image” on page 66, the final ridge feature map is again a compilation of separate ones, each representing a single rotated version of the supplied fingerprint image.

The creation of the ridge feature map is straight-forward: Per rotation step, for each of the eight standard deviation images, every 16^{th} pixel is sampled and stored in a ridge feature image in according structural layout. The overall ridge feature map is then stored as FingerCode in the gallery.

Using the default parameter settings of an angular range of $\pm 20^\circ$ and a step-width of 1° , the resulting ridge feature map hence contains 41 separate ones, each comprising 8 ridge feature images. Considering a fingerprint image with a resolution of 512×512 pixels, every ridge feature image contains 31×31 values. Hence the entire FingerCode for a single fingerprint image has a length of $41 \times 8 \times 31 \times 31 = 315208$ values.

Figure 4.16 on page 72 shows an example for a ridge feature map of the gallery image (Figure 4.13) at the rotation step of 0° . For the sake of perceptibility the ridge feature images have been “zoomed”, with each block representing a single ridge feature value.

An important figure, needed during the matching phase, providing information on the significance of the matching result, is the number of nonzero FingerCode values. (A detailed explanation on that topic is provided in section “Matching” on page 70).

With regard to simplicity and portability of the program for execution among different systems, the current implementation of the FingerCode matcher uses the file system as storage for the ridge feature maps. The maps are stored as .rfm files and the output folder as well as the folder from where they are loaded back into the system, can be specified per parameters on program start. Along with the feature values of the FingerCode also the corresponding rotation parameters – start- and end-value of the angular range and the step-width – as well as the number of feature values nonzero are stored in the output .rfm files.

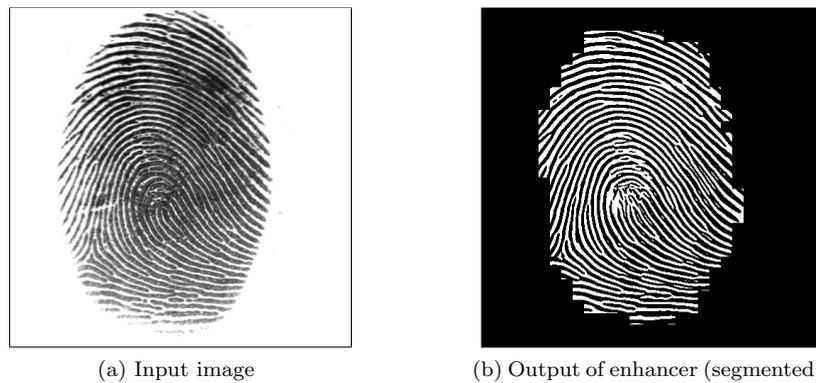


Figure 4.13: Example gallery fingerprint image (DB1, ID 90.3) and the respective output of the enhancement step, which will be used as basis for the subsequent steps in the ridge feature map generation.

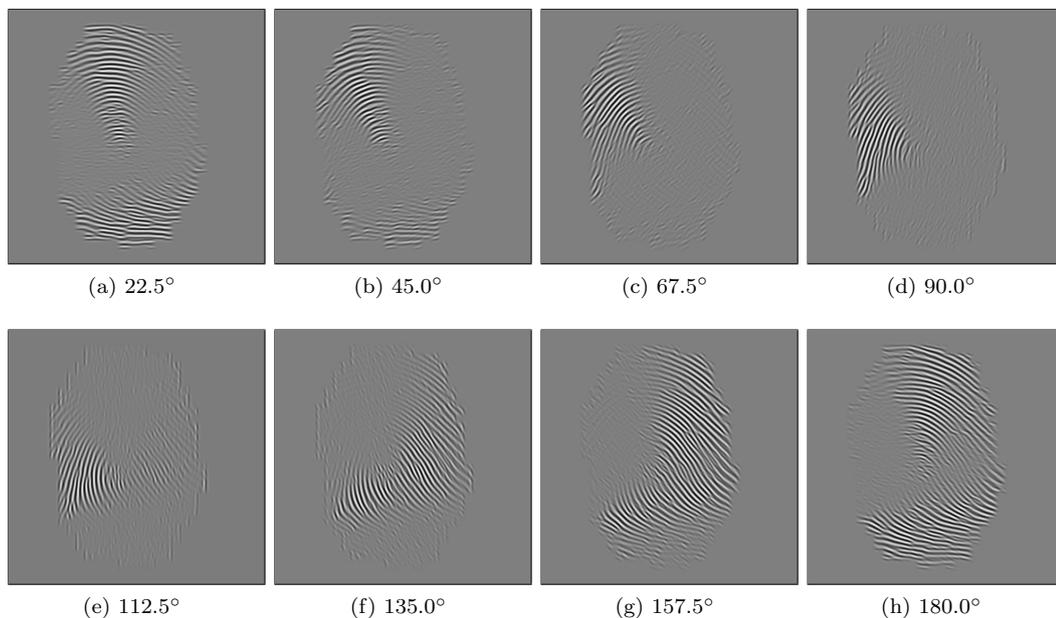


Figure 4.14: The result of applying the Gabor filter bank on the enhanced input image (Figure 4.13b). Each image is generated by one of the eight differently oriented filters within the filter bank.

Matching

When the FingerCode matcher is used in matching mode, the filename of a query fingerprint image is presented together with the ridge feature map from the gallery, that the query should be matched against. The query fingerprint is loaded and a standard deviation map is created as described on page 67. This is then matched against every single rotated version within the ridge feature map as follows:

The ridge feature map of current rotation step is loaded and each ridge feature image therein is expanded to the size of the query image. This is done by mapping feature values of a ridge

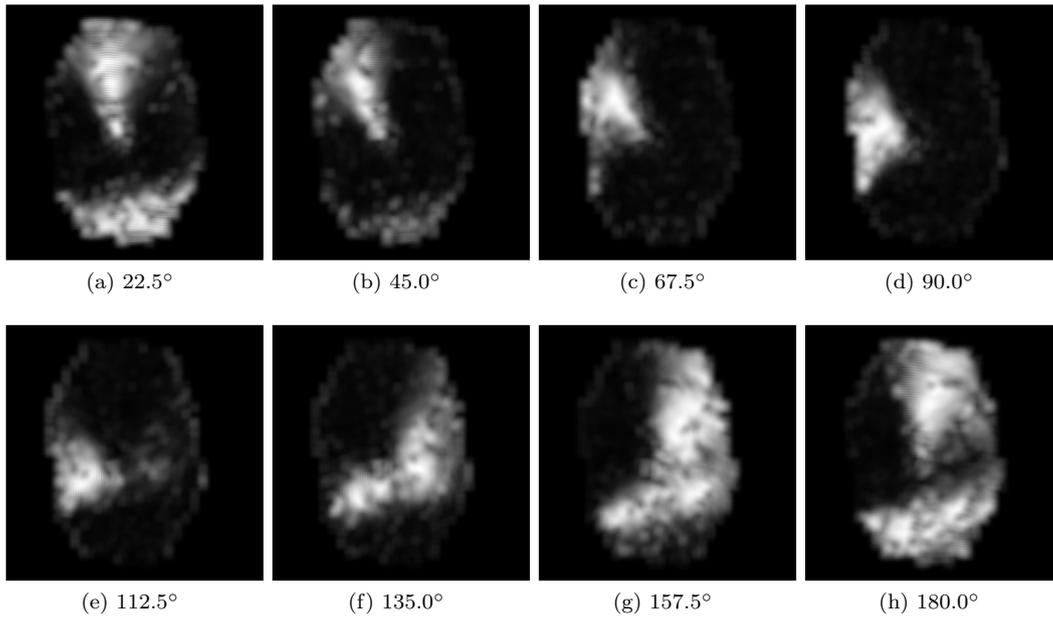


Figure 4.15: The eight images within the standard deviation map for the rotation value of 0° of the input image (Figure 4.13). Each standard deviation image is created from the corresponding Gabor filtered image (compare to Figure 4.14).

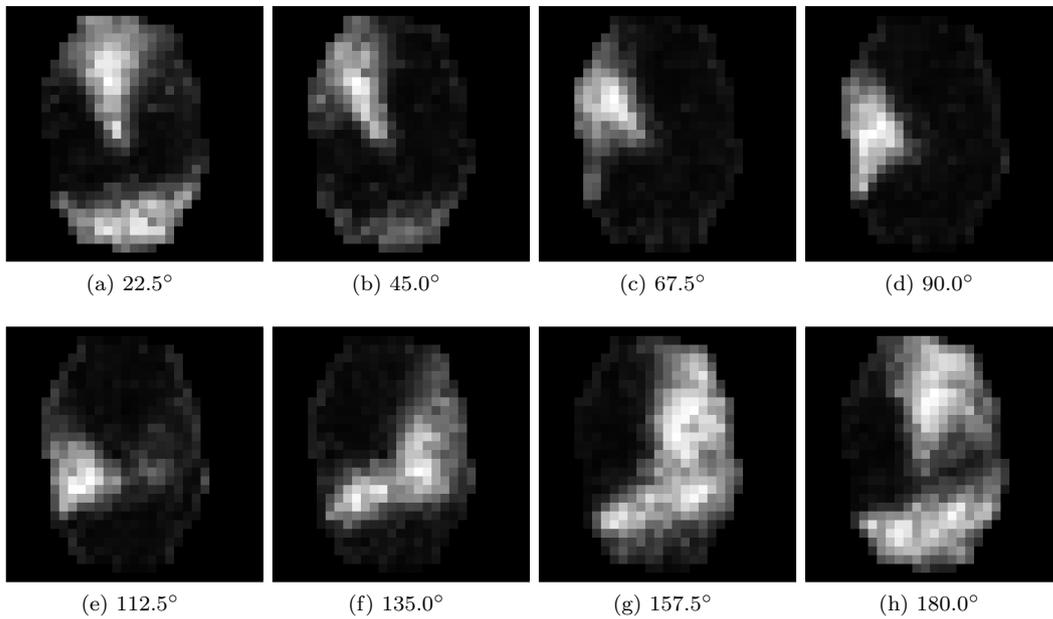


Figure 4.16: The eight images within the ridge feature map for the rotation value of 0° of the input image (Figure 4.13). The ridge feature map is created from the standard deviation map by basically sampling every 16^{th} pixel. For the sake of perceptibility the ridge feature images have been “zoomed”, with each block representing a single ridge feature value.

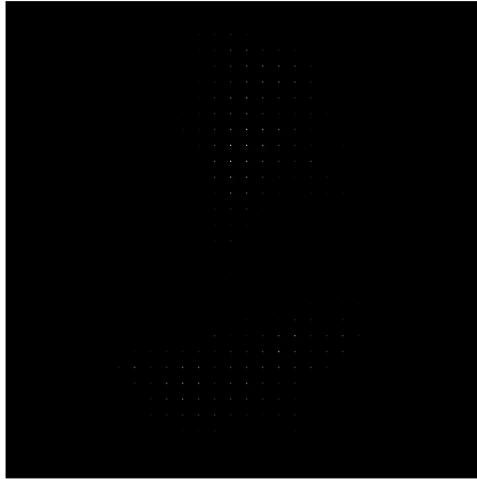


Figure 4.17: Ridge feature map expanded to size of probe image by interpolating with gray-level 0. The example image is the expanded version of the ridge feature image in Figure 4.16h, corresponding to Gauss filter orientation of 180.0° .

feature image back to their original locations (by simply multiplying their horizontal and vertical indices with 16) and setting all other pixels in the resulting image to 0. An example for an expanded ridge feature image can be seen in Figure 4.17.

Next a translation vector has to be determined, that represents the translational offset between the query fingerprint and the gallery fingerprint. This is done by correlating the two respective maps, or to be more precise, the eight images therein. To speed up the process, the correlations are performed in the Fourier domain. For that matter each standard deviation image and its corresponding expanded ridge feature image are Fourier transformed, the ridge feature image is conjugated and then both images are multiplied. The outcome of the multiplication is then inverse Fourier transformed.

The hereby computed eight correlation matrices are summed up and the result constitutes the “unweighted correlation matrix”. At this point Ross et al. apply a weighting function, to take the amount of overlap between the two maps into account, creating the “weighted correlation matrix”. However this is a step that I found to not have any positive effects on the matching performance of my implementation, even when trying out different parameter settings for the proposed weighting function. This is probably due to the number of adaptations that I introduced at several points in the enhancement and matching algorithm. (For further details please also refer to section 4.7.5). Hence I pass the weighting procedure and instead use a different criterion to account for the overlap between the standard deviation map and the expanded ridge feature map when calculating their euclidean distance. (See following paragraph for details)

Having established the correlation matrix of the two fingerprint maps, the translation vector then corresponds to the horizontal and vertical index of its maximum value.

Concluding a matching run per rotation, the matching score is established. The core of this processing step is to simply calculate the Euclidean distances between translation adjusted ridge feature values of the gallery fingerprint and standard deviation values at the same coordinates in the corresponding image of the query’s standard deviation map. The matching score is then determined by summing up these differences, yet only those, where both feature

values involved are nonzero.

One problem though, that has to be dealt with, is related to the amount of overlap between the two maps: For demonstration consider the theoretical extreme case, where the maps only overlap in a single nonzero feature value ³, as opposed to a “normal” case, where about 2000 differences are being summed up. If this one single difference is per coincidence quite small, then the query and gallery fingerprint would already be considered a good match. One measure that I included in my implementation, to deal with this problem, is to average the distance sum over the number of distances included. This already helps a lot to get a more stable overall matching performance, largely reducing the influence of the actual number of overlapping feature values on the matching results. Yet a significant set of fingerprint pairings still remain, where the query map and gallery map only have a very small area of overlap, though those few feature values that can be considered, exhibit a small overall distance. The resulting low matching scores then cause a number of false accepts. To also handle these cases, I included a threshold in my implementation, that defines the minimum amount of overlap, a matching pair has to have, in order for its matching score to be considered significant. If the threshold is not reached, the matching score is set to a pre-defined maximum distance value. As criterion for the amount of overlap, I calculate the ratio of the number of distances included in the matching score to the number of maximum possible ones, i.e. the number of feature values nonzero in the gallery images’ ridge feature map. (Also refer to Section “Amount of Minimum Overlap” on page 78). In my tests, this additional quality criterion clearly reduced the number of false accepts in the overall matching results.

When the matching scores for every single rotated version of the gallery fingerprint has been calculated, then the lowest one (lowest, because the matching score represents a distance measure) is seen as representing the best alignment and is returned as final result of the matching process.

Batch Runner for FVC Test

The batch runner for the FingerCode matcher follows in purpose and functionality the one created for the POC matcher. Please refer to the corresponding section on page 48 for details. The only major difference lies in the integration of a template gallery in the FingerCode matcher. While in the POC matcher case query and gallery fingerprint images are presented in equal form to the matching algorithm, the FingerCode matcher, as noted on numerous occasions, relies on a gallery of ridge feature maps, to match the query fingerprints with. This means for the batch runner, a complete FVC-conform test run basically is separated in two stages:

Enrollment The batch runner loops through the set of required gallery fingerprint images and creates or loads the corresponding ridge feature map. When creating, the batch runner invokes the steps described in section “Ridge Feature Map” on page 69. When loading from file, a check is made, if the rotation parameters equal those necessary for

³Admittedly the “only one value overlaps” case will not appear in real-life applications of the FingerCode matcher, yet cases where less than 100 overlapping values exist, may occur. This can be the case when then query fingerprint area is rather small, like for example is quite common in latent fingerprints and this area’s structure is additionally quite different from the ridge structure of the gallery fingerprint. A significant translation alignment will not be found then and the resulting translation vector might be so “wrong”, that the fingerprint maps will only overlap in a very small section.

the current test run. If not, then the .rfm file is unsuitable and the ridge feature map has to be created anew in the required configurations.

Matching The batch runner establishes the necessary pairings for a test run and calls the matching routine of the FingerCode matcher with the filename of the query image and the according ridge feature map from the gallery.

The results of the batch run are once again saved to a csv-file.

4.7.5 FVC2004-Database Specific Differences

Among the 42 different parameters set via the batch runner, also the FingerCode implementation has a set of database specific settings, that fine-tune the enhancer and matcher for the different types of fingerprint images in the FVC2004 databases.

Segmentation Thresholds

As already mentioned in Section “Filtering and Segmentation”, on page 64, individual ridge frequency thresholds and dynamic range thresholds had to be established for each database. The values were primarily chosen to separate the image background as effective as possible from the fingerprint. Further, for application together with the FingerCode matcher, the dynamic range value was set with the aim to also eliminate a moderate amount of unreliable blocks, stemming from areas of indistinct ridge and furrow structure in the original fingerprint.

Erosion

As noted in Section “Standard Deviation Map” on page 67, the implementation offers the possibility to apply the morphologic operation *Erosion* on the region mask returned by the fingerprint image enhancer. A plus-sign shaped structuring element of size 3×3 is used to basically reduce the foreground area by one block along its border, in order to further eliminate blocks that only partially contain foreground data (for example ridge line endings). An example can be found in Figure 4.18.

Erosion is used when dealing with fingerprint images of DB1. For databases DB2 and DB3 erosion did not lead to an improvement in matching performance.

Gabor Filter Frequency

As was already mentioned on several occasions, in the FingerCode matching algorithm, the frequency of the eight Gabor filters in the filter bank is constant and pre-set to a value representing the inter-ridge spacing of an average fingerprint image. Obviously an individual value for each database had to be determined. For this I adapted the procedures used in the fingerprint enhancer to create the frequency image. My related tests then showed, that a precision of at least 10^{-3} is necessary, to achieve best possible matching performance. While in [31] the filter frequency was set to 0.125, the values I established are as follows: 0.136 for DB1, 0.147 for DB2 and 0.153 for DB3.

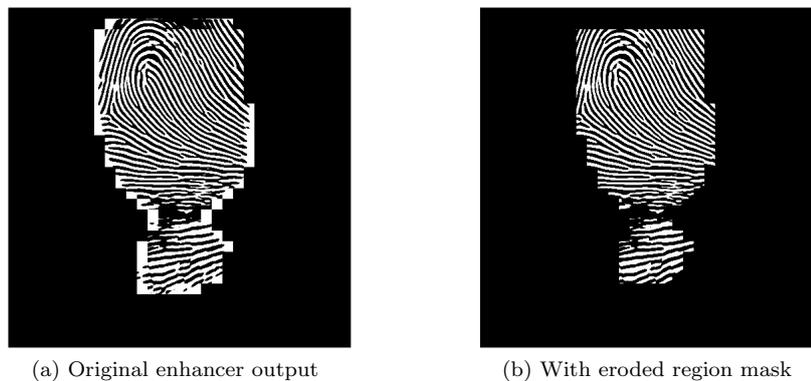


Figure 4.18: This Figure displays the effect of additionally eroding the region map for the enhancer output of a fingerprint image of DB1 (here: image ID 38.1) to remove bordering blocks only partially containing foreground information.

Variance Map, not Standard Deviation Map

For databases DB2 and DB3 better matching performance can be achieved by using the variance instead of the standard deviation as representation of the ridge features. This substitution that was for example also applied by Ross et al in [30].

For DB1 using the standard deviation still leads to better matching results.

Convolution - Extended Range

This options can be applied in the stage of standard deviation map creation. Tests during implementation showed, that for image databases DB2 and DB3 obvious improvements in matching performance can be achieved by normalizing the values of the Gabor filtered images. This positive effect is be amplified, when the values are normalized to a relatively large range – in current version a range of $[-1000,1000]$ is used. For images of DB1 however, the normalization does not improve the matching performance.

Weighting of Ridge Feature Values for Translation Alignment

My implementation of the translation alignment method according to the algorithm by Ross et al. and explained in Section “Matching” on page 58, did not lead to satisfactory results. In preliminary tests with images manually shifted by specified offsets, the majority of translation vectors, established from the correlation of the query’s standard deviation map and the gallery’s ridge feature map, was too far from the expected results. Also the weighting function proposed in [31] did not bring noticeable improvement.

When inspecting a ridge feature map, it can be found, that each of the eight images within contains a small set of high values, corresponding to those fingerprint ridges that responded strongly to the Gabor filter of respective orientation and a larger set of low values, analogously representing weaker ridge responses. Now to improve the results of the correlation of the feature maps, I sought to heighten the influence of the crucial, high feature values. For that purpose I am using an altered version of the ridge feature map in the correlation. Two

strategies that lead to considerably improved alignment results are:

Power of 3 Every value within the ridge feature map is taken to the 3rd power, thereby obviously increasing the distance between high and low feature values. (The 2nd power turned out to be insufficiently strong, while powers higher than 3 did not display better matching results). This strategy is applied when matching fingerprint images of DB1.

Percentage Threshold The highest value within the ridge feature map is determined. Now only those ridge feature values are transferred in the altered map, whose magnitude is at least a certain pre-defined percentage of the maximum value. This method leads to notably improved matching results for fingerprint images of databases DB2 and DB3. The most effective percentage-value determined in my tests is 97%.

Amount of Minimum Overlap

Another parameter, that had to be individualized for each database, is the threshold that determines the minimum amount of overlap between the expanded ridge feature map of the gallery image and the standard deviation map of the query image necessary, for the matching score to be considered significant. In my implementation I take the amount of overlap into account by calculating the ratio of Euclidean distances included in the matching score to the number of maximum possible ones (i.e. the number of nonzero values in the ridge feature map)

The thresholds per database were empirically determined, based on an extensive series of tests using different parameter values. For DB1 a minimum overlap of 30% is necessary to obtain stable matching scores, for DB2 just 15% and for DB3 20%.

Maximum Matching Score

As already noted before, the matching score returned by the FingerCode algorithm is a distance measure, based on the Euclidean Distance between the ridge feature representations of the query and gallery images. Therefore the lower the score, the better – the matching score for two similar images is 0. An interesting question is, what value to set the maximum score to. (The maximum score is needed, as the score for matching pairs with insufficient overlapping area will be set to this value). A theoretical maximum for each type of database is very hard to establish so I chose to rather use empirically determined values instead. (A value signifying positive infinity or the maximum representable number of the matching score's data type would be possible as well, yet is rather troublesome when normalizing the results for analysis). I refined my choices with increasing number of test runs performed, so that the values that I finally came up with – the values set as defaults in the current program version – are above the highest score found in the results of all tests conducted for present work together (StirMark related and others). As these tests contain a very broad variety of image qualities, the risk in a real-life application of a fingerprint matching pair to ever produce a matching score exceeding my default maximum values is very small and even if this were to happen, the impact would be very limited, as by then the score is already at a level, that undoubtedly signifies a mismatch.

Chapter 5

StirMark Experiments

This chapter presents details and results regarding the experiments aimed to investigate the influence of “natural perturbations” in fingerprint images on the matching performance of different fingerprint matchers and fingerprint matcher types. For more information on the motivation behind these tests, as well as the particular objectives, please refer to section 1.1.

The testing procedure in the experiments follows the protocol specified in the Fingerprint Verification Competition 2004 and 2006 and as fingerprint test data I likewise apply the sample images of the FVC2004. Further details on the matching protocol and the test data can be found in sections 3.2 and 3.1 respectively. As explicated in section 2.3, the “natural perturbations” will be simulated and introduced in the test images by individual “attacks” provided by the StirMark Benchmark utility. In each pair of fingerprint images to be matched, only the probe image will be perturbed, while the image representing the enrolled gallery fingerprint, equals the original sample from the FVC2004 databases (see also section 2.3.2 for further explanation).

In the following, each section deals with a specific type of perturbation. I will briefly explain the respective StirMark attack utilized and in how far I consider it fit, to simulate a “natural perturbation”. In line with the possible application of present experiments in form of aforesaid *extended benchmark* (please refer to page 3 for explanation), I will list the specific parameter settings, thereby providing for repeatability of the tests. For visualization of a parameters’ influence, samples of perturbed images in the lowest, a medium and the highest parameter configuration will be provided. As representative example, an image from the FVC2004 database DB1 (ID 91.2) will be used. For brevity, samples from the databases DB2 and DB3 will only be shown, if the StirMark manipulations lead to considerably differing results.

After this introduction and explanation of each test, I will present and discuss the experimental results. The discussion will have two main focuses: on the individual reaction of the matchers to the perturbations as well as on the comparison of respective performances. Due to the large amount of separate tests – per perturbation type, five matchers, on the images of three databases, times the number of various StirMark parameter configurations – I kindly ask for the readers understanding, that not all results can also be displayed graphically. As far as possible, I was trying to make a selection of those plots, that show the most interesting, most distinct results.

Comments on Notation

In current chapter the following abbreviations will be used to denote the different fingerprint matchers:

- *bozo3* – for the bozorth3 matcher
- *VF* – for the VeriFinger matcher
- *GF180* – for the GrFinger matcher with rotational alignment set to a range of $\pm 180^\circ$.
- *GF20* – for the GrFinger matcher with rotational alignment set to a range of $\pm 20^\circ$.
- *POC* – for the Phase Only Correlation matcher
- *FC* – for the FingerCode matcher

Furthermore, when using the term *DB1*, I will be referring to the accustomed version of the original FVC2004 database DB1, where the image dimensions were adjusted from originally 640×480 to 512×512 . Please refer to section 3.1.1 for details.

5.1 Original Images - Basis of Comparison

Before we examine the performance of the fingerprint matchers for StirMark perturbed images, it is necessary, to first establish a basis of comparison. In other words, we conduct the FVC-protocol compliant tests on the original “unperturbed” test images from the FVC2004 databases DB1 - DB3.

Following now, a discussion of the individual matching performances per database, together with a visualization of the results in form of receiver operating characteristic (ROC) curves. The corresponding equal error rates (EER) are listed in Table 5.1.

	bozo3 (%)	VF (%)	GF180 (%)	GF20 (%)	FC (%)	POC (%)
DB1	13.90	5.59	10.94	13.30	12.42	21.91
DB2	10.86	5.12	12.33	13.56	10.21	9.51
DB3	6.73	3.22	7.11	5.85	8.88	15.13

Table 5.1: The equal error rates EER for the array of fingerprint matchers used in present experiments, when applied to the original, “undistorted” sample image databases.

DB1

Regarding the ROC curves for DB1 in Figure 5.1, it is clearly observable, that here the **POC** matcher shows the worst matching performance of all considered matchers. Correspondingly Table 5.1 shows, that also the EER of the POC matcher is a full 8% higher than that of the second worst rate, of bozorth3.

bozorth3 itself does not perform all too well on DB1 either. For most of the considered range it shows the second worst results. Only in areas of low FMR ($< 8\%$) and higher FNMR ($> 17\%$), it exchanges places with the FingerCode matcher.

The **FingerCode** matcher shows interesting results: starting, as mentioned, with the second worst performance in areas of low FMR and high FNMR, it subsequently exchanges places with the other matchers, until, from a point of high FMR (about 47%) and low FNMR (about 2%) on, it even achieves the best results – in how far the matching performance at such high levels of false matches is still relevant for real-life applications is left to the reader to judge.

As for the GrFinger matcher, as expected, the setup with rotational alignment within a range of $\pm 180^\circ$ (**GF180**) quite clearly outperforms the setup, where rotational alignment is limited to $\pm 20^\circ$ (**GF20**). Only in an area of high FMR and low FNMR – from about the same point on, where FingerCode and VeriFinger exchange positions – the range-limited setup shows better results. Interestingly the FingerCode matcher, which is also limited to a rotational alignment range of $\pm 20^\circ$, beats GF20 already from a point on, where the FMR is still lower than the FNMR (FMR > 10% and FNMR < 15%) while GF180 only in areas, where the FMR is larger than the FNMR (FMR > 21% and FNMR < 8%).

Overall **VeriFinger** clearly shows the best matching performance and this only changes, as mentioned before, from a very high false match rate of > 47% on, when the FingerCode matcher has better results. Up until a FNMR of about 11% the VeriFinger matcher is able to perform on DB1 without any false matches – recorded in the so called *ZeroFMR* measure – while the second best matcher in that area, GF180, has a *ZeroFMR* of already 36%.

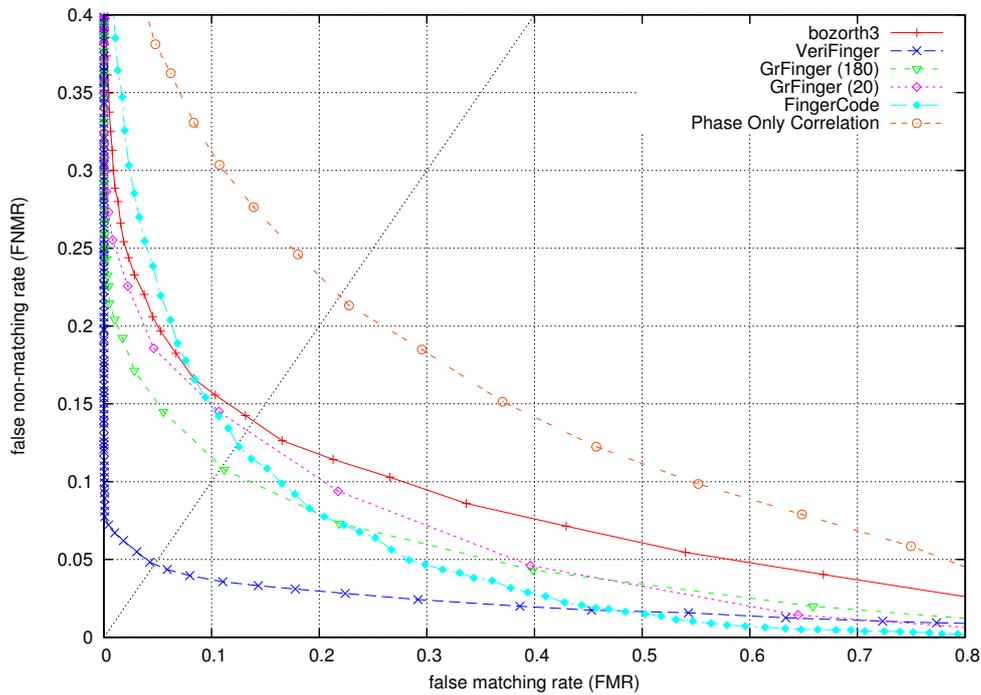


Figure 5.1: Receiver operating characteristics of the fingerprint matchers used in present work, applied to the test database DB1.

DB2

Compared to the results for DB1, the situation changes quite some, when regarding the ROC curves for fingerprint images of DB2 (Figure 5.2). Rather undisputed, **VeriFinger** once again exhibits the best results among the matchers and again only in the area of very high FMR ($> 57\%$) and very low FNMR ($< 2\%$) the FingerCode matcher manages to perform better. Like for DB1, also here in DB2, VeriFinger clear shows the best ZeroFMR characteristic: It operates without any false matches until a FNMR of about 9%, as compared to the second best ZeroFMR of 39%, again by GF180.

As for the GrFinger matcher: Here in DB2, for most of the considered range, the two configurations show the worst performance, with **GF180** once again producing slightly better matching results than **GF20**. For FMR close to 0 and FNMR $> 25\%$ the GrFinger matcher shows a behavior quite equal to that of the other matchers, then the GF20 rates get worse. At this point GF180 actually still has the second best results, until about a FNMR of 18%, when its performance in comparison to the other matchers drops as well .

The two non-minutiae-based fingerprint matchers produce surprisingly good results for the images of DB2. Regarding the EER, the **Phase Only Correlation** matcher has the second best, the **FingerCode** matcher the third best value. For the area of FMR $< 18\%$ and FNMR $> 7\%$ the POC outperforms FC, while FC is better in the complementary area (FMR $> 18\%$ and FNMR $< 7\%$).

For the largest part **bozorth3** performs decidedly better than GrFinger and at least equally good on an area of FMR close to 0 and FNMR $> 18\%$. On the other hand, from about the same point on, bozorth 3 shows performance mostly inferior to that of the non-minutiae-based matcher. Only at a FMR of about 57% and FNMR of about 3% POC and bozorth3 exchange position, while FC is still clearly better.

DB3

As with the results for images of DB1, the one thing that is most obvious, when regarding the ROC curves for DB3 in Figure 5.3, is, that clearly the **Phase Only Correlation** matcher shows the worst matching performance. Once again referring to the table of EER values (Table 5.1), we see a difference of about 6% to the second worst result by the **FingerCode** matcher. Another aspect that catches the eye when looking at Figure 5.3, is, that the ROC curves of all the other matchers are lying quite closely together – at least in comparison to the plots for DB1 and DB2.

The **VeriFinger** matcher decidedly exhibits the best matching performance, including the best ZeroFMR characteristic: at a FNMR of 5% versus a FNMR of 15% for the second best, **GF20**. And here is a very interesting detail in the results of DB3: GF20, the GrFinger matcher with a limited rotational alignment range of $\pm 20^\circ$, quite clearly shows the second best matching performance, while the unlimited version, **GF180**, is in parts even outperformed by the **bozorth3** matcher. In detail: for low FMR of $< 5\%$ and increasing FNMR of $> 7\%$ the ROC curve for GF180 lies between those of GF20 and bozorth3, while for the complementary ranges, i.e. increasing FMR of $> 5\%$ and FNMR of $< 7\%$, bozorth3 produces better results than GF180.

The FingerCode matcher once again exhibits the second worse performance and only manages to beat GF180 and subsequently also bozorth3 in an area of high FMR ($> 38\%$).

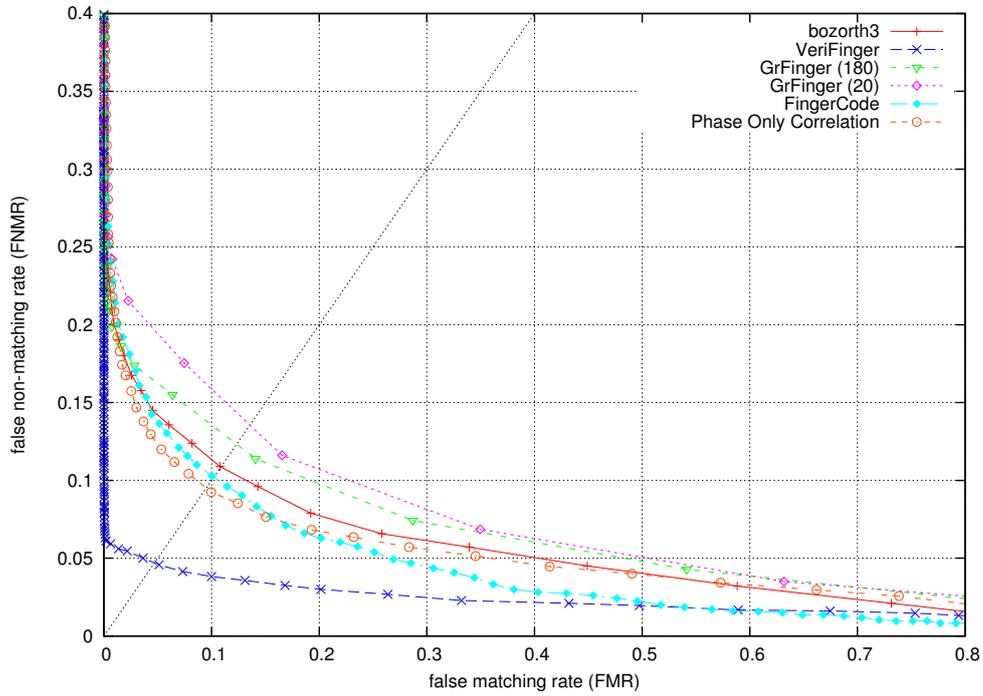


Figure 5.2: Receiver operating characteristics for database DB2.

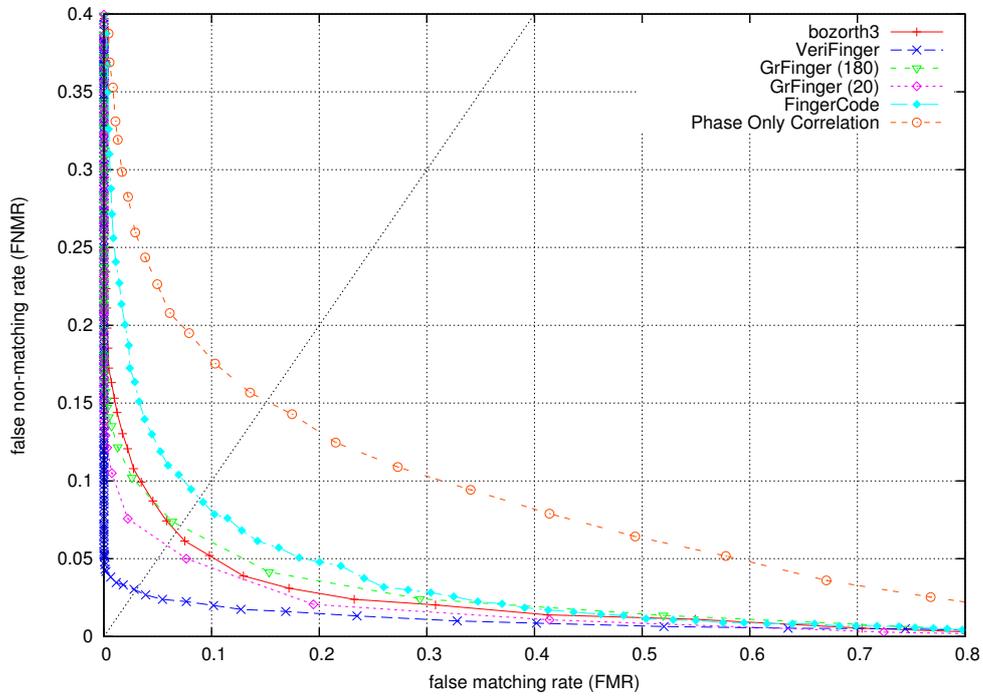


Figure 5.3: Receiver operating characteristics for database DB3.

General Observations

Comparing the ROC plots for all three databases, following observations can be made: Overall the VeriFinger matcher shows the best matching performance. It also stands out with a clearly superior ZeroFMR characteristic. An equally clear statement about the second best, third best, etc. performance is not easily possible, as notably the individual matching performances of the other matchers fluctuate differently strong, depending on the respective database. For example:

- The ROC curve for POC “jumps” from “clearly worst” in DB1 to “among top three” in DB2, then back to “clearly worst” in DB3.
- bozorth3, GF180 and GF20 exchange positions.
- Even the expected impairment of the matching performance of GrFinger, when limiting the angular range considered for rotational alignment, can only be observed in DB1 and DB2.
- Aside from VeriFinger, FingerCode seems to be subject to the least amount of fluctuation, as when comparing the ROC curves of FC for all three databases, the respective shape and location stays relatively constant, as opposed to that of the other matchers. An observation that is also backed up, when regarding the table of equal error rates. The difference between lowest and highest of the three values per matcher, is least for VeriFinger, followed by FingerCode and then by the others.

5.2 General StirMark Influence

Another question, that has to be dealt with before conducting the experiments on perturbed sample images, is: “Does the application of the StirMark Benchmark by itself already have any influence on the results of the regarded fingerprint matchers?”

As mentioned in section 2.1, when running a StirMark test on a sample image – the means, by which we introduce a specific, predefined perturbation into the fingerprint images – the benchmark also automatically embeds a theoretically imperceptible watermark “*with the greatest strength which does not introduce annoying effects*” [20]. While essential for the robustness test of watermarking techniques that the StirMark Benchmark was originally designed for, here, for the experiments of present work, the particular embedded watermark is of no interest. Yet the question, that *is* of relevance to us in this context is, if the watermark or the embedding process itself, has any influence on the matching results of the individual fingerprint matchers.

To get to an answer, I applied a StirMark test, employing a *convolution* attack to the images of the three fingerprint databases. In order to single out the influence on the sample images caused by the StirMark Benchmark itself, the convolution was performed using a filter mask of the form:

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Therefore the convolution operation, in theory, has no effect on the images, hence any changes that appear, can be contributed to the benchmark process itself.

Results and Discussion

	bozo3 (%)	VF (%)	GF180 (%)	GF20 (%)	FC (%)	POC (%)
DB1	13.90	5.59	10.94	13.30	12.42	21.91
marked DB1	14.81	5.87	11.41	13.61	12.54	22.60
DB2	10.86	5.12	12.33	13.56	10.21	9.51
marked DB2	11.12	5.01	11.72	12.89	9.60	9.69
DB3	6.73	3.22	7.11	5.85	8.88	15.13
marked DB3	6.68	3.60	6.90	5.82	8.98	15.07

Table 5.2: Comparing equal error rates of the fingerprint matchers when operating on the original fingerprint images (DB1 – DB3) and on images processed by a basic StirMark test without purposefully generated perturbations (marked DB1 – marked DB3).

As can be seen in Figure 5.4, matching results show, that the application of a StirMark test actually *does* have an influence on the matching performance of the various fingerprint matchers. The deviation is comparatively small, so for better perceptibility a logarithmic scale was used in the plots.

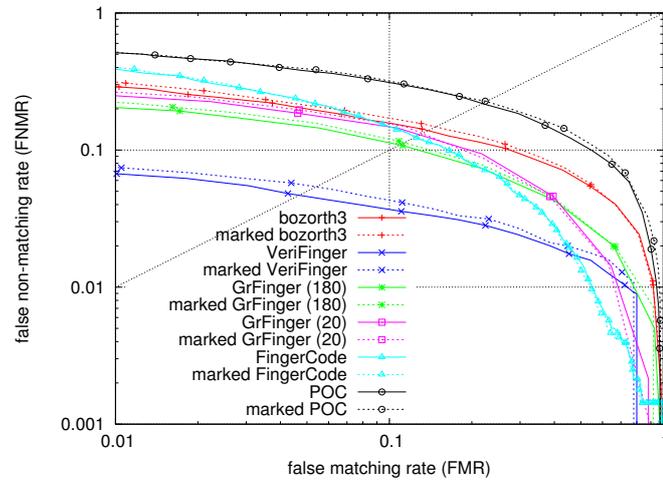
Comparing the results for the three image databases, we notice, that StirMark processing has seemingly the largest impact on images of DB1, with a general tendency to worsen the matching performance. Here only the FingerCode matcher appears to be largely indifferent to the changes.

For DB2, the noticeable differences are less distinct than for DB1, yet interestingly we can even observe a slight improvement in the results of the GrFinger matcher in both configurations, as well as for the FingerCode matcher. As can be found in Table 5.2, this behavior is also reflected in the corresponding EER.

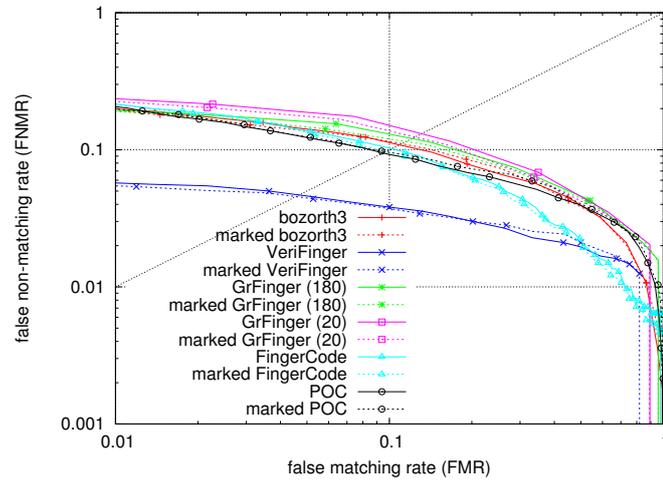
The matching results for images of DB3 show the least response to the StirMark manipulation. Regarding Figure 5.4c, we see that the ROC curves, representing the results of matching the StirMark processed images, get to lie almost on top of those stemming from the original database images – for all matchers but VeriFinger. Here, for DB3, VeriFinger exhibits a comparatively strong impairment of its matching performance, when dealing with the StirMark manipulated images.

Trying to figure out, where this small, yet definitely observable alterations in the matching results for the StirMark manipulated sample databases come from, closer inspection of the involved images leads to following observation: When subtracting an original fingerprint image from its StirMark processed version, a difference image akin to the example for a DB1-image shown in Figure 5.5 is created. (For clarity and to simplify the recognition of details, the image values in Figure 5.5 have been normalized to a range of [0,255]).

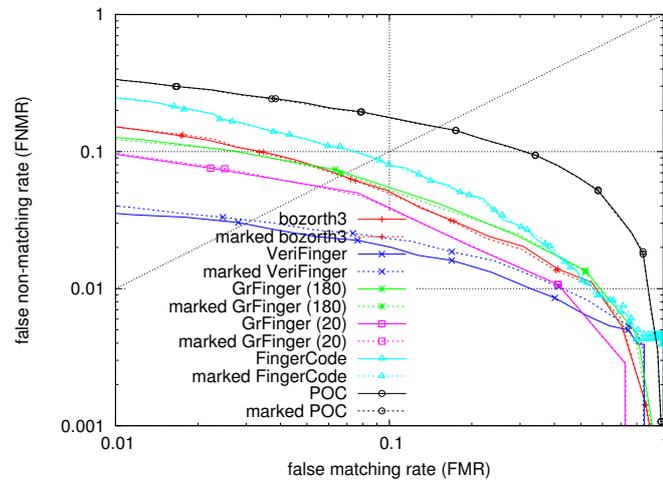
When regarding said example image, several isolated spikes can be detected (most easily discernible in the black background area). In all likelihood this noise represent the traces of the embedded watermark and was to be expected. Something that was *not* to be expected though, is, that we also see the clear traces of the actual finger imprint. While shown in the enhanced example image in a gray-level value of 255, in the actual difference image the fingerprint traces have a constant value of 13. After further inspecting difference images for databases DB2 and DB3, we get to following realization:



(a) Receiver operating characteristics for DB1 (logarithmic scale)



(b) Receiver operating characteristics for DB2 (logarithmic scale)



(c) Receiver operating characteristics for DB3 (logarithmic scale)

Figure 5.4: Figures a), b) and c) show the differences in matching performance of the individual fingerprint matchers, between using the original sample images as provided in the FVC2004 databases and using images that have been processed once by a StirMark test (yet without deliberately introducing image perturbations).

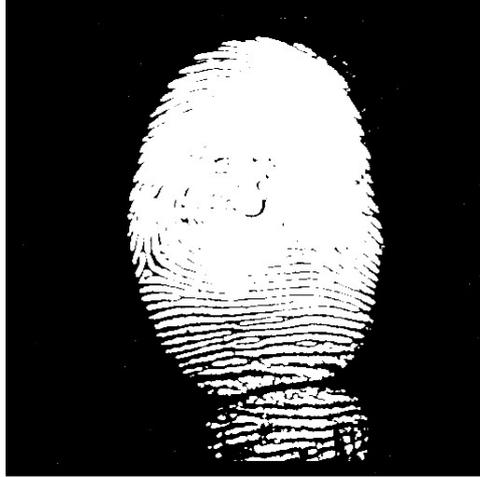


Figure 5.5: Example difference image created, when subtracting the original fingerprint image (present example taken from DB1) from the StirMark processed version. (For better visibility, the image values have subsequently been normalized to $[0,255]$). Two kinds of residuals can be observed: Sparsely distributed spike noise (seen for example close to the right border) and a trace of the finger imprint itself, with a constant gray-level of 13.

The StirMark test not only embeds a watermark into a processed image, but also brightens the entire image by a value of 13. Yet it has to be noted, that the maximum gray-level value in an output image is still 255, hence the brightening caused by StirMark leads to a loss of information for any image values within the range of $[243,254]$, as these can not be proportionally raised. An observation that leads me to the conclusion, that it is probably this brightening feature of a StirMark test, that causes the deviation of the matching results, as the sparsely distributed spike noise should, in theory, not have any real influence on the performance of the fingerprint matchers, especially not on that of the minutiae-based ones.

Besides affecting the matching results, the brightening of image values also has a second consequence: It affects/limits the possibility of combining several different perturbation types, as the repeated application of the StirMark tests also leads to consecutive image brightening, going along with potential loss of image information.

5.2.1 Consequence

In consequence of the detected alterations of the overall matching results, introduced alone by processing the fingerprint images with the StirMark Benchmark, henceforth these results will be used as basis for comparison in the subsequent experiments using StirMark perturbed images, instead of the matching results created, when performing the FVC-conform performance evaluation for the original, “unprocessed” fingerprint images of databases DB1-DB3.

5.3 Additive Noise (noise)

The *Additive Noise* test simply adds an adjustable amount of random noise to the input image.

Relation to Fingerprints

This test is intended to simulate noise, that might naturally appear in fingerprint images. Possible causes for this kind of noise could for example be actual dust on the contact area during acquisition of the imprint, graining caused by the acquisition equipment itself or other kind of systematic errors introduced during processing, transmission and/or storage of the collected images.

Parameter Configurations

The amount of noise that will be added by the StirMark test is adjusted using a single parameter, ranging from 0 to 100. A parameter value of 0 results in an identical image while a value of 100 results in a completely random one.

Following parameter values are used in the experiments of present work: {3, 5, 7, 9, 11, 13, 15}. As noted on page 18 within section 2.3.1, only a reduced number of tests will be executed on the non-minutiae-based fingerprint matchers, due to the enormous amount of necessary processing time. This reduced selection covers parameter values {3, 7, 11, 15}. Noise for a value < 3 is barely noticeable in the images, while values > 15 create an amount of noise, that I would hardly consider “natural” any more. Examples for the noise levels 3, 9 and 15 are given in Figure 5.6

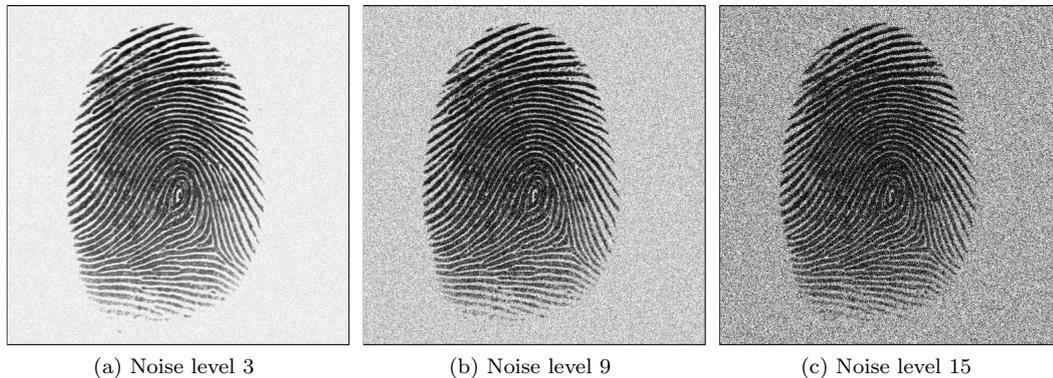


Figure 5.6: Examples for the *Additive Noise* test, applied to an image from DB1 (ID 91_2).

Results and Discussion

DB1

When gradually introducing random noise into the fingerprint images, the matcher, whose matching performance is influenced the strongest, is GrFinger (with both rotational alignment configurations exhibiting about the same degrees of impact, **GF180** only slightly stronger than **GF20**). As shown in the corresponding Figure 5.7a, the ROC curves are very clearly separated, indicating the continuing degradation of the matching results with each level of increased noise intensity. When referring to Table 5.3 this behavior can also be asserted in the respective equal error rates: From “unperturbed” to “Noise Level 15” the EER

of GF180 gradually deteriorates up to 8.56% (6.21% for GF20).

For the **bozorth3** matcher, the influence of random noise in the fingerprint images is already weaker than for the GrFinger matcher, yet we can still observe an obvious spreading of the ROC curves. One interesting point though is, that for weak to medium noise levels (levels 03 to 09) an increase in the amount of noise does not necessarily cause a correlative decrease of matching performance: Within the observed range, the corresponding ROC curves intersect each other and change “positions”. For example: While expecting the third noise-level in the experiments (level 07) to cause worse matching results than the two weaker levels (03 and 05), this is only true for an area of $FMR > 21\%$ and $FNMR < 14\%$. Up until this point, the ROC curve for level 07 indicates less impairment of the matching performance than is caused by the two weaker noise levels.

Regarding the plot of the ROC for the **VeriFinger** matcher in Figure 5.7b, we see, that only for the two strongest levels of random noise within the fingerprint images (levels 12 and 15), the matching performances is intelligibly impaired. The ROC curves originating from the weaker noise levels are lying very close together and are likewise close to that for the unperturbed fingerprint images.

Overall, for fingerprint images of DB1, the matching performances of the non-minutiae-based fingerprint matchers are very little influenced by additive random noise. In fact, as can be seen in Figure 5.7c, the ROC curves for the **FingerCode** matcher lie comparatively close together and also Table 5.3 shows, that the difference between the EER for the unperturbed images and the EER for the strongest noise level is only 5.3%.

Figure 5.7d portrays the ROC curves for the matching results of the **Phase Only Correlation** matcher. Also here the curves lie relatively close together. Yet a very interesting observation is, that only the strongest noise level (15) produces slightly worse matching results than the unperturbed images. For the other three regarded noise levels, the matching performance actually improves, with best results here caused by noise level 07. As for why the POC matcher produced better matching results for the theoretically impaired fingerprint images of DB1 than for the original uninfluenced ones, is still subject to further investigation.

Noise Level	bozo3 (%)	VF (%)	GF180 (%)	GF20 (%)	FC (%)	POC (%)
unperturbed	14.81	5.87	11.41	13.61	12.54	22.60
03	16.07	5.76	11.86	13.43	12.97	21.89
05	16.61	6.05	13.03	14.07		
07	16.27	5.76	13.37	14.15	12.99	20.68
09	17.19	5.82	13.84	15.43		
11	17.45	6.42	16.23	16.32	12.65	21.18
13	19.38	6.65	17.92	17.99		
15	20.27	7.76	20.06	19.82	13.01	22.96

Table 5.3: Equal error rates for *Additive Noise* test conducted on sample image database DB1.

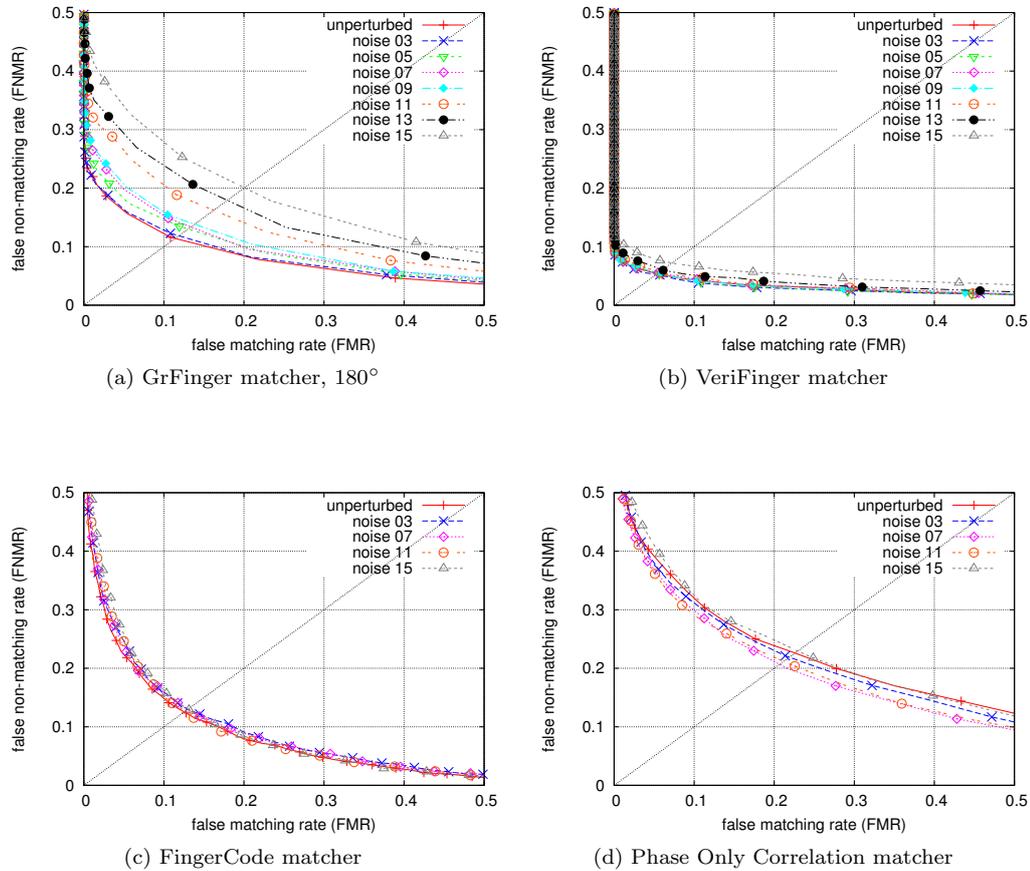


Figure 5.7: Receiver operating characteristics for StirMark test *Additive Noise* on fingerprint images of DB1.

DB2

For fingerprint images of DB2, the introduction of additive noise leads to slightly different effects in the matchers' performance:

Once again the **GrFinger** matcher is clearly influenced. As can be seen in Figure 5.7a, the ROC curves corresponding to the individual noise-levels are clearly separated and also notably distant from the ROC curve indicating the matchers performance for the original, unperturbed fingerprint images. As would normally be expected, the results show, that an increase in added random noise leads to an according decrease in matching performance.

Somewhat different from the results for fingerprint images of DB1, the **bozorth3** matcher and even the **POC** matcher exhibit an impairment of their matching performance, which is quite similar to that for the GrFinger matcher. Only for the two lowest noise levels (03 and 05) bozorth3 shows comparatively better results: Just mildly impaired matching performance, with noise level 03 even slightly outperforming the test run on unperturbed images in an area around the EER. (Please refer to Figure 5.8a and Table 5.4 for details).

As for the **VeriFinger** matcher, rather contrary to the observations for fingerprint images

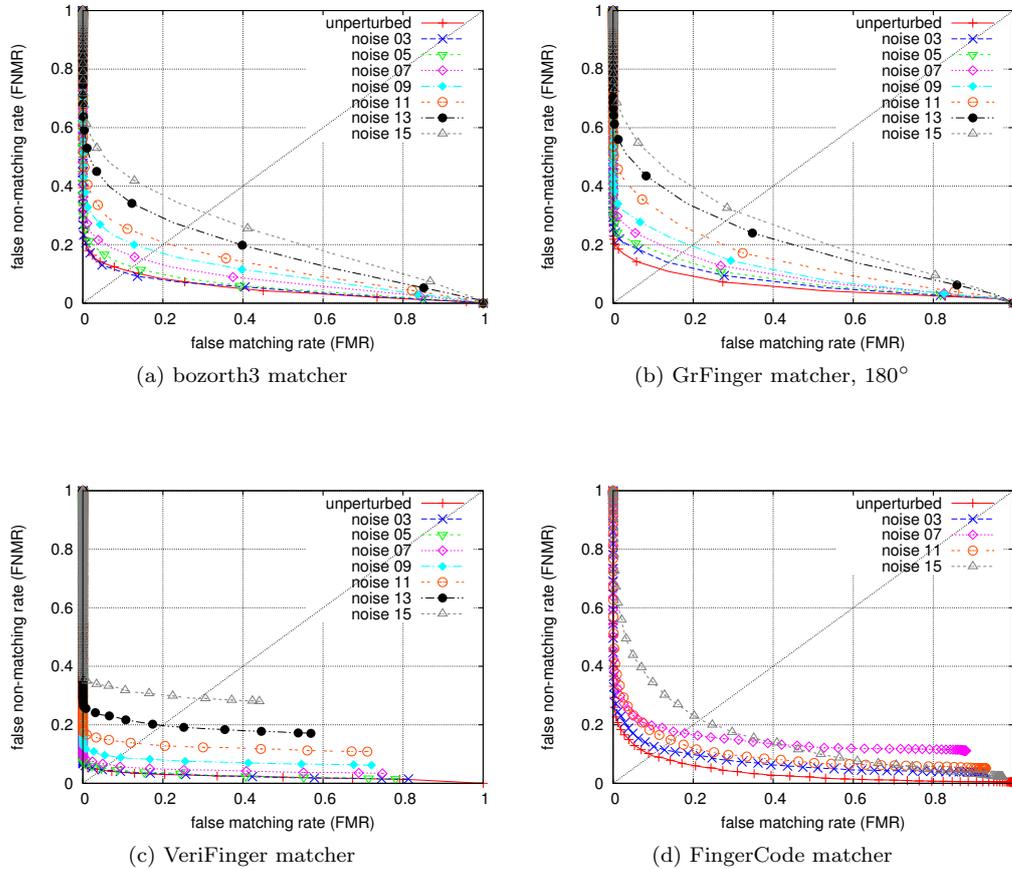


Figure 5.8: Receiver operating characteristics for StirMark test *Additive Noise* on fingerprint images of DB2. (As for the particular appearance of Figures 5.8c and 5.8d, please also refer to the “Notes on ROC plots” on page 10).

of DB1, Figure 5.8c displays, that additive noise in images of DB2 has a distinct impact on the matchers’ performance. For noise levels 03 and 05 no separation of the ROC curves can be determined. For the subsequent higher noise levels on the other hand, the curves are clearly separated, depicting a progressive impairment of the matching performance with rising amount of noise. In particular, examining for example the results of VeriFinger for fingerprint images with added noise of level 15, we see that already 46% of all match-pairs have a matching score of 0. Consequently, in the rate analysis, with the first increment of the threshold value, we obtain a FMR of 44% and a FNMR of 28%.

Among all the tested matchers, **FingerCode** seems to, once again, be the least influenced by the added noise, even though the impairment of the matching performance is already stronger than in the DB1 case. Interesting is also, that here stronger noise does not automatically correlate with worse results. As can be seen in Figure 5.8d, noise level 11 leads to clearly better results than noise level 07. And while noise level 15 expectedly causes the worst matching results over most of the regarded value range, at a point of FMR of 42% and FNMR of 14% its ROC curve starts intersecting the other curves. In other words, for high FMR noise level 15 causes less false non-matches than added noise of lesser density.

Noise Level	bozo3 (%)	VF (%)	GF180 (%)	GF20 (%)	FC (%)	POC (%)
unperturbed	11.12	5.01	11.72	12.89	9.60	9.69
03	10.86	5.05	13.90	14.90	11.85	10.65
05	12.40	5.43	15.48	15.76		
07	15.03	7.07	17.27	17.91	17.22	14.36
09	17.90	9.50	19.63	19.08		
11	20.54	14.20	22.89	21.56	14.74	20.22
13	25.99	20.74	28.10	26.44		
15	30.78	29.24	31.17	29.40	21.80	26.94

Table 5.4: Equal error rates for *Additive Noise* test conducted on sample image database DB2.

DB3

Now, comparing with the results for DB1 and DB2, how does the introduction of random noise into fingerprint images of database DB3 influence the respective performance of the fingerprint matchers?

Both configurations of the **GrFinger** matcher show only a light impact in their matching results. Generally the ROC curves for the individual noise levels lie close together. Mostly around the area of the EER a moderate separation of the curves is observable. Inspecting this area closer, we can see, that the matching results basically follow our expectation: Gradually worsening matching performance per increased noise level. The corresponding plot for GrFinger with the rotational alignment range of 180° can be seen in Figure 5.9a.

Roughly the same can be said about the performance of the **bozorth3** matcher. Even though the ROC curves lie slightly closer together and the separation around the area of EER is decidedly less distinct. Additionally, the test for noise level 11 exhibits a surprisingly good matching performance: After noise level 03, it produces the second best results of the noise-tests. This can also be seen in the respective EER rates: 6.68% for unperturbed images, 7.05% for noise level 03 and 7.08% for noise level 11. (For further equal error rates, please refer to Table 5.5).

The **VeriFinger** matcher shows to be very robust against additive noise in images of DB3. The ROC curves for the different noise levels appear to almost lie on top of each other. Only by zooming in closely to the area of the EER (see Figure 5.10), we can reveal an interesting behavior: Though the differences are only comparatively marginal - a maximum difference in EER of 0.57% - the unperturbed fingerprint images produce the worst results, closely followed by images of noise level 15. The best results, on the other hand, could be obtained for noise level 13, followed by level 07. These observation can also be confirmed, when inspecting the corresponding EER values in Table 5.5. So obviously, in the case of VeriFinger, increased noise in the fingerprint images of DB3 does not correlate with decreased matching performance.

Turning to the non-minutiae-based fingerprint matchers, this time it is the **Phase Only Correlation** matcher, that is less influenced by the added noise in the fingerprint images: The ROC curves for the lower noise levels almost fall together with that, produced by unperturbed images. The ROC curve for the results of noise level 11 then, lies slightly

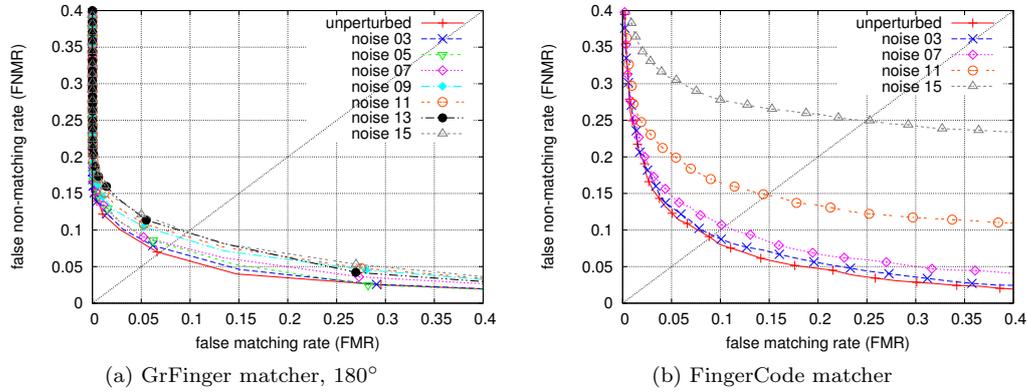


Figure 5.9: Receiver operating characteristics for StirMark test *Additive Noise* on fingerprint images of DB3.

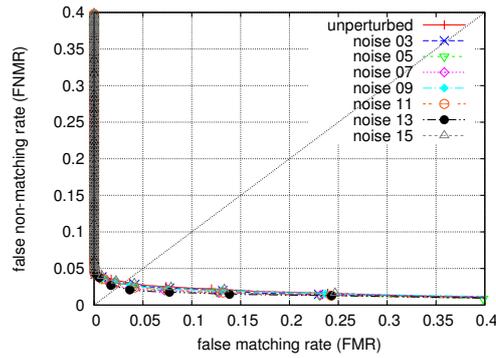


Figure 5.10: Close detail of the receiver operating characteristics of matcher VeriFinger for StirMark test *Additive Noise* on fingerprint images of DB3. The ranges are set to FMR [0%, 6%] and FNMR [1.5%, 5%].

above aforementioned ones and finally, the ROC curve caused by noise of level 15 is clearly separated from the others (see also the EER values in Table 5.5).

The response of the **FingerCode** matcher, on the other hand, stands in strong contrast to the behavior displayed for noisy images of databases DB1 and DB2: While so far the FingerCode matcher appeared to be the matcher the most robust to noise, here, for fingerprint images of DB3, it exhibits the strongest impairment of matching performance among the whole set of considered fingerprint matchers. Regarding Figure 5.9b, we see, that the ROC curves are clearly separated. Each level of increased noise causes a decrease in matching performance. Yet this decrease is not linear, but from the ROC curve representing the results for unperturbed images on, the gap between the curves of the individual noise levels constantly widens. An effect, also mirrored in the corresponding EER values (see Table 5.5 for details).

Noise Level	bozo3 (%)	VF (%)	GF180 (%)	GF20 (%)	FC (%)	POC (%)
unperturbed	6.68	3.60	6.90	5.82	8.98	15.07
03	7.05	3.25	7.43	6.09	9.25	15.28
05	7.21	3.23	7.92	6.12		
07	7.19	3.03	8.06	6.59	10.50	15.16
09	7.31	3.41	8.97	7.37		
11	7.08	3.29	9.28	7.86	14.79	15.71
13	7.99	3.11	9.69	8.08		
15	7.91	3.25	9.78	8.14	24.99	17.46

Table 5.5: Equal error rates for *Additive Noise* test conducted on sample image database DB3.

5.4 Median Cut Filtering (medianCut)

This test applies a median cut filter of adjustable size on the input image. I.e. per pixel p of the input image, the median of all image values within an area of pre-selected size, centered at p , is established, to be returned as value of the corresponding pixel in the output image.

Relation to Fingerprints

The *Median Cut Filtering* attack is employed in present experiments to simulate smudgy fingerprints, as they are common in real-life applications, for example when the fingertip is too moist during the acquisition via a scanner. Compared to the *Mean Filtering* attack (5.5), *Median Cut Filtering* likewise adds a certain amount of blur to the image, but additionally it also corrupts the clarity of the ridge-and-furrow structure of the imprint, inducing the desired smudginess. This effect is strongest in the interior of the imprint.

Parameter Configurations

Via a single parameter the size of the median cut filter – and therefore the intensity of its effect – can be adjusted. Height and width of the filter take the same value and only odd-valued dimensions are accepted. The upper limit is a size of 15, thus resulting in a 15×15 filter.

For the experiments of present work, following parameter values have been chosen: $\{3, 5, 7, 9\}$. A filter size < 3 obviously is of no real benefit. On the other hand, values > 9 have been excluded, as in these cases the median cut filter leads to the creation of spurious ridges, even with higher frequency than the original ones. As interesting as this effect might be, as little it complies with our demand for “natural” perturbations. Examples for filter sizes 3, 7 and 9 can be seen in Figure 5.11.

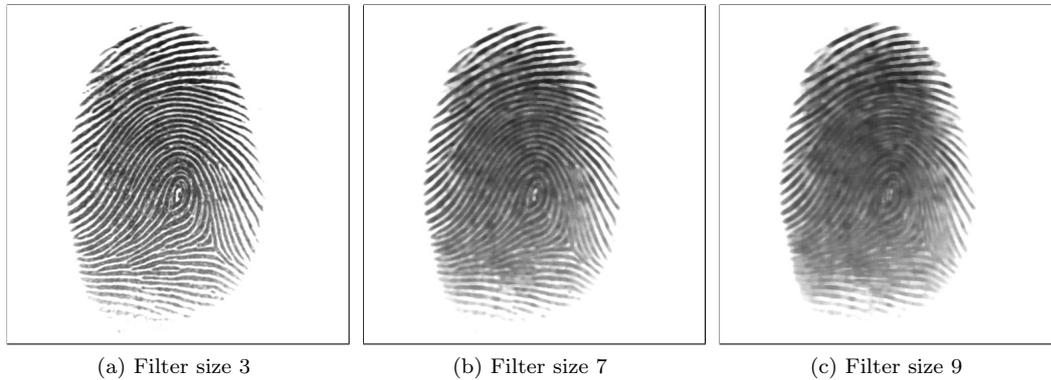


Figure 5.11: Examples for the *Median Cut Filtering* test, applied to an image from DB1 (ID 91.2).

Results and Discussion

DB1

Of the array of fingerprint matchers employed in the *Median Cut Filtering* test for images of database DB1, **VeriFinger** is the one matcher, whose performance clearly is influenced the most by the “smudge-like” perturbations introduced. Regarding Figure 5.12b, we see, that for a 3×3 filter, the resulting ROC curve lies close to the one for unperturbed images. When applying a filter of size 5×5 , the ZeroFMR value rises from 14% to 27% and for increasing FMR, the FNMR values lie continuously above those of the previous curve. The big “drop” in matching performance then occurs for a filter-size of 7×7 : The ZeroFMR value goes up to 75%, yet compared to the results of the other matcher even more drastic: the EER value changes from 12.82% to 40.42% (see also Table 5.6). In comparison: GF20, for example, has an EER of 40.19% only for a filter-size of 9×9 (GF180: 44.23%) and an EER of just 27.52% for the 7×7 filter (GF180: 31.65%). Finally the ROC curve, representing VeriFinger’s results for the test with a 9×9 filter, lies clearly above those, produced by all the other matchers for median cut filtered fingerprint images of database DB1. Interestingly the curve also is of convex shape, i.e. in comparison, the FNMR per FMR are particularly high.

Looking at **GrFinger**, we can clearly observe the influence of the introduced image perturbations in the matching performance (see also Figure 5.12a). The ROC curves representing the individual median cut filter sizes are well separated and, as was expected, the higher the degree of perturbations, the worse the results. For the first two levels, the impairment of the matching performance is somewhat comparable to the one observed with VeriFinger: A filter size of 3×3 causes the ROC curve to lie only slightly above the curve produced by the unperturbed images. The curve for the 5×5 filter, is comparatively close, but clearly separated. The gaps to and between the following two ROC curves then, representing the results for fingerprint images filtered with filter of sizes 7×7 and 9×9 , are considerably larger, yet interestingly quite equal – regarding the equal error rates in Table 5.6, we find a distance of 12.81% between the EER of tests medianCut 05 and 07 and a distance of 12.58% between the EER of tests medianCut 07 and 09.

A further point that can also be verified by inspecting the respective EER values is, that overall the results of GF20 – GrFinger with rotational alignment restricted to only $\pm 20^\circ$ – are better, than the results of GF180.

On first glance, the ROC plot for the matching results of **bozorth3** looks quite similar to the ones, produced by the GrFinger matcher configurations. On closer inspection, we find, that bozorth3 is slightly more robust to the perturbations introduced by the median cut filtering test, than GrFinger: For one, the separation between the ROC curves for the application of 3×3 and 5×5 filters is considerably smaller. Further, while the ROC curves for tests medianCut 07 and 09 have nearly the same properties as for GF180 (also the EER values are close), we have to consider, that the basis of comparison (i.e. the matching performance for unperturbed images) of GrFinger is superior to the one of bozorth3.

Generally we can establish, that the **non-minutiae-based fingerprint matchers** are clearly less influenced by “smudginess” introduced in fingerprint images of database DB1, than the minutiae-based ones: As can be seen in Figure 5.12c for the FC matcher and in Figure 5.12d for the POC matcher, the overall separation of the ROC curves is evidently smaller than in the previously discussed matchers.

As for the **Phase Only Correlation** matcher, the ROC curves corresponding to filter-sizes 3×3 and 5×5 lie relatively close to the curve for unperturbed images, the relative distances quite similar to the results for bozorth3. Subsequently we have two larger “jumps” to the graphs for filter-sizes 7×7 and 9×9 . How much less the POC matcher is influenced by the median cut filtering of fingerprints of DB1 than the minutiae-base matchers, can also be observed in the equal error rates in Table 5.6: For instance the distance between the EER for unperturbed images and the one caused by filtering with a filter of size 9×9 , is 15.15% for POC, while 26.58% for GF20, 32.82% for GF180, 32.07% for bozorth3 and even 50.21% for VeriFinger.

Concerning the first three levels of perturbation-intensities, the **FingerCode** matcher displays an even better matching performance (see also Figure 5.12c): The ROC curves for both filter-sizes, 3×3 and 5×5 lie nearly on top of each other and also the curve related to filter-size 7×7 lies so close, that also its EER is only 4.01% higher than that for unperturbed images (in comparison: 8.11% for POC, followed by 13.91% for GF20). The influence of filter-size 9×9 on FingerCode’s matching performance is relatively strong, when compared to the influence of the smaller filter-kernels. Yet still the results get only about as “bad”, as for the POC matcher.

Filter Size	bozo3 (%)	VF (%)	GF180 (%)	GF20 (%)	FC (%)	POC (%)
unperturbed	14.81	5.87	11.41	13.61	12.54	22.60
03	15.50	7.09	13.68	14.52	12.90	23.63
05	17.69	12.82	18.84	17.86	13.52	24.92
07	32.17	40.42	31.65	27.52	16.55	30.71
09	46.88	56.08	44.23	40.19	28.26	38.11

Table 5.6: Equal error rates for *Median Cut Filtering* test conducted on sample image database DB1.

DB2

Regarding the results of the *Median Cut Filtering* tests for fingerprint images of database DB2, especially in comparison to the previous observations for images of DB1, we can say, that for the **minutiae-based fingerprint matchers**, the results in general look quite alike

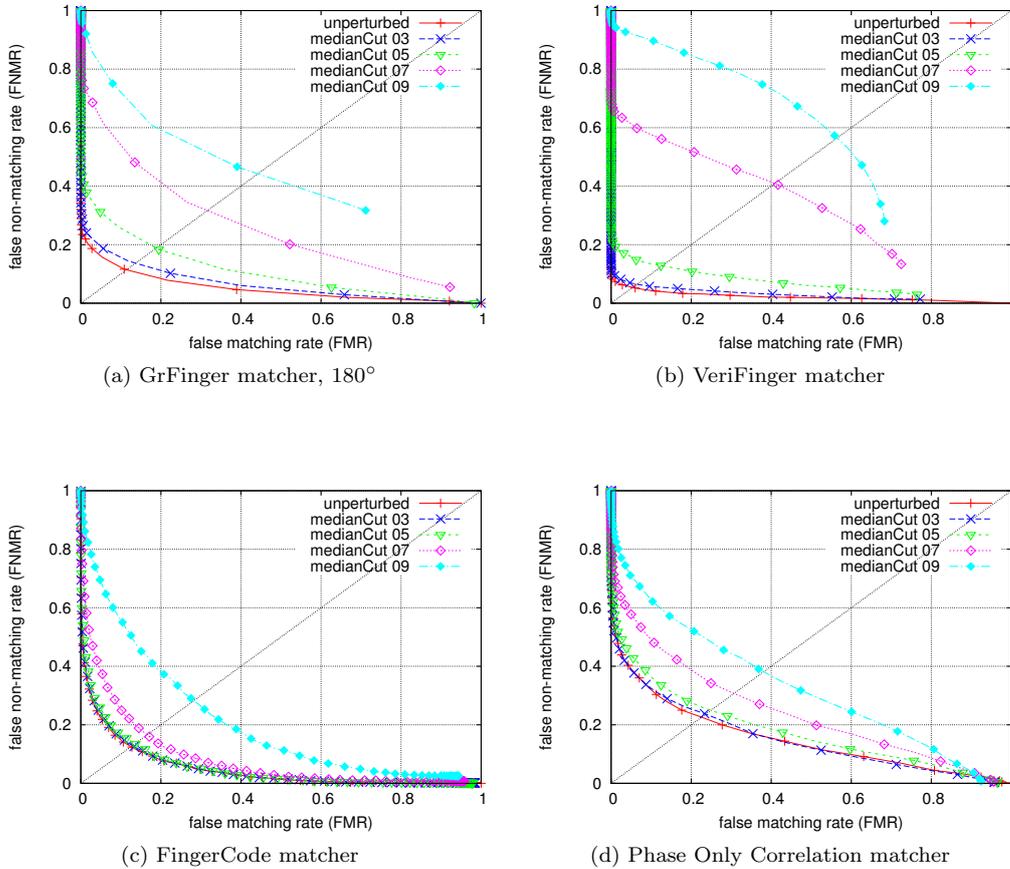


Figure 5.12: Receiver operating characteristics for StirMark test *Median Cut Filtering* on fingerprint images of DB1. “median Cut 03 .. 09” refers to the size of the median cut filter.

those for DB1: Filtering with a filter-size 3×3 has only a minor influence on the matching performance. Filtering with a 5×5 filter affects the matching results already more, yet the really drastic impairment of the matching performance comes with the employment of the filters of sizes 7×7 and 9×9 , where the results for the latter filter configuration are roughly twice as bad, as those of the former. Inspecting said results a little bit more detailed:

For **VeriFinger**, the impairment caused by median-filtering the fingerprint images of DB2 with a filter of size 7×7 is slightly less than for images of DB1, while the situation is clearly contrary in case of the 9×9 filter. This behavior can also be witnessed in the respective equal error rates: Regarding the difference from the EER for unperturbed images to the EER for medianCut 07 we get 34.55% for DB1 and 33.23% for DB2, while to the EER for medianCut 09 we get 50.21% for DB1 and 56.84% for DB2.

In case of **bozorth3**, the influence of perturbed images from DB2 is notably stronger than that of images from DB1. All ROC curves relating to the filtered images lie comparatively farther above the curve for unperturbed images, with the respective deviation even slightly increasing per step in filter-size.

The same can be said, regarding the matching performance of both **GrFinger** configura-

tions, for the two larger filter settings 07 and 09. In particular, in case of the 9×9 filter, the results are so bad, that in analysis, already with the first incrementation of the threshold, the rate values rise to about 47% FMR and 64% FNMR for GF180 and 38% FMR and 66% FNMR for GF20. For fingerprint images of DB2 filtered with the filters of sizes 3×3 and 5×5 on the contrary the matching performance is slightly superior than in the results for DB1 images.

Comparing the performance of the **non-minutiae-based** fingerprint matcher in response to the “smudgy” perturbations in fingerprint images of DB2 with the results achieved for images of DB1, we can observe some major differences:

Of all the matchers employed in the experiment, it is now the **Phase Only Correlation** matcher, which is clearly the most robust to the median cut filtering. In Figure 5.13d we see, that especially the ROC curves representing the tests with the two smaller filter-sizes lie closer together now, and also the gap to the ROC curve for the 7×7 filter got somewhat reduced. The distance between the ROC curve for unperturbed images to the curve for medianCut 09, on the other hand, got a little larger, which we can also see in the difference of the respective equal error rates: 16.54% in present case, compared to 15.51 for images of DB1.

More drastic are the changes in the results for the **FingerCode** matcher, as can be seen in Figure 5.13c: The ROC curve for filter-size 3×3 still lies almost on top of the curve for unperturbed images and also the performance for images filtered with the 5×5 filter dropped only relatively little. Yet the distances to the ROC curves of medianCut 07 and furthermore medianCut 09 increased largely. Although the curve related to the 7×7 filter is still below the corresponding ones of the minutiae-based matchers, the ROC curve for filter-size 9×9 indicates a respective matching performance about as bad as the one of GrFinger – potentially worse, as the rate values immediately jump from the initial 100% FMR and 0% FNMR to 38% FMR and 68% FNMR.

Filter Size	bozo3 (%)	VF (%)	GF180 (%)	GF20 (%)	FC (%)	POC (%)
unperturbed	11.12	5.01	11.72	12.89	9.60	9.69
03	12.17	5.97	13.06	14.13	9.42	11.03
05	18.22	11.44	17.75	17.19	10.95	11.38
07	37.36	38.34	35.08	33.66	24.94	16.94
09	51.97	61.85	–	–	–	26.23

Table 5.7: Equal error rates for *Median Cut Filtering* test conducted on sample image database DB2.

DB3

Now turning to fingerprint images of DB3, we can observe, that for the **minutiae-based matchers**, in general, the influence of the median cut filtering on the matching performance is not as strong, as it was the case for images of DB1 and DB2. And once again the reactions of the **non-minutiae-based matchers** is different from the behavior witnessed so far.

Both GrFinger configurations show the same kind of overall performance, even though the relative deviations are smaller for **GF20** than for **GF180**. For the filter-size of 3×3 the results are about as good as for unperturbed images. In the area around the equal error rate,

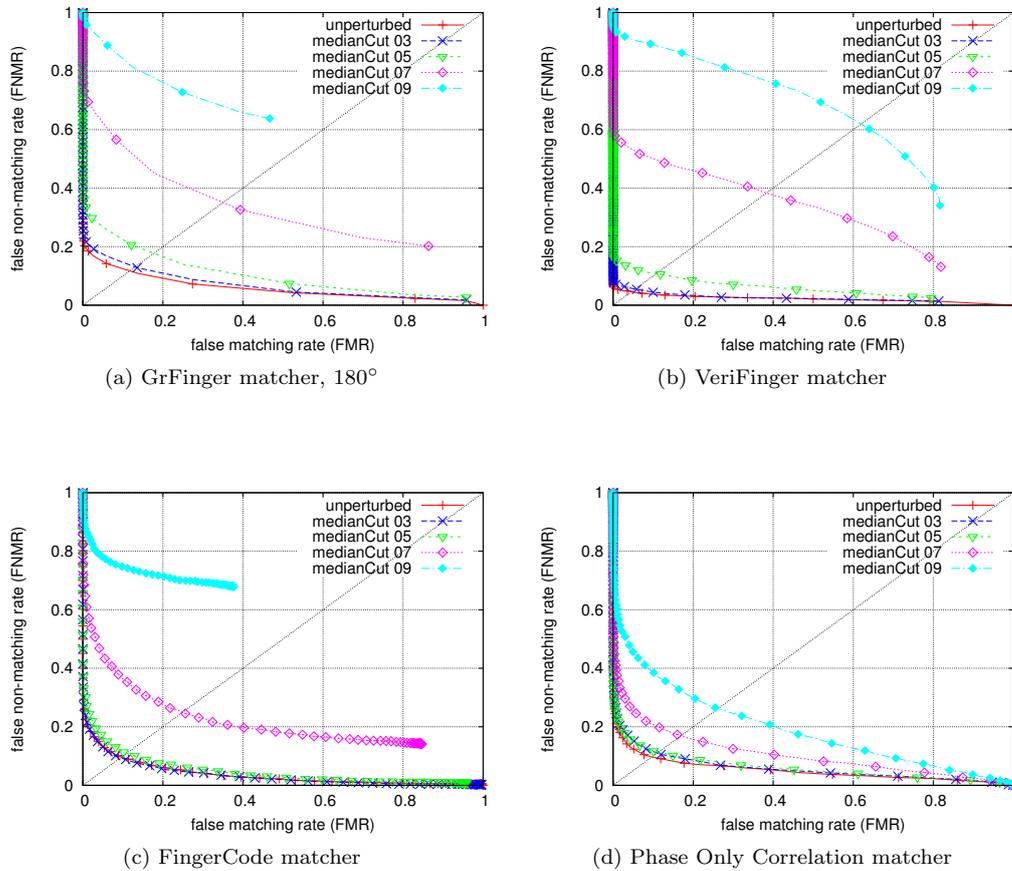


Figure 5.13: Receiver operating characteristics for StirMark test *Median Cut Filtering* on fingerprint images of DB2. “median Cut 03 .. 09” refers to the size of the median cut filter.

the filtered images even produce slightly better results (also compare to Figure 5.14a). The impairment caused by filtering with a 5×5 median cut filter is already clearly evident yet still relatively minor. Again, the drops in matching performance get increasingly larger for filter sizes 7×7 and 9×9 . For GF180 and medianCut 09 for example, the EER then lies at 30.29%, which results in a difference from the EER for unperturbed fingerprint images of 23.39% – for comparison, this distance was 32.28% for images of DB1 and for DB2, the number of false non-matches was constantly so large, that no point of equal error could be reached.

The ROC plot displaying the results for **bozorth3** looks quite similar to that of GF180, except that the relative distances of the individual ROC curves are slightly larger – a behavior that can also be verified in the respective EER values in Table 5.8.

Corresponding to aforesaid general observation, also the overall matching performance of **VeriFinger** improves for images of DB3, as compared to its previous results for fingerprint images of DB1 and DB2. Still, among the minutiae-based fingerprint matchers, VeriFinger stays the one, displaying the strongest sensibility to the “smudgy” perturbations generated by the *Median Cut Filtering* StirMark test.

The restriction to “minutiae-based fingerprint matchers” in previous statement was necessary, as for fingerprint images of DB3, the **FingerCode** matcher displays an extensive impairment of its matching results. So while being the matcher the most robust to median cut filtering for fingerprint images of DB1 and still being the second most robust matcher in case of DB2, now here, for images of DB3, the FingerCode matcher exhibits clearly the worst performance. Although the results produced by images filtered with the 3×3 filter are as good as those for unperturbed images, as can be seen in Figure 5.14c, for larger filter-sizes, the relative distance between the corresponding ROC curves increases strongly. Additionally, the results of medianCut 09 show such high rates of false non-matches, that in this case an EER value can not be reached.

The **Phase Only Correlation** matcher shows a somewhat “mixed” reaction to the perturbations introduced in the fingerprint images of DB3: The application of filters of sizes 3×3 and 5×5 cause a stronger impairment of the matching results of the POC matcher, than it does in the results of the minutiae-based matchers. Then again, the contrary is true, when regarding the results caused by filters of sizes 7×7 and 9×9 : Here the POC matcher clearly turns out to be the matcher the most robust to the perturbations. An observation also backed up by the corresponding equal error rates (please refer to Table 5.8 for details).

Filter Size	bozo3 (%)	VF (%)	GF180 (%)	GF20 (%)	FC (%)	POC (%)
unperturbed	6.68	3.60	6.90	5.82	8.98	15.07
03	7.28	3.49	6.59	5.46	9.12	15.98
05	8.26	7.18	8.40	7.32	12.51	19.43
07	18.70	22.31	17.85	15.42	26.66	24.85
09	35.09	48.88	30.29	26.92	–	29.30

Table 5.8: Equal error rates for *Median Cut Filtering* test conducted on sample image database DB3.

5.5 Convolution Filtering – Mean Filtering (convMean)

Relation to Fingerprints

By itself the StirMark Benchmark only provides a generic *Convolution Filtering* operation, that filters the input image with an arbitrary user-defined filter matrix. This of course provides the possibility for a wide range of image manipulations. Looking at the most common utilizations of convolution filters, image sharpening, edge-detection and embossing did not appear to me, to have any real application in the attempt to introduce natural perturbations into fingerprint images. Blur on the other hand is an impairment that can very well be observed in original, real-life fingerprint images. The most common and renowned cause for this blur probably being slight movement of the finger during image acquisition.

Now to simulate this blurring effect, I chose to apply the most simple form of image blurring by convolution operation: *mean filtering*. The corresponding filter matrix can be described as follows: Let h be a mean filter of size $n \times n$, then it is given by:

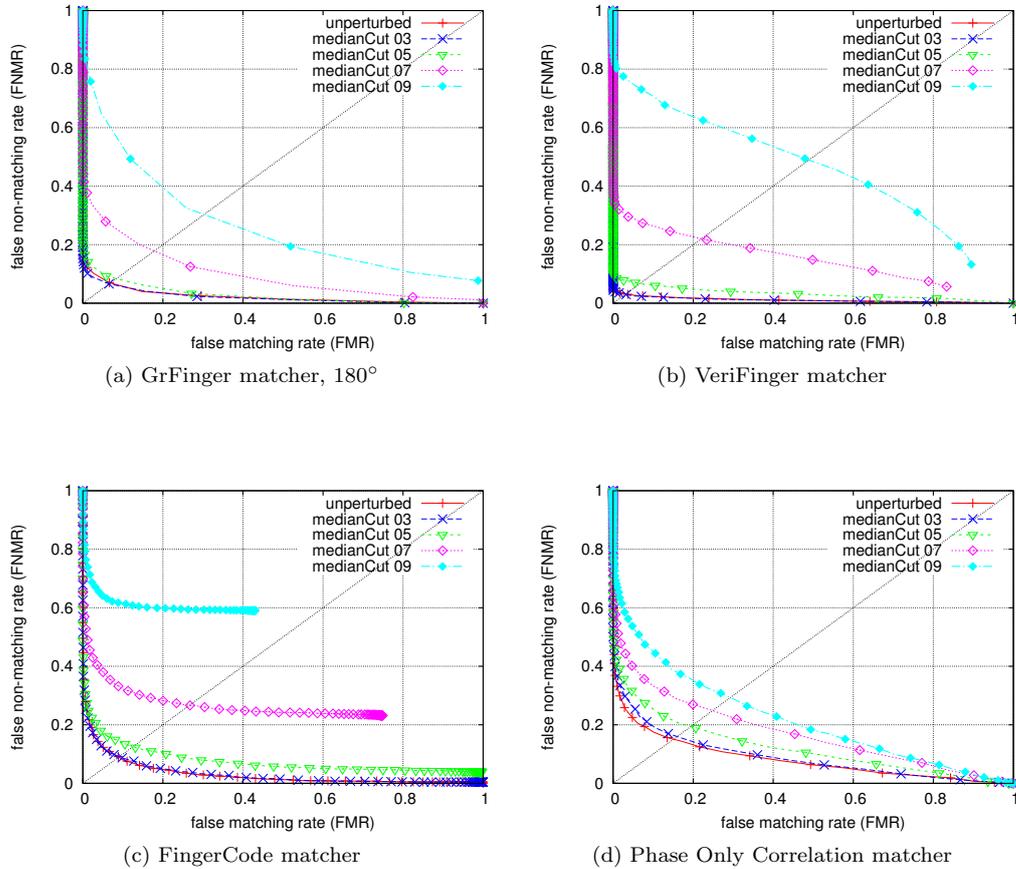


Figure 5.14: Receiver operating characteristics for StirMark test *Median Cut Filtering* on fingerprint images of DB3. “median Cut 03 .. 09” refers to the size of the median cut filter.

$$h_{n,n} = \frac{1}{n^2} \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{pmatrix}$$

Figure 5.15: $n \times n$ mean filter.

Parameter Configurations

In the experiments of present work, I use mean filters of sizes $\{3, 5, 7\}$. Larger filter matrices are not applied, as for once the blurring of the original ridge-and-furrows structure gets too strong to still be considered “natural” and second, like with median cut filtering (5.4), such excessive filtering also tends to create spurious ridges in the fingerprint. Examples of filtered images can be seen in Figure 5.16.



Figure 5.16: Examples for the *Mean Filtering* test, applied to an image from DB1 (ID 91.2).

Results and Discussion

DB1

Among the minutiae-based fingerprint matchers, **bozorth3** is clearly the one, that is the least influenced by the blur, introduced in the fingerprint images of database DB1. Inspecting the results more closely, we find, that for the 3×3 filter, the matching performance is equal and in parts even slightly better than that for unperturbed images. As can likewise be seen in Figure 5.17a, also the ROC curve caused by employing the 5×5 filter, lies comparatively close to that for unperturbed images. A significant drop in matching performance then only occurs, when filtering the images with the filter of size 7×7 . The difference between the EER for unperturbed images and the EER caused by the 7×7 filter is then 17.10%, as compared to 1.86% for the case of filter-size 5×5 .

The next best matching performance of a minutiae-based matcher is achieved by **GF20**. While the ROC plots of both GrFinger configurations look very much alike, the relative distances between the individual ROC curves are slightly smaller for GF20 than for **GF180** – a property, that can also be witnessed, when regarding the respective equal error rates in Table 5.9. Inspecting said ROC plots, we can see, that mean-filtering with a filter of size 3×3 has only a very small negative influence on the matching results. The more obvious difference to the performance of bozorth3 becomes apparent, when regarding the ROC curve produced by the 5×5 filter: While, as stated above, for bozorth3 it lies still very close to the curve for unperturbed images, here, for GrFinger it is already clearly separated. The differences between the corresponding EER values and that for unperturbed fingerprint images then results to 8.40% for GF180 and 5.26% for GF20. When finally regarding the response of GrFinger to fingerprint images filtered with the 7×7 mean-filter, we find, that the results of GF20 are slightly less and the results of GF180 are slightly more impaired than those of bozorth3. On the other hand we can also see in the related ROC curves of (for example) GF20, that with the first step of the threshold the FMR and FNMR jump to 70% and 13% respectively, while bozorth3 “delivers” distinct values until a point of 94% FMR and 1% FNMR.

When regarding the effect of blurring the fingerprint images of DB1 on the performance of the **VeriFinger** matcher, we can largely compare the results for mean-filter-sizes 3×3 and 5×5 to those obtained with GF180. Only in the area around the EER VeriFinger

achieves slightly better results in the 3×3 filter case, than GF180. However, when dealing with images perturbed by the 7×7 mean-filter, VeriFinger exhibits a strongly impaired matching performance: For example the distance of the corresponding EER to the EER for unperturbed images amounts to 39%, while the second worst distance, produced by GF180 in the 7×7 case, is only 23%. Further, when regarding the ROC curve, we note, that it forms a convex shape and ends already at a FMR of 77% and a FNMR of 15%.

Both **non-minutiae-based fingerprint matchers** are clearly less influenced by the blurring introduced by mean-filtering in the fingerprint images of DB1, than the minutiae-based matcher.

Of all tested fingerprint matchers, **FingerCode** turns out to be the most robust to the perturbations in DB1 fingerprints. Inspecting the ROC plot in Figure 5.17d, we see, that the curves relating to the filters of sizes 3×3 and 5×5 lie almost on top of the ROC curve for unperturbed images and also the distance to the ROC curve caused by the 7×7 filter is less, than in any other matcher’s results - the difference between the EER for unperturbed images and that caused by the 7×7 filter is only 3.90%.

The **Phase Only Correlation** matcher shows the second best matching performance (see also Figure 5.17d): Very much like in case of bozorth3, the ROC curve for the results caused by the 3×3 filter lies on top and in parts even a little under the ROC curve for unperturbed images, and the curve related to the 5×5 filter is only slightly separated from it. The filter of size 7×7 then influences the matching performance already stronger, yet still the relative impairment is weaker than in any minutiae-based matcher (compare also the respective equal error rates in Table 5.9).

Filter size	bozo3 (%)	VF (%)	GF180 (%)	GF20 (%)	FC (%)	POC (%)
unperturbed	14.81	5.87	11.41	13.61	12.54	22.60
3	14.41	6.89	13.33	14.25	12.75	22.70
5	16.67	15.03	19.80	18.86	13.07	24.58
7	31.91	44.64	33.98	29.15	16.43	31.13

Table 5.9: Equal error rates for *Convolution Filtering – Mean Filtering* test conducted on sample image database DB1.

DB2

Interestingly, for fingerprint images of DB2, the reaction of the **minutiae-based fingerprint matchers** to the blurriness introduced by mean-filtering, is largely the same: Apart from the specific values, we can observe, that a filter-size of 3×3 only has a very limited effect on the matching performance. Correspondingly we can see in the individual ROC plots, that the ROC curve caused by the 3×3 filter lies only slightly above that for unperturbed images. For a mean filter of size 5×5 then, the matching performance is already stronger impaired – referring to the respective equal error rates, (as listed in Table 5.10), we see, that the difference between the ERR of unperturbed images and the EER caused by the 5×5 filter is 6.67% for GF20 and around 8% for the other minutiae-based matchers. A large drop in matching performance then occurs with the application of the 7×7 mean filter. Once again referring to the equal error rates, we can find, that this impairment is about 3.5 (bozorth3) to 5 times (GF20) as strong as that introduced by the 5×5 filter.

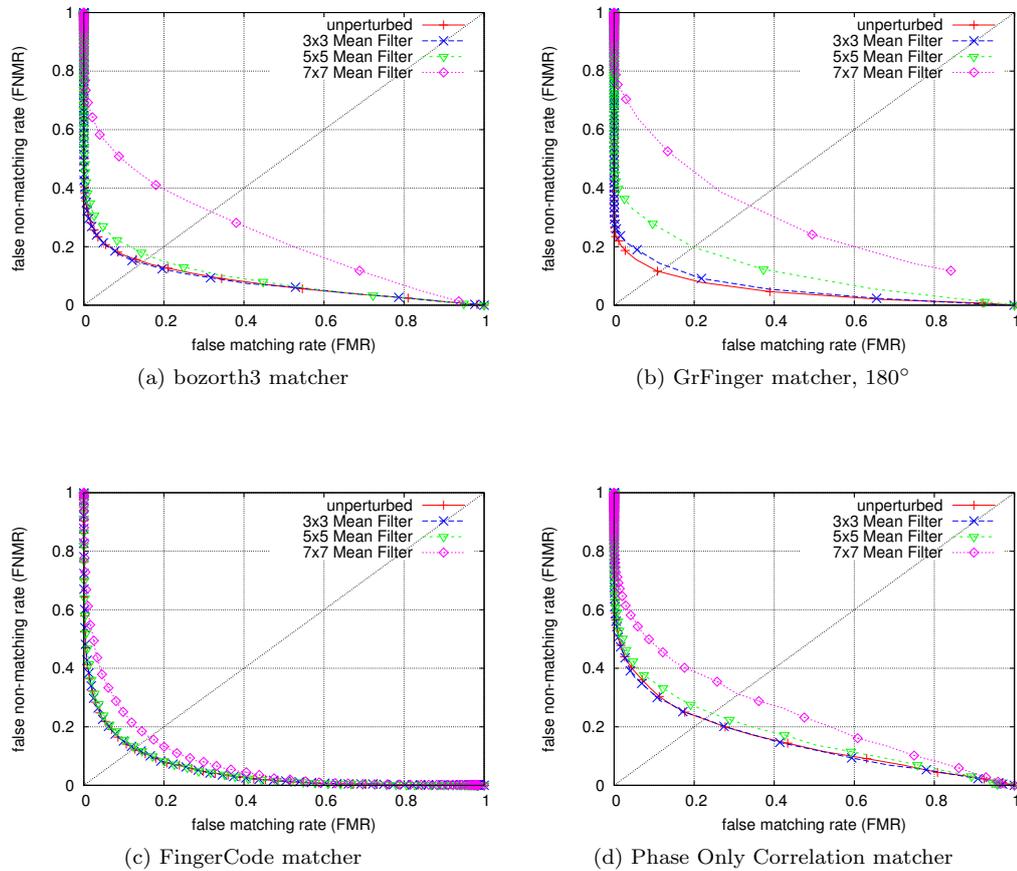


Figure 5.17: Receiver operating characteristics for StirMark test *Convolution Filtering – Mean Filtering* on fingerprint images of DB1.

While showing clearly the highest robustness to blurriness in fingerprint images of DB1, for images of DB2 the **FingerCode**'s matching results are decidedly the most impaired ones: As is the case with the other matchers, filtering the images with a 3×3 mean-filter only barely affects the matching performance of FingerCode. Yet already when regarding the influence of the 5×5 filter, we find, that the results are worse than for any other of the tested matchers, at this level of perturbations. And the same is true for fingerprint images filtered with the 7×7 mean-filter: When regarding Figure 5.18c, we not only see, that the ROC curve related to the 7×7 filter lies above the corresponding curves in the other matchers' plots, but we also notice, that already with the first incrementation of the threshold, we obtain a FMR of about 29% and a FNMR of 66% – an equal error rate is not even reached anymore.

Like was also the case in the *Median Cut Filtering* test for fingerprint images of DB2, the **Phase Only Correlation** matcher now is the matcher, that, in general, is the least influenced by the introduced perturbations. While according to the respective EER values, the POC matcher is still slightly beaten in the performance related to the 3×3 filter by bozorth3, GF20 and even FingerCode, for the two larger filter sizes its performance is undisputedly superior: Already when inspecting the ROC plot in Figure 5.18d, we see, that the relative distances among the individual ROC curves of the POC matcher are notably

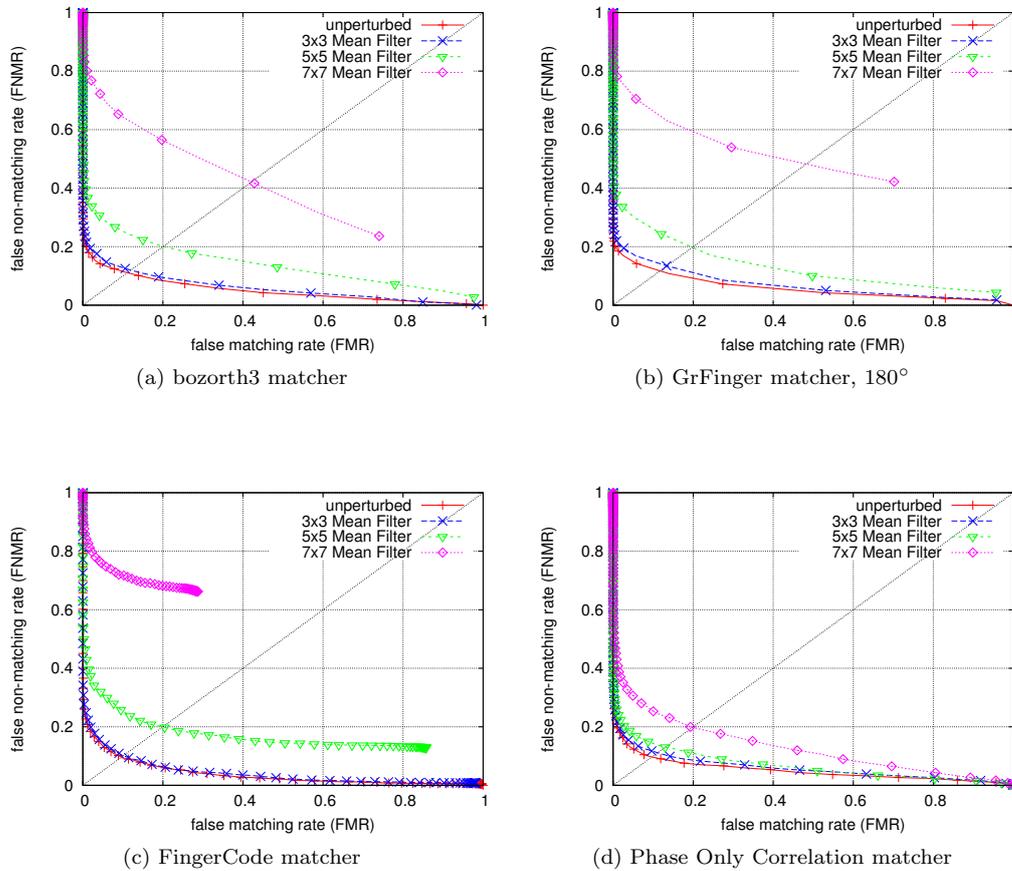


Figure 5.18: Receiver operating characteristics for StirMark test *Convolution Filtering – Mean Filtering* on fingerprint images of DB2.

smaller than is the case for any other matcher and the equal error rates listed in Table 5.10 back-up this observation.

Filter size	bozo3 (%)	VF (%)	GF180 (%)	GF20 (%)	FC (%)	POC (%)
unperturbed	11.12	5.01	11.72	12.89	9.60	9.69
3	11.99	6.50	13.44	14.16	10.08	11.14
5	20.06	13.39	19.78	19.57	19.95	13.03
7	42.10	44.22	48.23	46.62	–	19.80

Table 5.10: Equal error rates for *Convolution Filtering – Mean Filtering* test conducted on sample image database DB2.

DB3

Like previously witnessed for fingerprint images of DB2, also in the present test the individual reactions of the **minutiae-based** fingerprint matchers to the introduced blur, are very much alike. In fact, the only major difference, when comparing the matching performances for images of DB2 and images of DB3, is, that for fingerprint images of DB3 the relative impairment of the matching results, caused by the mean filters of sizes 5×5 and 7×7 , is clearly less extensive: Regarding the ROC plots for DB3 images, the curves relating to filter-size 5×5 are now very close to those for unperturbed images and only the ROC curves caused by the 7×7 filter lie clearly separated from the others. This aspect can also be observed, when comparing the respective equal error rates.

So while for the minutiae-based matchers the effect of the blur-simulating perturbations is less evident when dealing with images of DB3, the opposite is true for the **FingerCode** matcher: Although FingerCode’s performance was already very bad in the tests for fingerprint images of DB2, in present experiments for DB3 images, the results turn out to be even more impaired by the mean filtering. When regarding Figure 5.19c and comparing it to Figure 5.18c of the DB2 case, we can see, that the relative distance between the ROC curves produced by the mean-filtered images and that for unperturbed images increased in present tests. In particular we also see the extremely distorted ROC curve portraying the results for fingerprint images of DB3 filtered with the 7×7 mean-filter. The reason for its shape is, that the FingerCode matcher seemingly has such big difficulties, matching this filtered fingerprint images, that 89% of all match-pairs result to a matching score of 0. In consequence, with the first increment of the threshold, we obtain a false match rate of 12% but at a false non-match rate of 91%.

The **Phase Only Correlation** matcher performance in the *Convolution Filtering – Mean Filtering* tests for DB3 is very much comparable to its performance in tests for DB2. Comparing the results in detail, we see two relatively small differences: The impairment of the matching performance caused by filtering with the mean-filter of size 5×5 is slightly stronger for DB3 images – also the difference between the EER for unperturbed images and that for images filtered with the 5×5 filter increased by 1.22%. On the other hand for images of DB3 filtered with the 7×7 , the negative influences are less apparent than in the DB2 results – here above-described difference in the EER values is 3.31% smaller.

Overall, for fingerprint images of DB3, we can no longer easily determine, which fingerprint matcher is the most robust to the added blur. While POC still proves to be the least influenced by filtering the images with a mean-filter of size 7×7 , the minutiae-based matchers now display a clearly superior performance for filter-sizes 3×3 and 5×5 .

Filter size	bozo3 (%)	VF (%)	GF180 (%)	GF20 (%)	FC (%)	POC (%)
unperturbed	6.68	3.60	6.90	5.82	8.98	15.07
3	6.72	3.52	6.73	5.45	10.09	16.00
5	6.76	6.11	7.83	6.44	22.71	19.63
7	17.17	20.59	17.13	14.33	–	21.88

Table 5.11: Equal error rates for *Convolution Filtering – Mean Filtering* test conducted on sample image database DB3.

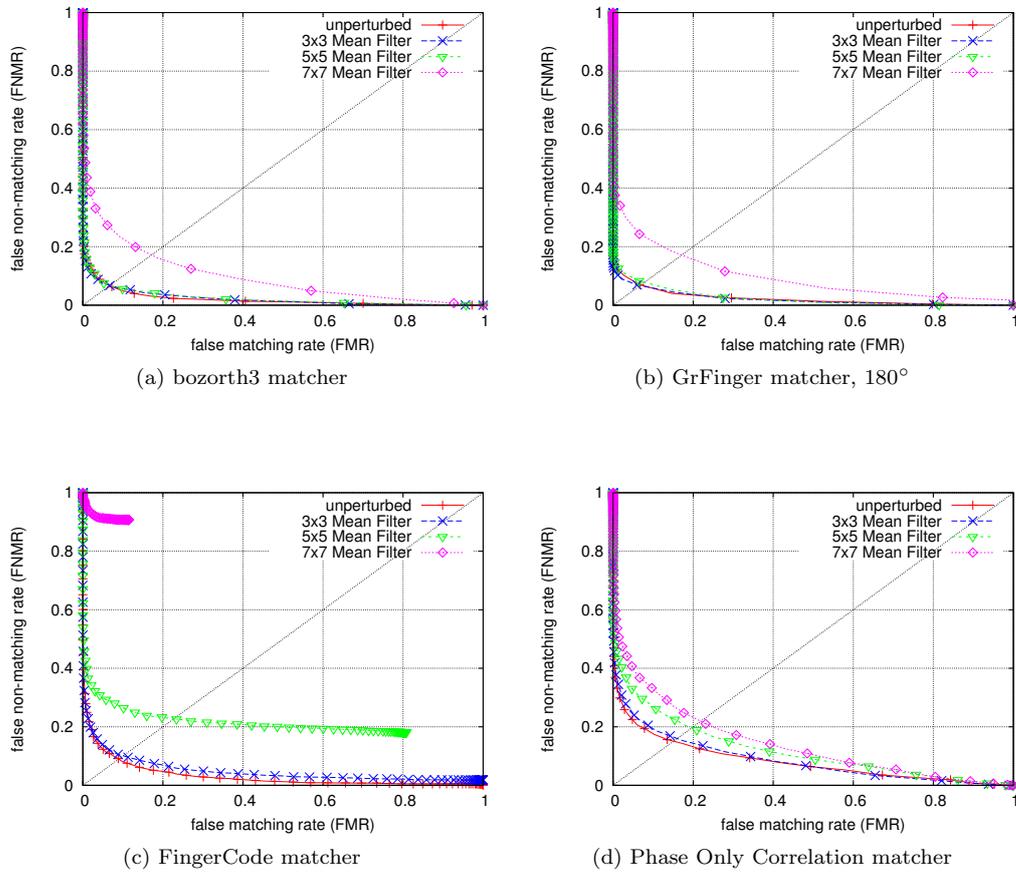


Figure 5.19: Receiver operating characteristics for StirMark test *Convolution Filtering – Mean Filtering* on fingerprint images of DB3.

5.6 Convolution Filtering – Modified Gaussian Filtering (convGauss)

Next to mean filters, a second commonly used convolution filter type to introduce blur to an image, are Gaussian filters. As starting ground for present StirMark test, I chose to employ the most simple version of a Gaussian filter, in form of the 3×3 matrix seen in Figure 5.20.

$$h_{3,3} = \frac{1}{16} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$$

Figure 5.20: Original 3×3 Gaussian filter.

Relation to Fingerprints

First trials with the filter matrix of Figure 5.20 produced blurry fingerprint images that looked very much like the output created by the *Mean Filtering* test. The differences were not distinctive enough to justify an additional, separate test. However, further experiments with slightly changed matrix values or with an altered normalization constant lead to following interesting results:

While blur is no longer introduced in the images, the finger imprints themselves tend to “thin out” – gradually details are being lost, the imprint’s gray-scale values get slightly brightened and the actual visible section of the imprint is reduced. Thus the images produced by this *Modified Gaussian Filtering* test very much resemble those original, real-life fingerprint images, that result, when during acquisition the fingertip was either too dry or the exercised pressure on the contact area too low. Examples can be seen in Figure 5.22.

Interestingly the described effect of this StirMark test is only observable in fingerprint images of databases DB1 and DB2. For images of DB3, the convolution with the modified Gaussian filter matrices has two types of impact: for one, it roughens the ridge lines, amplifying imperfections already present in the input image; and second, it tends to “clear” the overall image – the darkish background is brightened up and gets almost separated from the actual imprint. Corresponding examples can be seen in Figure 5.27.

Parameter Configurations

As also noted above, in the experiments of present work, we will be using eight different modifications of the commonly used 3×3 Gaussian filter matrix stated in 5.21. There configurations are as follows:

$$\begin{array}{ccc}
 h_{3,3} = \frac{1}{12} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} & h_{3,3} = \frac{1}{11} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} & h_{3,3} = \frac{1}{9} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} \\
 \text{(a) Configuration 1} & \text{(b) Configuration 2} & \text{(c) Configuration 3} \\
 \\
 h_{3,3} = \frac{1}{9} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 6 & 2 \\ 1 & 2 & 1 \end{pmatrix} & h_{3,3} = \frac{1}{9} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 8 & 2 \\ 1 & 2 & 1 \end{pmatrix} & h_{3,3} = \frac{1}{9} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 12 & 2 \\ 1 & 2 & 1 \end{pmatrix} \\
 \text{(d) Configuration 4} & \text{(e) Configuration 5} & \text{(f) Configuration 6} \\
 \\
 h_{3,3} = \frac{1}{9} \begin{pmatrix} 1 & 2 & 1 \\ 3 & 12 & 3 \\ 1 & 2 & 1 \end{pmatrix} & h_{3,3} = \frac{1}{9} \begin{pmatrix} 1 & 3 & 1 \\ 3 & 12 & 3 \\ 1 & 3 & 1 \end{pmatrix} \\
 \text{(g) Configuration 7} & \text{(h) Configuration 8}
 \end{array}$$

Figure 5.21: Eight modifications of a Gaussian filter matrix, that will be applied in the *Modified Gaussian Filtering* tests during the experiments of present work.

The reduced set of test, that will also be conducted on the non-minutiae-based fingerprint

matchers, covers the configurations $\{1, 2, 3, 4, 5, 6\}$ for DB1 and DB2, and configurations $\{1, 3, 6, 8\}$ for DB3. Example images for configurations 1, 4, and 8 being applied on a fingerprint image of database DB1, can be found in Figure 5.22. For comparison, the different reaction of images from DB3 to the *Modified Gaussian Filtering* test can be seen in Figure 5.23.



Figure 5.22: Examples for the *Modified Gaussian Filtering* test, applied to an image from DB1 (ID 91.2).

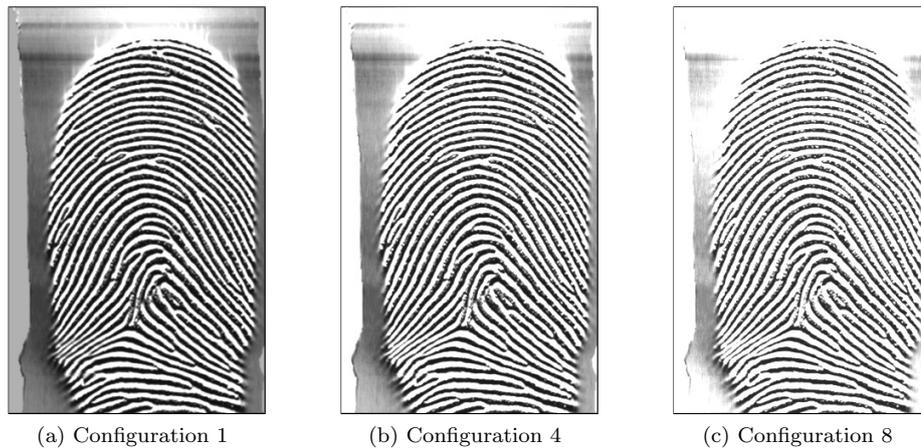


Figure 5.23: Examples for the *Modified Gaussian Filtering* test, applied to an image from DB3 (ID 49.4).

Results and Discussion

DB1

When comparing the matching performances of the fingerprint matchers in regard to their responses to the perturbed fingerprint images, we can primarily make following general observations:

The matching performance of every single matcher clearly worsens with each step of increased effectiveness of the modified Gaussian filter (identifiable by the configuration-numbers: 1

has weakest and 8 has strongest effect). The only two exceptions can be seen in GF180 and VeriFinger, where configuration 7 affects the results stronger than configuration 8.

Furthermore we notice when regarding the ROC plots (see also the selection of plots in Figure 5.24), that if, within a plot, we were to build clusters of ROC curves based on their relative distances, for all matchers we would get the same following groups: configurations 1 and 2, then 3, 4 and 5 and finally 6, 7 and 8. So the responses of the matchers to the various filter-configurations appears to be rather analogue – what mostly differs then from matcher to matcher (read: from ROC plot to ROC plot) are the intra- and inter-group distances in the results.

As third general observation we can state for the reduced set of configurations 1 to 6, that the **non-minutiae-based matchers** are less influenced by the perturbations introduced in the fingerprint images, than the **minutiae-based** matchers. Especially the **Phase Only Correlation** matcher appears to be fairly robust to the image manipulations. This can not only be seen in the respective ROC plots, but also the list of equal error rates in Table 5.12 confirms the observation – for instance, if we regard the difference between the EER for unperturbed images and that related to the strongest perturbations (here configuration 6), it is 9.70% for POC while 16.30% for **FingerCode** and even larger for the other minutiae-based matchers.

Turning to the set of minutiae-based matchers, we find, that the matcher, whose matching performance is the least influence by the effects of filtering the fingerprint images of DB1, is **GF20**, closely followed by the **bozorth3** matcher.

As for the two remaining matchers, **GF180** and **VeriFinger**, VeriFinger performs better for images filtered with the weaker filter-configurations 1 and 2, yet for stronger perturbation-levels then, GF180 clearly produces less-impaired results. This is especially true for filter-configurations 6, 7 and 8, where the results of VeriFinger are so bad, that an equal error rate can not be established anymore (see also Figure 5.24b)

Configuration	bozo3 (%)	VF (%)	GF180 (%)	GF20 (%)	FC (%)	POC (%)
unperturbed	14.81	5.87	11.41	13.61	12.54	22.60
1	20.91	11.54	17.63	17.25	14.42	25.27
2	22.73	14.16	20.57	20.87	15.36	25.31
3	26.45	21.56	26.25	24.23	19.06	28.68
4	28.27	24.42	29.09	25.71	20.93	29.49
5	30.65	27.18	30.87	27.54	23.04	30.15
6	32.37	–	34.78	30.65	28.84	32.29
7	34.11	–	37.28	32.14		
8	34.61	–	35.97	32.32		

Table 5.12: Equal error rates for *Convolution Filtering – Modified Gaussian Filtering* test conducted on sample image database DB1.

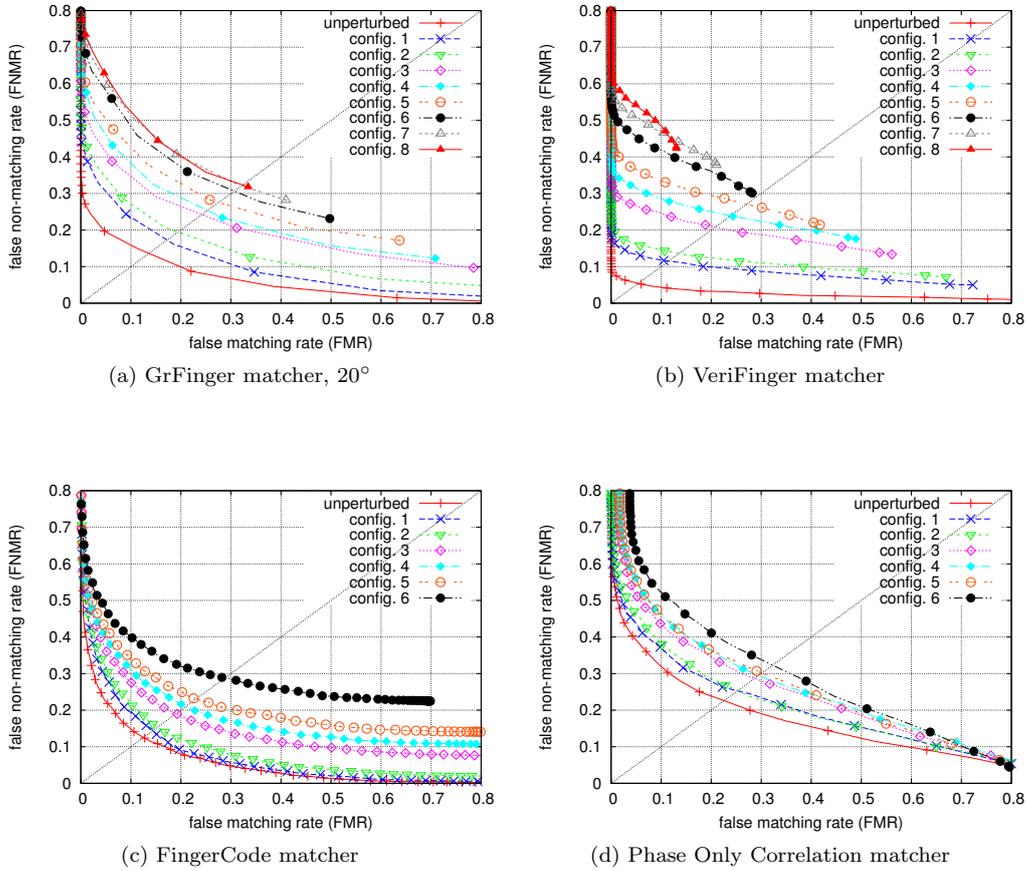


Figure 5.24: Receiver operating characteristics for StirMark test *Convolution Filtering – Modified Gaussian Filtering* on fingerprint images of DB1.

DB2

Regarding the matching results of the tested fingerprint matchers for perturbed fingerprint images of database DB2, we can see, that the impact of filtering the images with above-defined modified Gaussian filters, is clearly stronger, than it was for images of DB1. While basically being able to make the same general observations, as we did for fingerprint images of DB1, about the gradual worsening of the matching performance with increasing effectiveness of the modified Gaussian filters, as well as about the relative spacing of the ROC curves in the related plots, we have to state, that for filter-configurations 4 and above, the perturbations in the fingerprint images of DB2 lead to severely degraded matching results:

At configuration 4, it is only the matchers GF180, FingerCode and POC, that still deliver results, that enable us to determine their respective equal error rates. For the other matchers, the results are already so bad, that with the first incrementation of the threshold, the false non-match rates rise to values larger than the related false match rates.

From configuration 5 onwards, POC is the only matcher left, whose results still cross the equal error rate. Overall, when looking at the selected ROC plots in Figure 5.25 – for

“completeness”: the ROC plots for bozorth3 and GF20, which were left out to save space, look very similar to the plot for GF180 – we can find, that the POC matcher is quite exceptional in relation to the broad spectrum of its results. Even for the highest filter configuration, that POC was tested on – configuration 6 – the results still cover the ranges [0%, 63%] for FMR and [26%, 100%] for FNMR, while the second most detailed results for this perturbation level, by FingerCode, only cover the ranges [0%, 17%] for FMR and [64%, 100%] for FNMR.

It has to be noted though, that for the filter configurations 1 and 2, FingerCode and further also GF20 show to be more robust to the perturbations introduced in the images, than POC. An observation that can also be verified, when regarding the respective equal error rates in Table 5.13.

The two matchers that are the most sensitive to the reduction of the finger-imprint area, caused by filtering the images of DB2 with the modified Gaussian filters, are bozorth3 and VeriFinger. While for the weak filter configurations 1 and 2, VeriFinger still outperforms bozorth3, for configurations 3 to 8, it very evidently exhibits the worst matching performance of all fingerprint matchers tested (see also Figure 5.25). Inspecting for instance the results for the most “extreme” case – filter configuration 8 – VeriFinger assigns a matching score of 0 to 93% of all fingerprint pairings matched, while bozorth3 to 86% and GF180 to “only” 81% of all pairings.

Configuration	bozo3 (%)	VF (%)	GF180 (%)	GF20 (%)	FC (%)	POC (%)
unperturbed	11.12	5.01	11.72	12.89	9.60	9.69
1	17.93	10.96	16.73	16.81	12.80	15.14
2	25.49	17.93	22.48	21.83	17.00	19.14
3	37.67	–	38.18	35.74	31.53	32.09
4	–	–	42.57	–	39.42	37.46
5	–	–	–	–	–	39.13
6	–	–	–	–	–	42.60
7	–	–	–	–	–	–
8	–	–	–	–	–	–

Table 5.13: Equal error rates for *Convolution Filtering – Modified Gaussian Filtering* test conducted on sample image database DB2.

DB3

As mentioned before, the application of the modified Gaussian filters to images of database DB3 generally leads to a different kind of perturbations than it does for fingerprint images of databases DB1 and DB2: Instead of gradually reducing the section of the fingerprint, that is visible in the image, filtering an fingerprint image from DB3 tends to clear this image. Additionally the surface of the ridge-lines get roughened (please refer to section “Relation to Fingerprints” on page 113 for details).

Due to the fact, that the described types of perturbations do not essentially influence the ridge-and-furrows structure of a finger imprint (in best case, the “clearing” effect of the

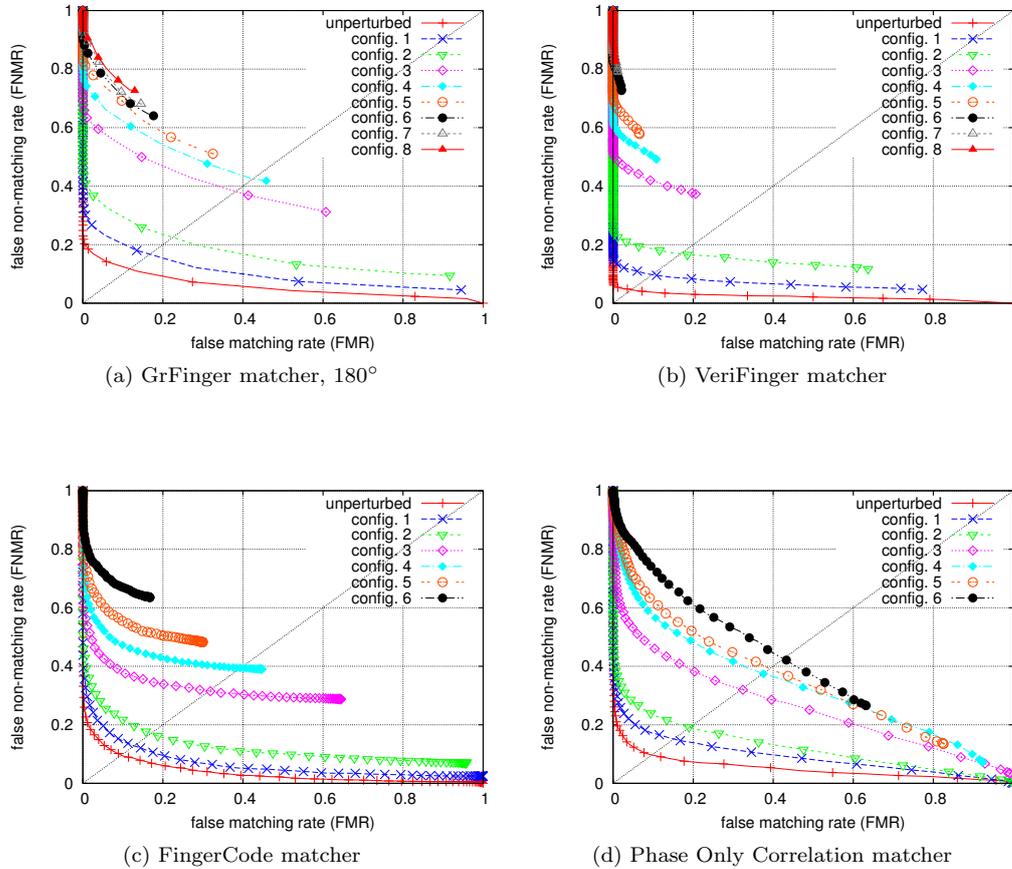


Figure 5.25: Receiver operating characteristics for StirMark test *Convolution Filtering – Modified Gaussian Filtering* on fingerprint images of DB2.

filtering might even lead to a improved perceptibility), no major influence of the modified Gaussian filtering test on the performance of the individual fingerprint matchers is to be expected.

The experiments confirm this assumption. In fact, the only matcher, where a rather distinct impairment of the matching results occurs, is the **Phase Only Correlation** matcher. The POC matchers' special sensitivity to the filtering process might be contributed to said roughening of the ridges surface – as it adds a large amount of medium to high frequency noise to the image, it is very possible, that this affects the exactness of the highly frequency-dependent Phase Only Correlation.

Inspecting the results for the POC matcher more closely, we find, that the matching performance worsens with increasing effectiveness of the modified Gaussian filters. The only exception is around the area of the EER, where locally configuration 8 produces slightly better results than configuration 6. The corresponding equal error rates (listed in Table 5.14) show, that the maximum distance between the EER for unperturbed images and the EER values for the filtered images is 4.64% – which actually is still a comparatively small value.

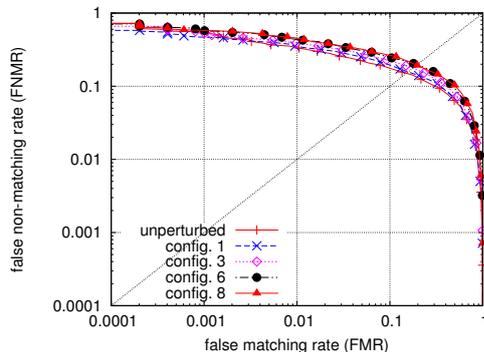


Figure 5.26: Receiver operating characteristics of the Phase Only Correlation matcher for StirMark test *Convolution Filtering – Modified Gaussian Filtering* on fingerprint images of DB3.

When looking at the ROC plots for the **other fingerprint matchers** considered in the tests, we find, that the ROC curves produced by the fingerprint images filtered with the filters of given configurations, all lie almost on top of the respective ROC curves for unperturbed images. In other words, the effect of filtering DB3-images with the modified Gaussian filters, on the matching performance of the fingerprint matchers, is marginally at best. It is only by looking very closely at the ROC plots and by regarding the EER values, that we find minor differences in the matching results, produced by the **individual matchers**:

Probably the most noteworthy fact is, that for matchers GrFinger, VeriFinger and FingerCode, the filtered fingerprint images generally lead to better results than the unperturbed ones (only exceptions: GF180 and VeriFinger for those filter configurations, that create the strongest perturbations – configurations 7 and 8). In the respective EER values we can see relative improvements of 0.22% to 1.07%. bozorth3 appears to be the only matcher (despite POC of course), where filtering the fingerprint images of DB3 has a negative effect on the matching performance – once again referring to the equal error rates, we find a maximum difference of 1.02% between the EER for unperturbed images and those values caused by image filtering.

For illustration, in Figures 5.27a and 5.27b ROC plots for matchers bozorth3 and GF20 respectively, can be found (the plots apply a logarithmic scale with ranges set to FMR [1%, 60%] and FNMR [1%, 15%]).

5.7 Remove Lines (rml)

This StirMark test removes single rows and columns from the input image, in adjustable, quasi-periodical intervals.

Relation to Fingerprints

This test aims to simulate errors in fingerprint images, that occasionally occur during fingerprint acquisition, when the scanner is not able to read the fingerprint in its entirety, but misses/skips certain lines. Especially sweep sensors are prone to this kind of complications. Two corresponding examples can be found in Figure 5.28.

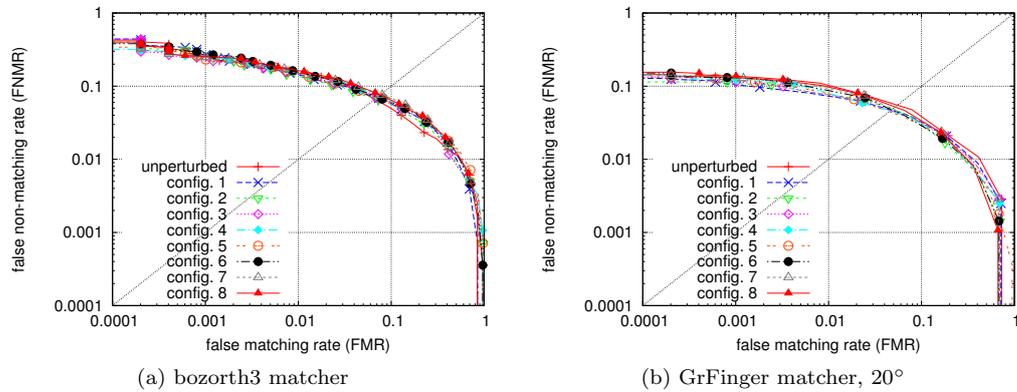


Figure 5.27: Receiver operating characteristics for StirMark test *Convolution Filtering - Modified Gaussian Filtering* on fingerprint images of DB3. (Plots apply logarithmic scales. Ranges are set to FMR [1%, 60%] and FNMR [1%, 15%].

Configuration	bozo3 (%)	VF (%)	GF180 (%)	GF20 (%)	FC (%)	POC (%)
unperturbed	6.68	3.60	6.90	5.82	8.98	15.07
1	6.99	3.44	5.83	5.02	8.75	15.73
2	6.79	3.40	6.01	5.15		
3	6.92	3.34	6.09	4.97	8.46	17.70
4	7.48	3.35	6.06	4.82		
5	7.22	3.13	6.18	4.78		
6	7.12	3.35	6.08	5.08	8.64	19.71
7	7.71	3.77	6.66	5.27		
8	7.62	4.04	7.07	5.48	8.76	19.43

Table 5.14: Equal error rates for *Convolution Filtering - Modified Gaussian Filtering* test conducted on sample image database DB3.

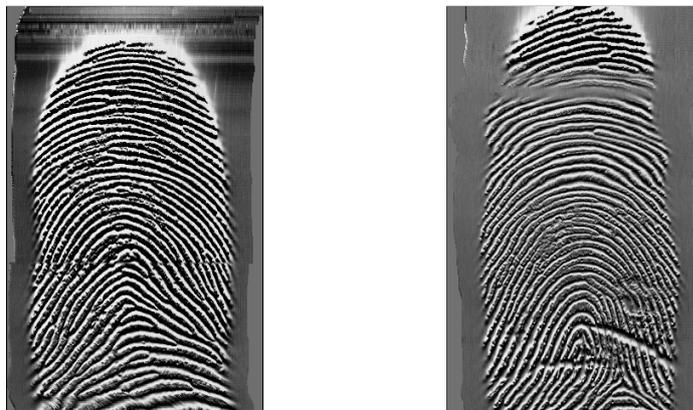


Figure 5.28: Two examples showing the type of error introduced in the finger imprint, when the fingerprint scanner misses certain lines during acquisition.

Parameter Configurations

The frequency of line removal can be adjusted by a single parameter k , following the principle: “remove 1 line in every k lines”. For added randomization, per single line removal the system also shifts the value of k by ± 1 on a random basis. The possible range of values for k is $[1, \min(\text{image_width}, \text{image_height})]$. One thing, that has to be kept in mind though, is, that the line removal operation naturally also reduces the size of the output image.

In the experiments of present work, the frequency parameter will be set to the following values: $\{100, 90, 80, 70, 60, 50, 40, 30, 20, 10\}$. The limited set of test runs, that will be run on the non-minutiae-based matchers, will cover the parameter values $\{90, 70, 40, 20\}$. Examples for the *Remove Lines* test being applied to an image of DB1 for the parameter values 100, 50 and 10 can be seen in Figure 5.29.

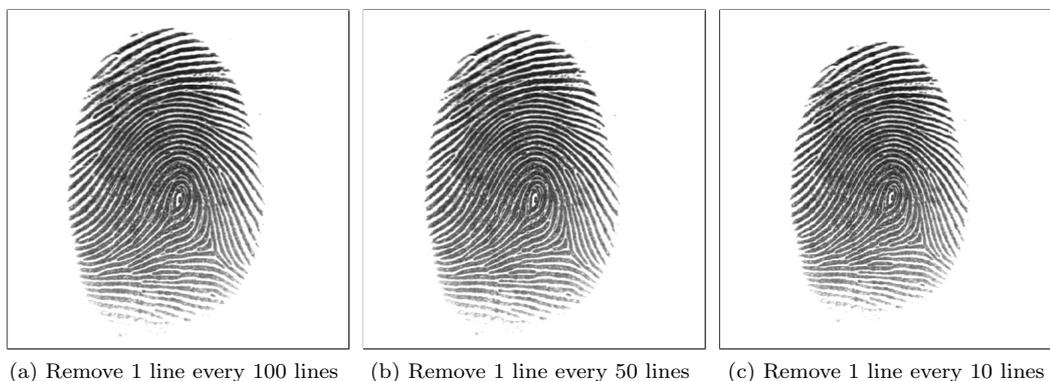


Figure 5.29: Examples for the *Remove Lines* test, applied to an image from DB1 (ID 91-2).

Results and Discussion

First of all we can state, that the influence of the *Remove Lines* perturbations on the matching performance of the fingerprint matchers tested, is very limited. For example, inspecting the equal error rates of the matching results for perturbed images of all three databases (see Tables 5.15, 5.16, 5.17) we find, that 80% of the 144 equal error rates relating to perturbed images lie within a distance of $\pm 1\%$ of the respective EER for unperturbed images and 91% lie within $\pm 2\%$. In fact, there are only two results, where the difference between their EER and the EER for unperturbed images is larger than 4% and both are produced by POC at the most effective *Remove Lines* configuration tested with this matcher – “1 in 20”: a difference of 5.06% for images of DB2 and a difference of 8.53% for images of DB3.

Furthermore, due to the general low influence of the *Remove Lines* test on the individual matching results, we find, that the reaction of the fingerprint matchers to the perturbations in the fingerprint images is basically the same for all three image databases, DB1 to DB3. For illustration, Figure 5.30 shows examples for images of DB1.

First, regarding **minutiae-based fingerprint matchers**: for the effectiveness levels “1 in 100” to “1 in 40”, the results are very close to those for unperturbed images – sometimes slightly worse, sometimes slightly better (compare also with listing of EER values in Tables 5.15, 5.16, 5.17). As can be seen in Figures 5.30a and 5.30a, the corresponding ROC curves

lie in a comparatively close surrounding of the ROC curve representing the matching performance for unperturbed images. From level "1 in 30" on then, the matching results exhibit a permanent decrease in matching performance, with the biggest "drop" evident in the results for the most effective perturbation level of "1 in 10".

As previously stated, the **non-minutiae-based** fingerprint matchers were only tested on a limited set of perturbation-levels. Yet the inspection of the results and the corresponding EER values, hints to following tendencies:

The **FingerCode** matcher seems to be comparatively robust to the perturbations introduced. Especially for images of DB1, the equal error rates for perturbed images never differ more from that for unperturbed images than $\pm 0.1\%$ (for DB2 and DB3 the difference values are $\pm 3.8\%$ and $\pm 3.0\%$ respectively). An observation that can for instance also be seen in Figure 5.30c.

The **Phase Only Correlation** matcher, on the contrary, is the matcher, whose matching performance clearly displays the strongest influence by the image perturbations. Especially for the two more efficient perturbation-levels tested – "1 in 40" and "1 in 20" – the results of POC are notably stronger impaired than those of any other matcher regarded. An effect that is also visible in the corresponding ROC plots (see Figure 5.30d for example). The current estimation, as to why the POC matcher reacts decidedly more sensitive to the image manipulations caused by the *Remove Lines* test, than the competitors, is as follows: The down-scaling of the finger imprint, that is inevitably caused by the line-removal operation of the test, changes the intrinsic frequency of its ridge-and-furrows structure, which on the other hand implicitly is the distinctive feature used by the POC matcher to identify a fingerprint. Ergo this is bound to affect POC's matching accuracy. Generally though, this topic would be a subject for further investigation.

k	bozo3 (%)	VF (%)	GF180 (%)	GF20 (%)	FC (%)	POC (%)
unperturbed	14.81	5.87	11.41	13.61	12.54	22.60
100	13.60	5.94	11.70	13.27		
90	13.48	6.08	12.00	13.86	12.56	22.35
80	14.12	5.96	12.27	14.23		
70	14.10	5.60	11.62	14.23	12.45	22.63
60	14.29	5.47	11.95	13.59		
50	13.77	5.48	12.14	13.86		
40	14.45	5.97	12.57	14.03	12.51	23.82
30	14.46	5.63	12.30	14.18		
20	14.52	6.33	12.97	14.48	12.53	26.02
10	17.93	7.12	15.27	16.15		

Table 5.15: Equal error rates for *Remove Lines* test conducted on sample image database DB1. (Parameter k : "Remove 1 line in every k lines".)

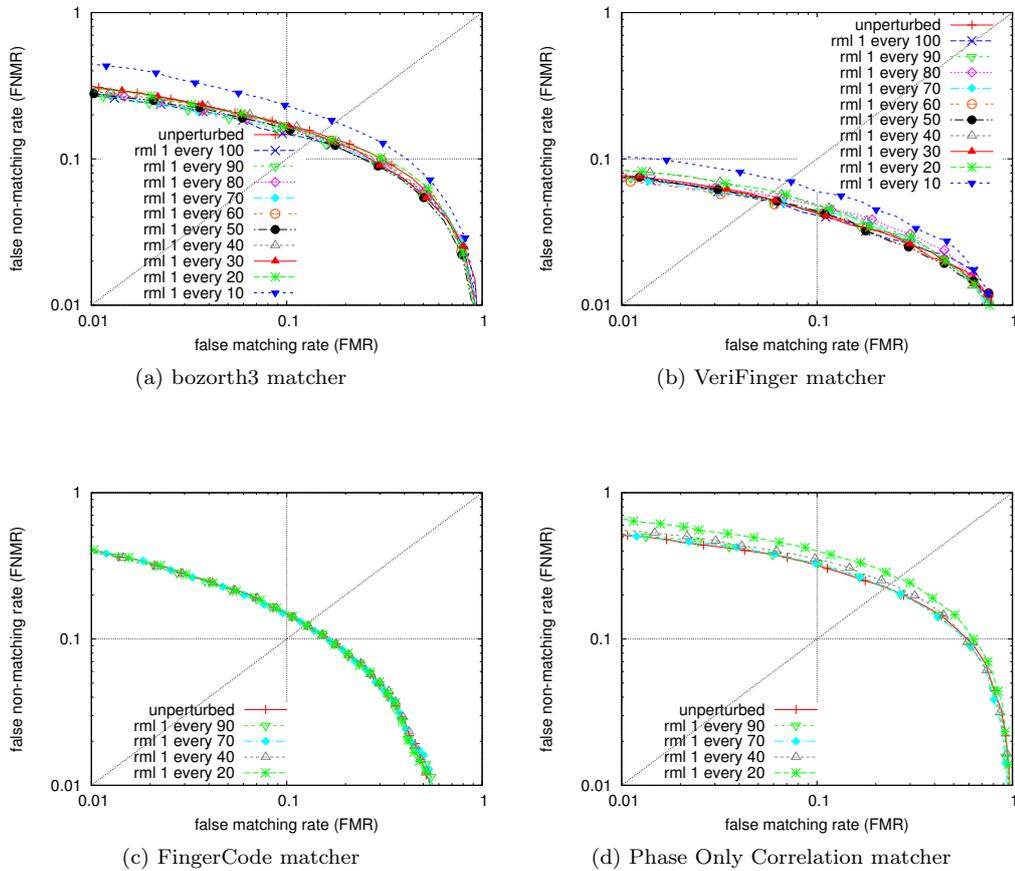


Figure 5.30: Receiver operating characteristics for StirMark test *Remove Lines* on fingerprint images of DB1. (Plots apply a logarithmic scale.)

5.8 Rotations (rot)

Relation to Fingerprints

Rotation is a very typical, not to say: omnipresent, challenge for fingerprint matching, as in the very least cases a finger will be presented twice in exactly the same orientation to the contact area during image acquisition. Thus the *Rotations* test is included in the experiments of present work, aiming to provide the means for comparison of the rotational alignment capabilities of the various fingerprint matchers.

Parameter Configurations

The set of angular values that will be inspected in the experiments of present work is as follows: $\{-20^\circ, -15^\circ, -10^\circ, -5.5^\circ, -5^\circ, 7^\circ, 7.5^\circ, 13^\circ, 18^\circ, 20^\circ\}$. The limited selection, used in the experiments that will also be run on the non-minutiae-base fingerprint matchers, covers the angles $\{-15^\circ, -5.5^\circ, 13^\circ, 20^\circ\}$. Examples for rotations of -15° , -5.5° and 20° can be seen

k	bozo3 (%)	VF (%)	GF180 (%)	GF20 (%)	FC (%)	POC (%)
unperturbed	11.12	5.01	11.72	12.89	9.60	9.69
100	10.86	5.12	11.87	13.02		
90	11.04	5.09	12.43	13.57	9.71	10.00
80	10.85	5.22	12.38	13.41		
70	11.60	5.37	12.09	13.39	9.73	10.24
60	11.28	5.43	12.01	13.70		
50	11.39	5.14	12.21	13.80		
40	11.99	4.96	12.59	13.79	9.47	11.18
30	12.23	5.07	12.70	13.76		
20	12.92	5.27	12.96	14.46	9.97	14.75
10	16.20	5.51	15.18	15.79		

Table 5.16: Equal error rates for *Remove Lines* test conducted on sample image database DB2. (Parameter k : “Remove 1 line in every k lines”.)

k	bozo3 (%)	VF (%)	GF180 (%)	GF20 (%)	FC (%)	POC (%)
unperturbed	6.68	3.60	6.90	5.82	8.98	15.07
100	6.95	3.38	6.84	5.65		
90	7.23	3.30	6.76	5.71	8.68	16.51
80	7.04	3.59	7.23	5.63		
70	7.08	3.61	6.46	5.52	9.14	17.10
60	7.47	3.37	6.99	5.69		
50	7.48	3.47	7.05	5.78		
40	7.53	3.68	7.48	6.08	8.78	18.46
30	7.75	3.09	7.19	5.89		
20	8.52	3.53	7.82	6.27	8.94	23.60
10	11.60	4.25	9.48	7.65		

Table 5.17: Equal error rates for *Remove Lines* test conducted on sample image database DB3. (Parameter k : “Remove 1 line in every k lines”.)

in Figure 5.31

Results and Discussion

As stated above, the aim of the *Rotations* test is to establish how well the individual fingerprint matchers cope with rotation in fingerprint images.

In theory, those matchers, that consider an angular range of $\pm 180^\circ$ for their rotational align-

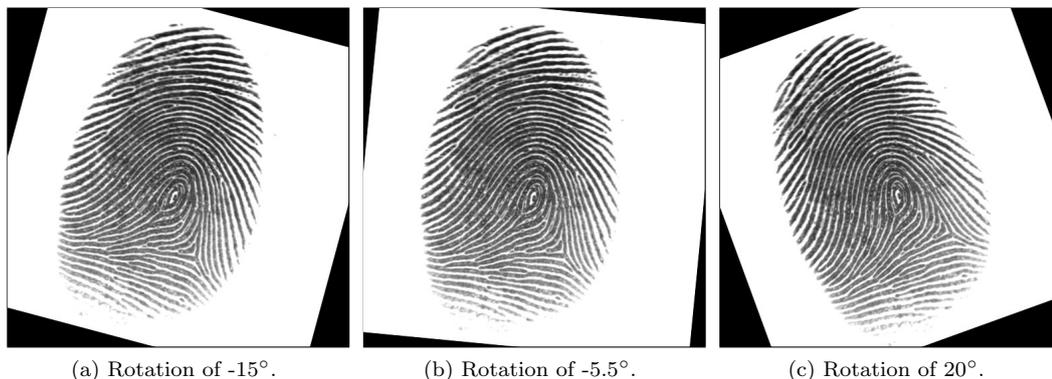


Figure 5.31: Examples for the *Rotations* test, applied to an image from DB1 (ID 91.2).

ment functionality, should not show any noteworthy differences in their results for rotated fingerprint images, compared to the respective results for the original, unperturbed images. On the other hand, those matchers (matcher configurations), where the rotational alignment is limited to $\pm 20^\circ$ (the two non-minutiae-based matchers and the GrFinger configuration GF20) should provide more diverse results – already caused by the fact alone, that some of the rotational displacements that per se existed between two imprints of the same finger in the original fingerprint image sets, will be expanded past the amount of $\pm 20^\circ$.

The test results are conform to our expectations: For those fingerprint matchers (configurations) taking a possible angular range of $\pm 180^\circ$ into account – **bozorth3**, **VeriFinger** and the GrFinger-configuration **GF180** – the effects of rotation in the fingerprint images is very limited, and this is true for fingerprints of all three databases, DB1 to DB3, alike. When for example regarding the listings of equal error rates for said matchers (see the corresponding columns in Tables 5.18, 5.19, 5.20), we find, that besides two exceptions, the equal error rates of the results for rotated images differ no more than $\pm 1\%$ from the respective EER for the original images. The exceptions are for once GF180 for images of DB1 and DB3 rotated at 20° , as here said differences are 1.16% and 1.56% respectively and second bozorth3 for rotated images of DB1. What is special about bozorth3 and rotated fingerprint images of DB1, is, that the results for every rotational step tested are better than the results for the original, unperturbed images – in Figure 5.32a we see, that the ROC curves corresponding to the *Rotations* tests all lie below that for unperturbed DB1 images and when regarding the related equal error rates, we find, that those rates, corresponding to the rotated images, are between 1.37% and 1.87% better than the equal error rate for the unperturbed images.

Also the results of the tests for **GF20** and the non-minutiae-based matchers are conform to afore-stated expectations: While the matching performance of GF180 proves to be largely unaffected by any of the *Rotations* tests performed, inspecting the results of GF20, we can clearly witness a gradually decrease in matching performance with increasing amount of rotational displacement: Referring for example to Table 5.19, to the equal error rates of GF20 for fingerprint images of DB2, we find, that the differences between the EER caused by rotated images and the one for unperturbed images rise, up to an amount of 10.16% at a deliberate rotation of 20° , while the same difference is only 0.03% for GF180.

The **non-minutiae-based matchers** likewise display the effects of the rotations introduced to the fingerprint images, in their results. Yet even though these two matchers were tested

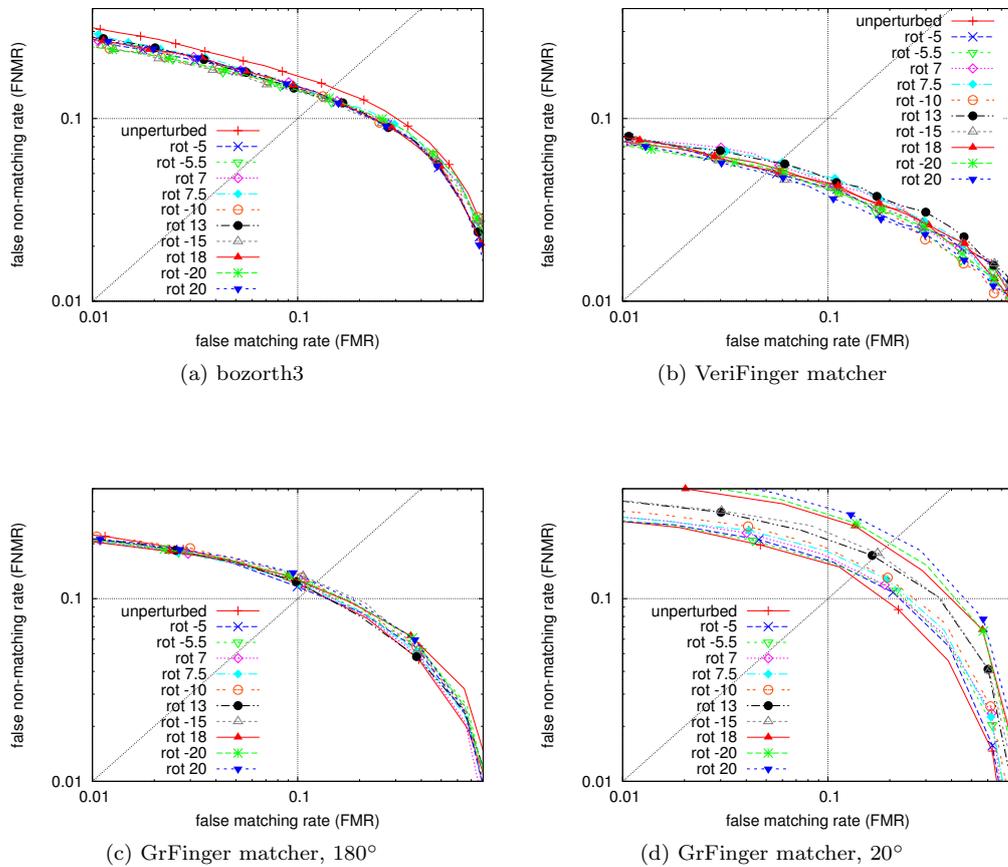
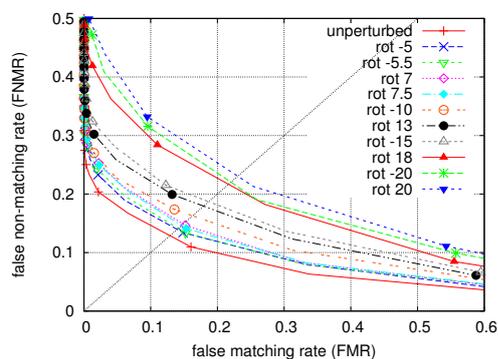
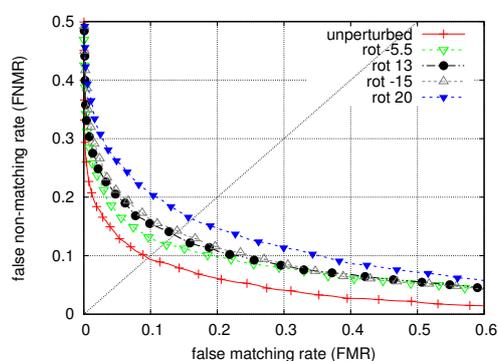


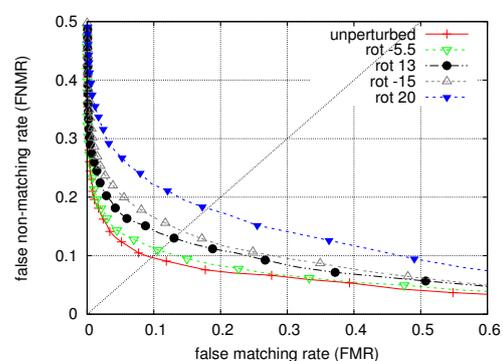
Figure 5.32: Details of the receiver operating characteristics for StirMark test *Rotations* on fingerprint images of DB1. (Plots apply a logarithmic scale. Ranges set to FMR [1%, 80%] and FNMR [1%, 40%])

only on a reduced set of rotational displacement values, they appear to be generally less sensitive to the rotations than GF20. Especially for the higher rotation-levels within the reduced test set – 13° , -15° and 20° – FingerCode and POC show a clearly better matching performance. This behavior can for instance be seen in the corresponding ROC plots (see Figure 5.33 for examples of DB2), where the ROC curves caused by the different rotational values and the respective ROC curve for unperturbed images lie relatively closer together in the plots relating to the non-minutiae-based fingerprint matchers, than they do in the plot of GF20. Further also the corresponding equal error rates show, that the relative degradation due to the influences of the rotations, introduced to the fingerprint images, is less distinct for FingerCode- and POC matcher, than for GF20.

When finally directly comparing the matching performances of the two non-minutiae-based fingerprint matchers, we find, that in case of images of DB3, the FingerCode matcher persistently shows to be less influenced by the deliberate rotations. In cases of DB1 and DB2 on the other hand, only for fingerprint images with an introduced rotation of 20° , FingerCode clearly outperforms the POC matcher, while for the other rotation-levels the results are either equally strong impaired or POC manages to deliver better results.

(a) GrFinger matcher, 20° 

(b) FingerCode matcher



(c) Phase Only Correlation matcher

Figure 5.33: Receiver operating characteristics for StirMark test *Rotations* on fingerprint images of DB2. The plots compare those fingerprint matchers (matcher configurations), that consider only a limited angular range of $\pm 20^\circ$ for possible rotational alignment of two finger imprints.

Rotation	bozo3 (%)	VF (%)	GF180 (%)	GF20 (%)	FC (%)	POC (%)
unperturbed	14.81	5.87	11.41	13.61	12.54	22.60
-20	13.27	5.40	12.14	21.09		
-15	13.00	5.28	12.59	17.66	14.74	24.34
-10	13.25	5.72	12.26	15.54		
-5.5	12.94	5.78	11.80	14.14	12.90	22.67
-5	13.26	5.37	11.25	14.12		
7	13.34	6.44	11.83	14.50		
7.5	13.30	6.01	11.60	15.16		
13	13.05	5.89	11.62	17.02	13.63	23.79
18	13.44	5.36	12.05	20.25		
20	13.41	5.12	12.56	22.38	15.44	26.18

Table 5.18: Equal error rates for *Rotations* test conducted on sample image database DB1.

Rotation	bozo3 (%)	VF (%)	GF180 (%)	GF20 (%)	FC (%)	POC (%)
unperturbed	11.12	5.01	11.72	12.89	9.60	9.69
-20	10.40	5.66	12.28	22.03		
-15	11.00	5.57	12.62	18.70	14.13	14.22
-10	10.36	5.49	11.95	16.29		
-5.5	11.28	5.49	12.62	14.12	12.04	10.89
-5	10.14	5.85	12.23	13.96		
7	11.19	5.47	11.97	14.86		
7.5	10.56	5.50	12.26	14.63		
13	10.59	5.65	12.15	17.93	13.57	13.01
18	10.67	5.58	12.00	21.67		
20	10.94	5.59	11.75	23.05	16.27	18.08

Table 5.19: Equal error rates for *Rotations* test conducted on sample image database DB2.

Rotation	bozo3 (%)	VF (%)	GF180 (%)	GF20 (%)	FC (%)	POC (%)
unperturbed	6.68	3.60	6.90	5.82	8.98	15.07
-20	6.66	3.34	7.65	16.10		
-15	6.83	3.04	7.53	9.28	9.62	16.06
-10	6.79	3.23	6.75	5.82		
-5.5	6.81	3.21	6.99	6.03	9.57	15.96
-5	6.63	3.41	6.92	5.47		
7	6.79	3.39	6.60	5.60		
7.5	6.88	3.46	6.75	5.55		
13	6.72	3.47	6.75	6.67	9.02	15.80
18	7.29	3.16	7.51	10.24		
20	7.41	3.21	8.48	14.01	10.22	18.93

Table 5.20: Equal error rates for *Rotations* test conducted on sample image database DB3.

5.9 Affine Transformations – General Notes

Relation to Fingerprints

The application of affine transformations to fingerprint images is intended to simulate distortions of the entire finger imprint, that can appear in real-life situations during fingerprint acquisition, depending on the way, the finger is pressed on the contact area. In the experiments of present work I will to examine two such distortion types in particular:

- A *shearing* effect can occur, when the force that is exercised while pressing the finger on the contact area is not exerted perpendicular to this area. For example, if the finger is presented, with the user pushing rather in direction to the upper-right of the sensor, than straight downwards.
- A certain *stretching* might appear in the finger imprint, when the amount of force applied while pressing the finger on the contact area is large/larger than usual. And leaving the conceptual model of a fingerprint-acquired-by-scanner situation and considering a scenario that might for example be of forensic interest: Stretching of a fingerprint can also very well appear, if the finger was imprinted on a soft or flexible surface.

As already noted in section 2.2, the StirMark Benchmark provides a generic procedure for applying affine transformations to input images. A specific transformation can then be generated by adequately setting the parameters of the inverse transformation matrix, stated in Figure 5.34.

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix}$$

Figure 5.34: Parameterized inverse transformation matrix applied in *Affine Transformations* tests.

While this enables a user to generate a broad variety of different affine transformation types and especially also combinations thereof, for the experiments of present work, I chose to limited the number of tests to the following four representative basic types: *stretching in X-direction*, *stretching in Y-direction*, *shearing in Y-direction*, *shearing in X- and Y-direction*. The corresponding parameter configurations, example images and experimental results will be presented in the subsequent sections.

5.10 Affine Transformations

Stretching in X-Direction (affineXstretch)

Parameter Configurations

Table 5.21 lists the parameter configurations for the inverse transformation matrix shown in Figure 5.34 as used for the StirMark test *Stretching in X-Direction* in the experiments of present work.

The reduced set of test, that will also be executed on the non-minutiae-based fingerprint matchers, includes configurations {2, 4, 6, 8}. Examples images for configurations 1, 5 and 8 can be found in Figure 5.35.

Results and Discussion

One general observation that can be made about the reaction of all tested fingerprint matchers to perturbed images of all three databases, DB1 to DB3, is, that except for very few,

Configuration	a	b	c	d	e	f
1	1.035	0	0	1	0	0
2	1.070	0	0	1	0	0
3	1.105	0	0	1	0	0
4	1.140	0	0	1	0	0
5	1.175	0	0	1	0	0
6	1.210	0	0	1	0	0
7	1.280	0	0	1	0	0
8	1.350	0	0	1	0	0

Table 5.21: Configurations of the parameters of inverse transformation matrix stated in Figure 5.34 as applied in the StirMark test *Affine Transformations – Stretching in X-Direction*



Figure 5.35: Examples for the *Affine Transformations – Stretching in X-Direction* test, applied to an image from DB1 (ID 91_2).

negligible exceptions, the individual matching performance of each matcher decreases gradually with increasing effectiveness of the stretching-operation applied to the test images. Yet an interesting point here is, that especially for fingerprint images of DB1, some of the matchers “start” with better matching results for the first level(s) of perturbations, than they produce for unperturbed images.

Furthermore, regarding the set of **minutiae-based fingerprint matchers**, we can state, that, as well for all three fingerprint image databases alike, GrFinger turns out to be the matcher the most robust to the perturbations introduced in the images, with the configuration GF20 (matching with rotational alignment limited to an angular range of $\pm 20^\circ$) noticeably outperforming the configuration GF180. Except in 4 of the total 3×8 *Stretching in X-Direction* tests performed, VeriFinger then shows relatively stronger impaired matching results than GF180 (for details please refer to the listings of the individual equal error rates in Tables 5.22, 5.22 and 5.22). bozorth3 finally is clearly the minutiae-based matcher, whose matching performance is influenced the most by the perturbations introduced in the fingerprint images of databases DB1 to DB3 – especially for the higher test-configuration levels, the relative distance to the results of aforementioned matchers is quite obvious, in

the related ROC plots and the corresponding equal error rates alike.

Still, of all fingerprint matcher considered, the one fingerprint matcher, that continuously shows the highest sensitivity to the image manipulations of the *Stretching in X-Direction* StirMark tests, is the Phase Only Correlation matcher.

About the results of the FingerCode matcher no meaningful general statement can be made, but a more detailed analysis is required.

DB1

Overall GrFinger appears to be the matcher, that is the least influenced by the *Stretching in X-Direction* perturbations introduced in the fingerprint images of database DB1 – with configuration **GF20** being even slightly more robust than **GF180**. A look at Figure 5.36b, for example, reveals, how close the ROC curves for the test-configurations 1 to 6 are to the ROC curve for unperturbed images, with the curves relating to the less-effective configurations even lying below it. The corresponding equal error rates, listed in Table 5.22, also confirm, that the EER for configurations 1 to 6 are no farther distant from the EER for the original, unperturbed images, than 0.54% for GF20 and 1.17% for GF180. For the remaining two configurations, 7 and 8 then, we notice a more distinct impairment in the matching results – visible also in the respective plots, as the ROC curves related to these stronger perturbation levels appear distinctively separated from the set of the other curves.

Even though the **FingerCode** matcher was only tested on the reduced set of *Stretching in X-Direction* StirMark tests, its results are still quite interesting, as they put FingerCode's matching performance right between those of GF20 and GF180. Hard to distinguish from the ROC plots, the listing of equal error rates in Table 5.22 leads clearly to the conclusion, that in each of the conducted tests, for images of DB1, FingerCode is less influenced by the perturbations introduced in the fingerprint images than GF180, yet more than GF20 shows to be.

The behavior of the **VeriFinger** matcher across the tests for DB1 is quite similar to that of GF180 and also in the respective equal error rates for VeriFinger, we see only a slight deterioration. As for bozorth3 on the other hand, the negative influences caused by the perturbations in the fingerprint images are definitely more apparent – for instance in Figure 5.36a, where we can see, that the ROC curves relating to the various test configurations, are decidedly more spread than in the plots for GF20 and FingerCode. Furthermore when inspecting Table 5.22 again, we find, that despite a relative good performance of bozorth3 for the two weakest perturbation levels, the matching performance then degrades faster than in any other of the aforementioned fingerprint matchers.

Yet, as already mentioned previously, when listing the general observations, the **Phase Only Correlation** matcher is undoubtedly the one, whose results are the most impaired by the effects of the *Stretching in X-Direction* tests. When regarding the relative difference between the equal error rates caused by perturbed images and the respective EER for the original, unperturbed images, we can make the following exemplary observations: Already for configuration 2 the EER is 1.42% worse than that for unperturbed images, which is a difference, that bozorth3 reaches only at configuration 4 or GF20 only at configuration 7. Further the relative difference of POC for configuration 4, 4.38%, is larger than the maximum differences reached in GF20 and FingerCode and for configuration 6 also the maximum differences reached in GF180 and VeriFinger have been surpassed. (For further details please refer to Table 5.22).

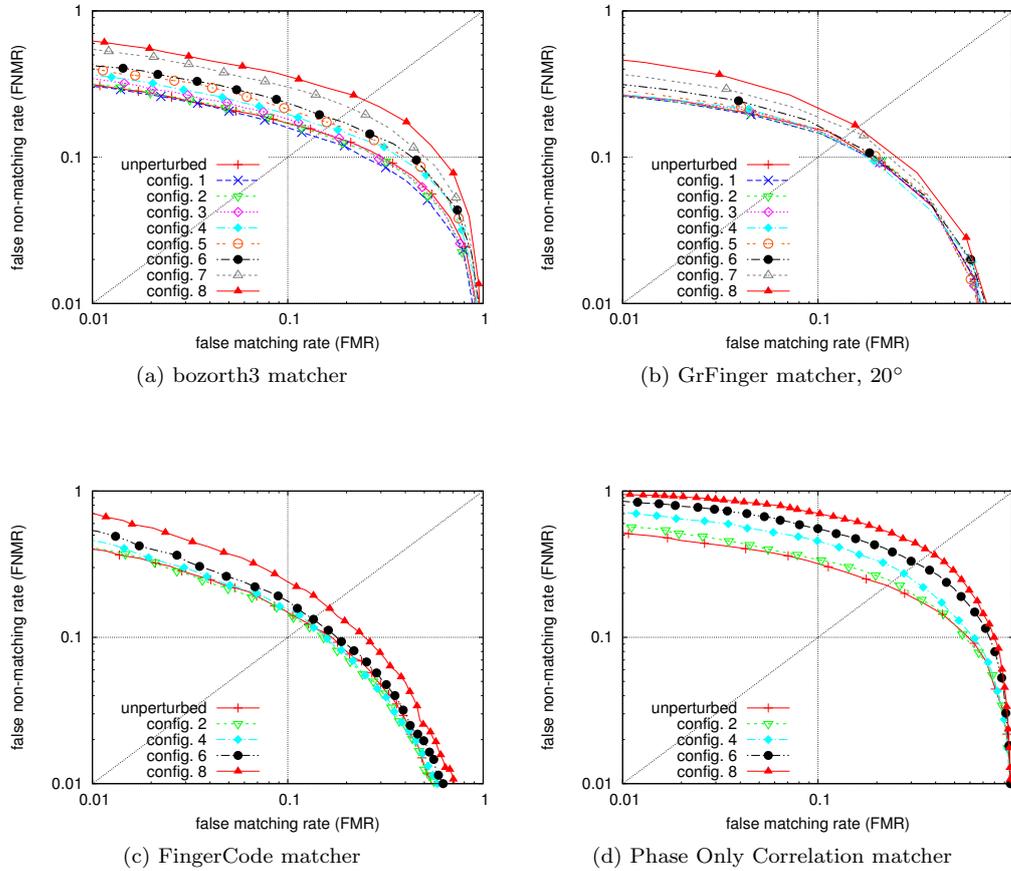


Figure 5.36: Receiver operating characteristics for StirMark test *Affine Transformations – Stretching in X-Direction* on fingerprint images of DB1. (Plots apply a logarithmic scale).

Configuration	bozo3 (%)	VF (%)	GF180 (%)	GF20 (%)	FC (%)	POC (%)
unperturbed	14.81	5.87	11.41	13.61	12.54	22.60
1	13.85	5.51	11.64	13.07	12.38	23.06
2	14.56	5.86	11.11	13.15	12.27	24.02
3	14.95	5.82	11.05	13.24		
4	16.10	6.78	12.14	13.28	12.73	27.43
5	16.93	7.50	12.20	13.58		
6	17.95	7.79	12.58	14.00	13.40	31.96
7	21.51	9.87	15.47	15.44		
8	24.78	12.45	17.11	16.17	16.08	37.91

Table 5.22: Equal error rates for *Affine Transformations – Stretching in X-Direction* test conducted on sample image database DB1.

DB2

In comparison to each other, the matching performances of the **minutiae-based fingerprint matchers**, related to the various *Stretching in X-Direction* tests for images of database DB2, are very much alike to those exhibited in the case of DB1-images. Only the influences of the perturbations are more distinctive in the individual results of current tests.

Like for images of DB1, the matcher that proves the most robust to the image-manipulations performed by the tests, is GrFinger, with **GF20** again slightly outperforming **GF180**. As can be seen in Figure 5.37b and in the corresponding equal error rates in Table 5.23, the results for test configurations 1 to 5 are still relatively close to the results for the original, unperturbed images, yet not as close, as was the case for images of DB1. For example: for DB1-images and matcher GF20 the differences between the equal error rates for perturbed images and the equal error rate for the original images get larger than 1% only for configurations 7 and 8. In the tests for DB2 this happens already from configuration 5 on.

Comparing **VeriFinger** and GF180, we once again see a very similar behavior. It is only for configuration 8, that the influence of the perturbations causes the results of VeriFinger to be notably worse than those of GF180 – with the corresponding EER of VeriFinger showing 2.25% more difference to the EER for unperturbed images, than GF180, while for configurations 1 to 7 the respective relative impairments permanently differ by less than 1%.

The **bozorth3** matcher, as well, displays a reaction to the perturbations, introduced into the fingerprint images of DB2, that is quite similar to that for images of DB1: A good performance for the test configurations 1 and 2, but then the effects of the perturbations manifest themselves comparatively stronger in the matching results of bozorth3 than they do in the results of the other minutiae-based matchers. Interestingly though, here, for the DB2-related *Stretching in X-Direction* StirMark tests, **FingerCode**'s matching performance is very much akin to that of bozorth3. When regarding Table 5.37c and analyzing both matchers' equal error rates in relation to the respective equal error rates for unperturbed fingerprint images, we find, that the relative degradations are quite close. Also when looking at the ROC curves in Figures 5.37a and 5.37d we can detect a certain degree of resemblance. However, one general difference we also see in the ROC plots is, that for FMR below the EER, the FingerCode matcher tends to have higher FNMR than bozorth3, while on the other hand for FMR above the EER, bozorth3's FNMR tend to be comparatively higher.

Like was the case for fingerprint images of DB1, also here, for images of DB2 the perturbations introduced by the *Stretching in X-Direction* test have the strongest impact on the matching performance of the **Phase Only Correlation** matcher. This observation is most comprehensible in Figure 5.37d, when regarding the comparatively large separation of the individual ROC curves.

DB3

When introducing the perturbations of the *Stretching in X-Direction* tests into fingerprint images of database DB3, we find, that overall the fingerprint matchers display largely analog reactions to those, previously shown for images of DB2. Only a detailed inspection of the individual matching results reveals rather small differences in case of the **minutiae-based matchers** and comparatively larger differences in case of the **non-minutiae-based fingerprint matchers**:

First of all, when considering both **GrFinger**-configurations and regarding Table 5.24, we

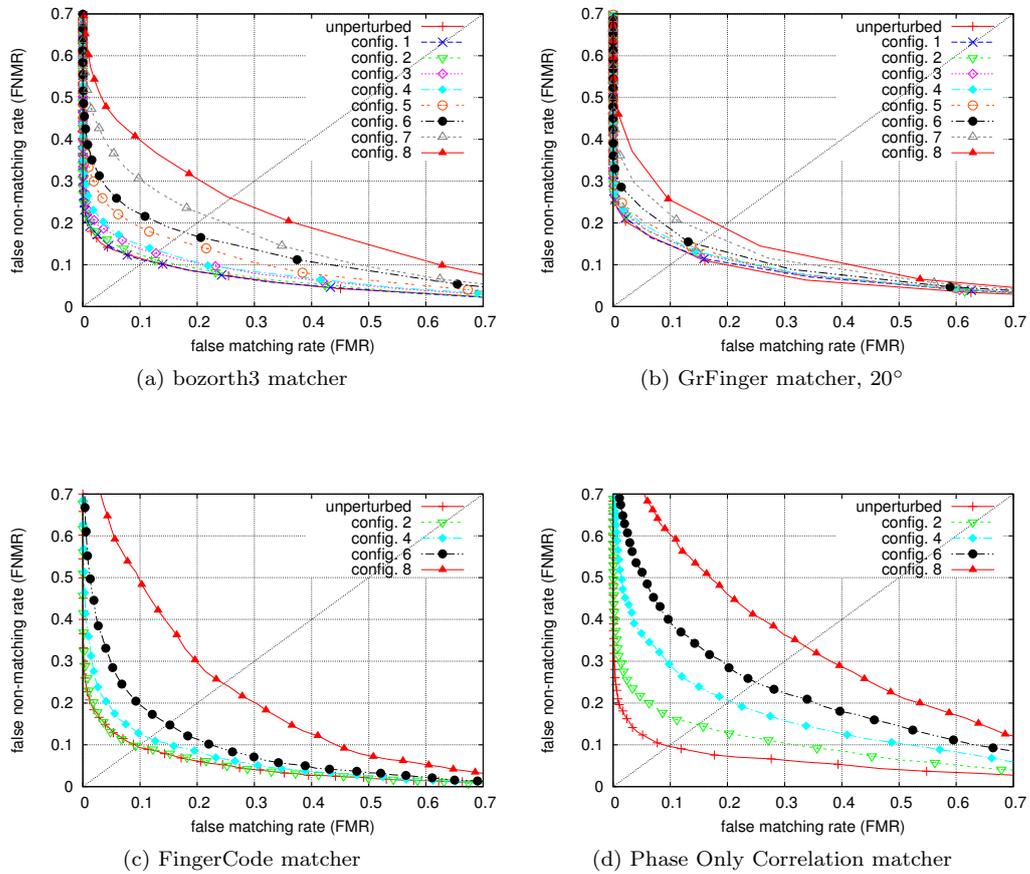


Figure 5.37: Receiver operating characteristics for StirMark test *Affine Transformations – Stretching in X-Direction* on fingerprint images of DB2.

Configuration	bozo3 (%)	VF (%)	GF180 (%)	GF20 (%)	FC (%)	POC (%)
unperturbed	11.12	5.01	11.72	12.89	9.60	9.69
1	10.96	5.65	12.00	13.00		
2	11.63	5.96	12.37	13.63	9.73	14.83
3	12.76	5.98	12.62	13.49		
4	13.28	6.59	12.93	13.64	11.60	20.47
5	16.01	7.41	13.38	14.17		
6	17.76	8.37	15.22	14.84	14.96	24.96
7	21.69	10.80	17.38	17.21		
8	25.90	15.05	19.51	19.09	24.64	33.59

Table 5.23: Equal error rates for *Affine Transformations – Stretching in X-Direction* test conducted on sample image database DB2.

see what also the ROC plots (see the example for GF20 in Figures 5.38b) hints at: The impairment caused by the perturbations in the fingerprint images on the matching results is slightly less, yet basically equally strong as it was in the corresponding results for images of DB2. In fact, when regarding the differences between the EER for perturbed images and the respective EER for unperturbed images, the values for DB3 are all smaller than those for DB2, yet by no more than 1.18%.

The same can be said about the relative matching performance of **VeriFinger**, with one exception: For configuration 8 the perturbations in the images of DB3 lead to clearly less impaired results than they did for images of DB2. Hence VeriFinger's performance is now almost similar to that of GF180.

When comparing the results of **bozorth3** for the StirMark tests on images of DB3 with those on images of DB2, we see a general, mild aggravation.

As we noted in the *Stretching in X-Direction* tests for images of DB1 and DB2 alike, also here, for images of DB3, **bozorth3** is the minutiae-base fingerprint matcher, that is the most sensitive to the perturbations introduced in the fingerprint images. When comparing the results for images of DB3 with those of DB2 we even see a further mild aggravation – particularly in the matching results related to test configurations 4, 7 and 8. This behavior is for example observable in Figure 5.38a, where the corresponding ROC curves are farther separated from the ROC curve for unperturbed images, as they were in the ROC plot for DB2 images, Figure 5.37a.

While for perturbed fingerprint images of DB2 the **FingerCode** matcher exhibits a better performance than bozorth3 for every configuration part of the reduced test-set, except for configuration 8, here, for images of DB3, bozorth3 permanently outperforms the FingerCode matcher. The biggest differences appear for configurations 2 and 6 – as is also very obvious when comparing the corresponding ROC plots in Figures 5.37a and 5.38c) or referring to the respective equal error rates in Table 5.24.

When likewise inspecting the results for the **Phase Only Correlation** matcher, it becomes very obvious, that for the *Stretching in X-Direction* tests on fingerprint images of DB3, the POC matcher once again exhibits the strongest-impaired matching performance of all regarded matchers. Additionally the results for images of DB3 are considerably worse than the results POC produced for images of DB1 or DB2. Accordingly, when regarding the ROC plot in Figure 5.38d, we see, how comparatively far the ROC curves for the various test configurations lie from the ROC curve representing the results for the original, unperturbed images. The listing of equal error rates in Table 5.24 confirms the observations. Therein we find for example, that for the second and third worst matcher in regard to the *Stretching in X-Direction* tests – FingerCode and bozorth3 – the EER for configuration 8 is worse than the respective EER for unperturbed images by 16.81% and 16.69% respectively. In case of the POC matcher, the EER of configuration 4 is already 16.65% worse, than its EER for unperturbed images and the difference for configuration 8 is even 27.62%.

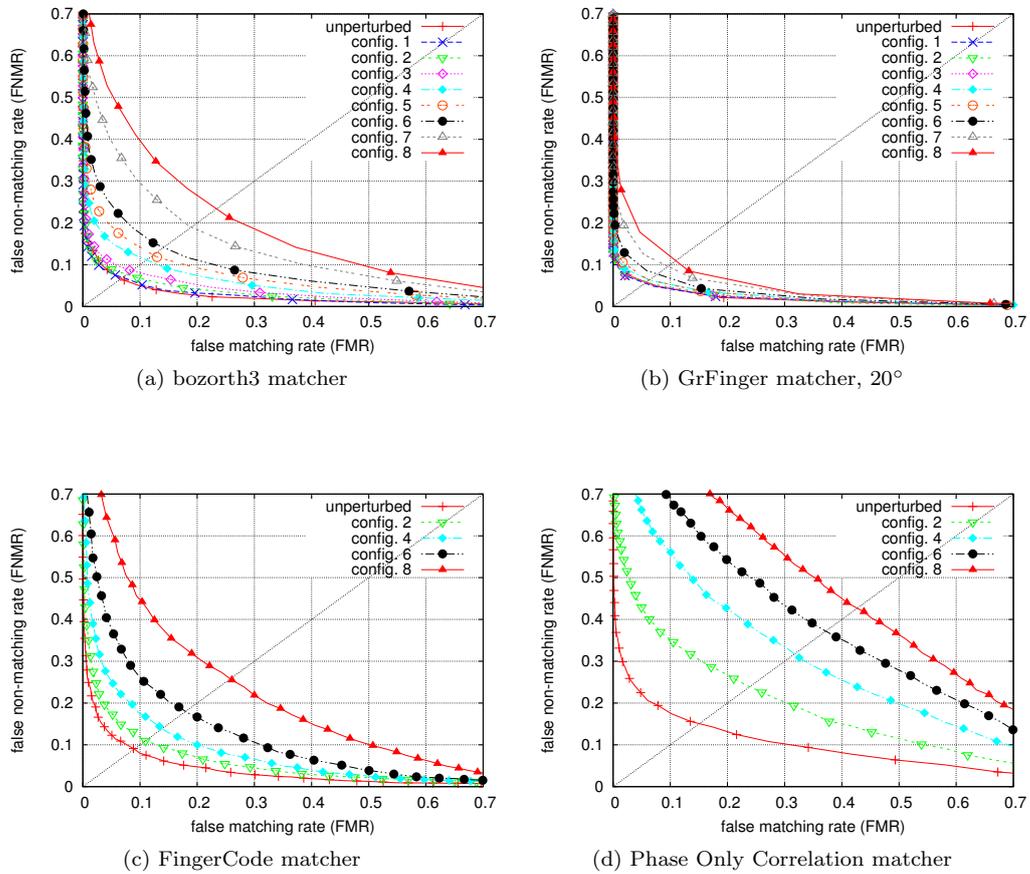


Figure 5.38: Receiver operating characteristics for StirMark test *Affine Transformations – Stretching in X-Direction* on fingerprint images of DB3.

Configuration	bozo3 (%)	VF (%)	GF180 (%)	GF20 (%)	FC (%)	POC (%)
unperturbed	6.68	3.60	6.90	5.82	8.98	15.07
1	6.95	3.41	7.01	5.59		
2	7.70	3.86	6.80	5.88	10.99	23.89
3	8.57	3.61	6.89	5.67		
4	11.13	4.80	7.63	6.38	13.90	31.72
5	12.33	5.67	8.34	6.37		
6	14.14	6.44	9.77	7.75	17.98	37.20
7	18.99	8.60	11.38	9.61		
8	23.38	11.18	14.09	10.96	25.79	42.69

Table 5.24: Equal error rates for *Affine Transformations – Stretching in X-Direction* test conducted on sample image database DB3.

5.11 Affine Transformations

Stretching in Y-Direction (affineYstretch)

Parameter Configurations

Table 5.25 lists the parameter configurations for the inverse transformation matrix shown in Figure 5.34 as used for the StirMark test *Stretching in Y-Direction* in the experiments of present work.

Configuration	a	b	c	d	e	f
1	1	0	0	1.035	0	0
2	1	0	0	1.070	0	0
3	1	0	0	1.105	0	0
4	1	0	0	1.140	0	0
5	1	0	0	1.175	0	0
6	1	0	0	1.210	0	0
7	1	0	0	1.280	0	0
8	1	0	0	1.350	0	0

Table 5.25: Configurations of the parameters of inverse transformation matrix stated in Figure 5.34 as applied in the StirMark test *Affine Transformations – Stretching in Y-Direction*

The reduced set of test, that will also be executed on the non-minutiae-based fingerprint matchers, includes configurations {2, 4, 6, 8}. Examples images for configurations 1, 5 and 8 can be found in Figure 5.39.

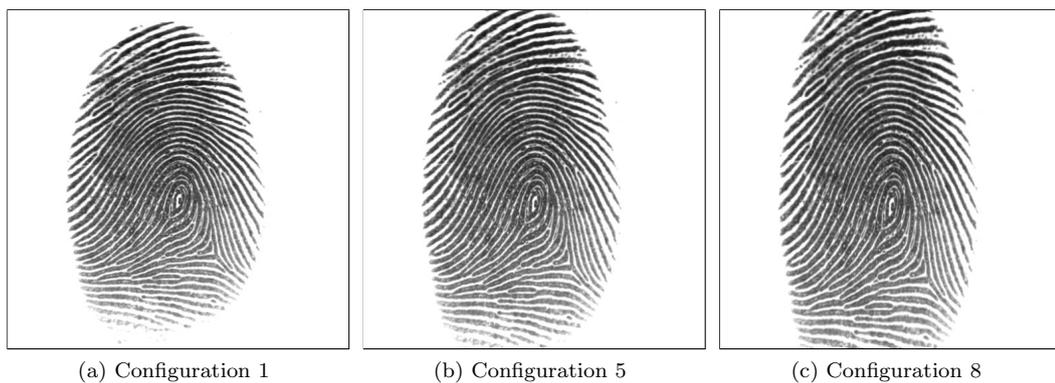


Figure 5.39: Examples for the *Affine Transformations – Stretching in Y-Direction* test, applied to an image from DB1 (ID 91.2).

Results and Discussion

DB1

When comparing the individual matching performances of the regarded fingerprint matchers (configurations) in relation to the perturbations introduced in fingerprint images of database DB1 by the *Stretching in Y-Direction* StirMark test, the most noticeable observation is, that, like in the related StirMark test *Stretching in X-Direction*, the **Phase Only Correlation** matcher is again the matcher, whose results exhibit the most distinct negative influences by the perturbations. When regarding POC's equal error rates in Table 5.26, we find for example, that already for test configuration 2, the corresponding EER is further apart from the EER for unperturbed images, than is the case for most of the other matchers for configuration 4.

When on the other hand looking for the fingerprint matcher that is the least influenced by the perturbations introduced in images of DB1, the answer is already a bit more complicated: Up until test-configuration 5, it is the **bozorth3** matcher, the proves to be the most robust. When regarding the ROC plot in Figure 5.40a, we see, that for configurations 1 to 3 the ROC curves even lie below that representing the results of the original, unperturbed images. A look at Table 5.26 then confirms, that the equal error rates of bozorth3 for configurations 1 to 3 are 1.48% to 1.14% smaller than that for unperturbed images. However, from configuration 6 on, bozorth3's results exhibit a distinct drop in matching performance, making it only the fourth best matcher, behind GF20, VeriFinger and FingerCode.

So, the second least influenced matcher in regard to perturbed fingerprint images of DB1, and the one, that takes the "lead" after bozorth3's drop for configuration 6, is GrFinger in configuration **GF20** (GF20 only considers a limited angular range of $\pm 20^\circ$ for its rotation alignment attempts), closely followed by the VeriFinger matcher. In fact, the manifestation of the perturbations' influence in the results of GF20 and VeriFinger is so much alike, that when regarding both matchers' equal error rates in Table 5.26 and calculating the relative impairments per test configuration level (by building the difference between the EER for unperturbed images and the EER for the respective perturbed images), the values of GF20 and VeriFinger never differ more than 1%.

GF180 then turns out to be the minutiae-based matcher, that shows the most sensitivity to the perturbations of the *Stretching in Y-Direction* tests for images of DB1. When looking at the listing of equal error rates in Table 5.26 and for example comparing those of GF180 with those of GF20, we will notice, that especially for higher test-configuration levels the actual EER values of both matchers are very close, or GF180's is even higher. Yet when taking into account, that GF20's EER for unperturbed images is already 2.20% worse than that of GF180, the relatively stronger impairment of the results of GF180 gets apparent. Furthermore we find, that also the FingerCode matcher – for the limited set of test configurations that it was tested on – shows less impairment in its EER values than GF180.

In overall comparison, we can state, that the **FingerCode** matcher exhibits a worse matching performance than VeriFinger, yet its results are still less influenced by the perturbations in fingerprint images of DB1, than those of GF180 and of the Phase Only Correlation matcher.

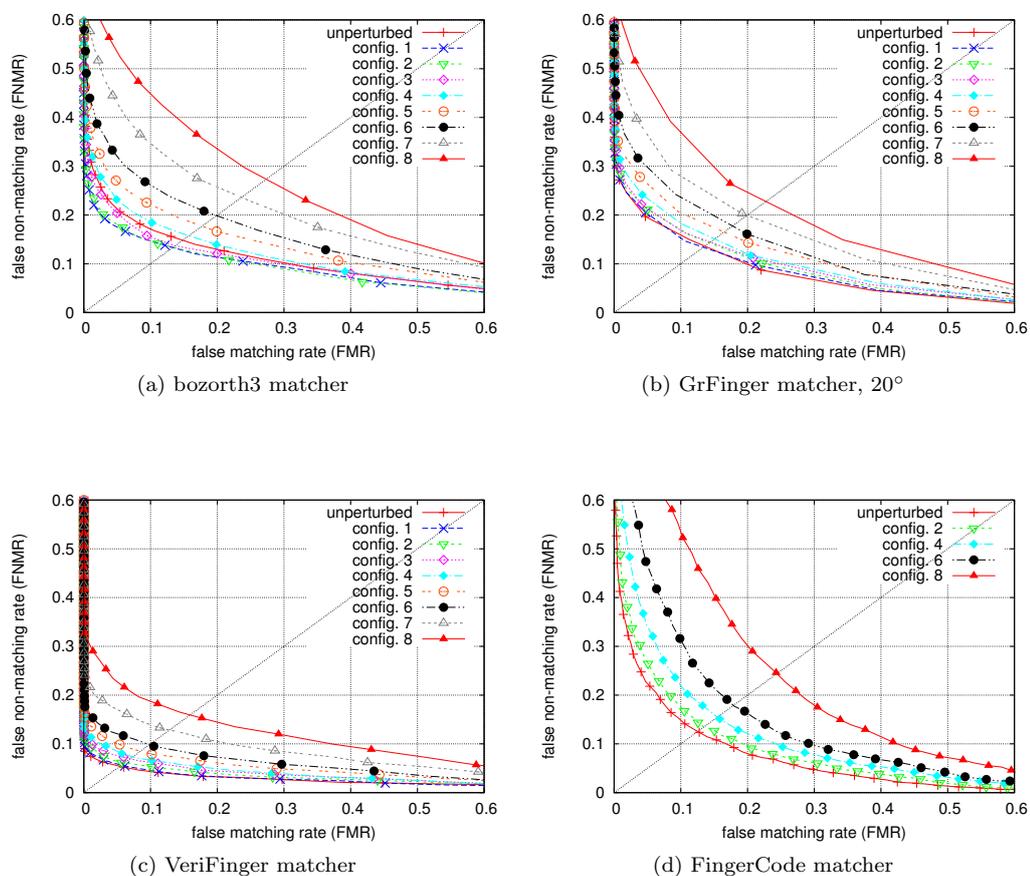


Figure 5.40: Receiver operating characteristics for StirMark test *Affine Transformations - Stretching in Y-Direction* on fingerprint images of DB1.

Configuration	bozo3 (%)	VF (%)	GF180 (%)	GF20 (%)	FC (%)	POC (%)
unperturbed	14.81	5.87	11.41	13.61	12.54	22.60
1	13.33	6.23	12.08	13.38	12.75	23.56
2	13.38	6.13	12.49	14.29	13.48	25.22
3	13.67	7.17	13.57	14.43		
4	15.75	7.40	14.56	15.08	15.49	30.63
5	17.72	9.11	15.40	16.49		
6	19.88	10.21	17.69	17.75	17.88	37.20
7	23.62	12.96	20.46	19.91		
8	27.33	16.01	23.98	22.76	24.40	45.71

Table 5.26: Equal error rates for *Affine Transformations - Stretching in Y-Direction* test conducted on sample image database DB1.

DB2

Most noticeable, also for the *Stretching in Y-Direction* test on images of DB2 the **Phase Only Correlation** matcher exhibits a clearly stronger impairment in its matching results, than any other of the examined fingerprint matchers. Already at first glance, the ROC plot of the POC matcher shows, how decidedly farther the ROC curves are spread, than they are in the plots for the other matchers. Also when regarding Table 5.27, we find, that already for the lowest perturbation level tested on the POC matcher – test configuration 2 – the distance between the corresponding EER and that for unperturbed images is 4.38%. Equally strongly impaired results are produced by the other matchers only from configurations 5 or 6 onwards.

The **bozorth3** matcher shows a comparatively high robustness to perturbations introduced by the two least efficient *Stretching in Y-Direction* test configurations – 1 and 2 – into the fingerprint images of DB2. Inspecting Figure 5.41a we can even observe, that the ROC curve corresponding to configuration 1 lies almost on and in parts even below the ROC curve representing bozorth3’s matching results for the original, unperturbed images. Yet like for images of DB1, also in the current tests, the higher-valued test configurations have a comparatively strong impact on bozorth3’s matching performance, leading to a more distinct gradual aggravation of the matching results, than we witness for the other matchers but POC.

Comparing the other three minutiae-based fingerprint matchers based on their respective equal error rates (see Table 5.27), we can state, that latest from test-configuration 3 on, **GF20**’s matching performance is clearly the least sensitive to the perturbations introduced in the fingerprint images of DB2. VeriFinger’s equal error rates reveal a slightly stronger impairment, yet the amount of aggravation per configuration level is quite alike to that of GF20. Regarding the equal error rates of **GF180** then, for configuration 2 to 5, the same can be said as for VeriFinger. For the more effective configurations, 6 to 8, though, the impact of the corresponding perturbations is more conspicuous than is the case for GF20 or VeriFinger. For illustration we can regard the actual EER values of GF180 and GF20: The initial EER for unperturbed images of GF180 is 1.16% below that of GF20 and the individual EER of GF180 stay below those of GF20, up until configuration 5. For configurations 6 to 8, though, GF180’s EER are permanently around 1% higher than the corresponding ones for GF20.

As stated above, when regarding the respective sets of equal error rates, both matchers exhibit a largely comparable amount of impairment, caused by the perturbations in the fingerprint images, with GF20 slightly outperforming **VeriFinger**. When additionally regarding the matchers’ ROC plots and comparing the corresponding ROC curves of both matchers, we can observe a likewise interesting behavior in the area of high(er) FMR (above the EER) and low(er) FNMR (below the EER): In case of VeriFinger, the curves relating to matcher configurations 1 to 4 trend relatively fast towards the curve representing the results for unperturbed images and for FMR > 25% about, they get to lie in a very narrow area around the latter. The ROC curves for configurations 5 to 8 also strive towards the curve for unperturbed images, yet for the most part they stay clearly separated from aforesaid group of curves. For matcher GF20 in contrast, the ROC curves corresponding to configurations 1 to 4 strive not so much for the curve for unperturbed images, but stay in a rather narrow band clearly above it. On the other hand though, the curves for configurations 5 to 8 approach way faster the set of curves for the lower configurations, than they do in the ROC plot for VeriFinger. So in summary, when comparing the responses of GF20 and

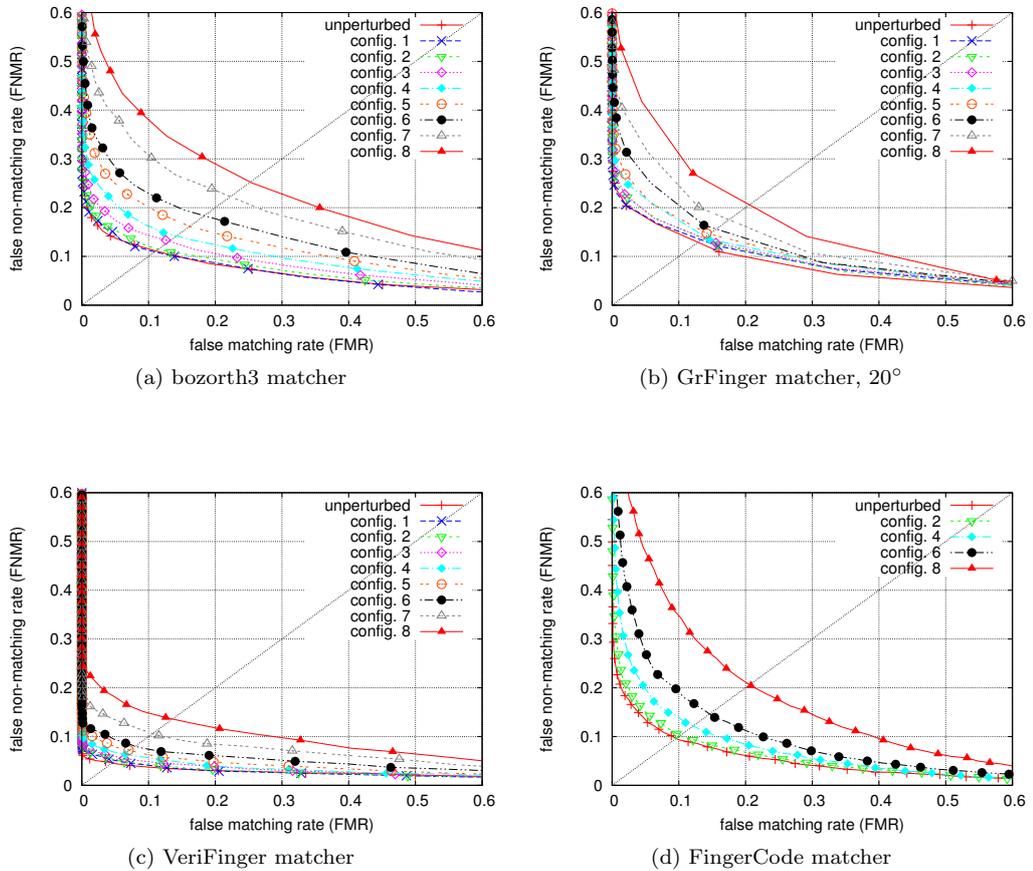


Figure 5.41: Receiver operating characteristics for StirMark test *Affine Transformations - Stretching in Y-Direction* on fingerprint images of DB2.

VeriFinger to perturbations introduced by test *Stretching in Y-Direction* in images of DB2 and observing the FNMR against increasing FMR, we find, that for test-configurations 1 to 4 the FNMR in VeriFinger's results reach the level of unperturbed images faster than the do in GF20's results, while for test-configurations 5 to 8 the FNMR drop faster in results of GF20.

The **FingerCode** matcher turns out to be more sensitive to the perturbations in fingerprint images of DB2 than it is for images of DB1. Comparing the respective equal error rates (see Tables 5.26 and 5.27), regarding the *Stretching in Y-Direction* tests for fingerprint images of DB1, FingerCode's rates exhibit less impairment than those of GF180 and bozorth3. In case of current test, for images of DB2, the FingerCode matcher is only able to outperform bozorth3 but not GF180 anymore. Also the corresponding exemplary plots in Figure 5.41 portray this behavior quite clearly. Especially the ROC curves relating to test configurations 6 and 8 make the considerably large discrepancy in the matching performances of bozorth3 and FingerCode apparent.

Configuration	bozo3 (%)	VF (%)	GF180 (%)	GF20 (%)	FC (%)	POC (%)
unperturbed	11.12	5.01	11.72	12.89	9.60	9.69
1	10.93	5.55	11.80	13.38		
2	11.51	5.24	12.64	13.67	10.20	14.52
3	13.11	5.91	13.05	13.76		
4	14.19	6.76	13.62	13.99	12.15	21.22
5	16.16	7.61	14.41	14.60		
6	18.35	8.07	16.66	15.60	14.63	27.14
7	22.27	10.90	18.70	17.52		
8	25.24	13.66	21.48	20.58	20.61	33.97

Table 5.27: Equal error rates for *Affine Transformations – Stretching in Y-Direction* test conducted on sample image database DB2.

DB3

Also in the *Stretching in Y-Direction* tests for fingerprint images of DB3, the **Phase Only Correlation** matcher is the one, in whose matching results the negative influence of the introduced image-perturbations is most intelligible. Once again, in the ROC plot for the POC matcher the individual ROC curves are farther separated from each other, than in any other matcher’s plot and accordingly distinct is also the increase in POC’s EER values (see Table 5.28 for details).

Interestingly, in relation to the fingerprint images of DB3, now the **FingerCode** matcher shows the second highest sensitivity to perturbations caused by the *Stretching in Y-Direction* StirMark tests. Already the comparison of the ROC plots for FingerCode and for bozorth3 – the third most sensitive matcher in current tests for DB3 – in Figures 5.42d and 5.42a shows quite clearly the relatively stronger impairment in FingerCode’s matching results – in particular, in the plot for the FingerCode matcher, the ROC curves representing the matching performances for test configurations 6 and 8 are unmistakably farther distant from the respective ROC curve for original, unperturbed images than they are in the plot for bozorth3.

Another major difference to the results of the *Stretching in Y-Direction* tests for fingerprint images of DB1 and DB2 can be seen in **VeriFinger**’s reactions to the perturbations introduced in the images of DB3: For test configuration 1, VeriFinger behaves like **GF20**, in that it exhibits quasi no influence in its results at all. From configuration 2 onwards though, VeriFinger shows decidedly more robustness to the perturbations, than any other matcher regarded in the tests. Even GF20, being the strongest matcher in the tests for DB1 and DB2, is clearly outperformed by the VeriFinger matcher. This behavior becomes very evident, when regarding the corresponding ROC plots in Figure 5.42. For further illustration we can have a look at the equal error rates in Table 5.28: Calculating per matcher the differences between the EER for unperturbed images and the EER for the respective test-configurations, aiming to establish the individual deteriorative effect of the image-perturbations, we find, that from configuration 2 on, GF20’s difference-values are between 0.65% and 1.91% higher than those of VeriFinger.

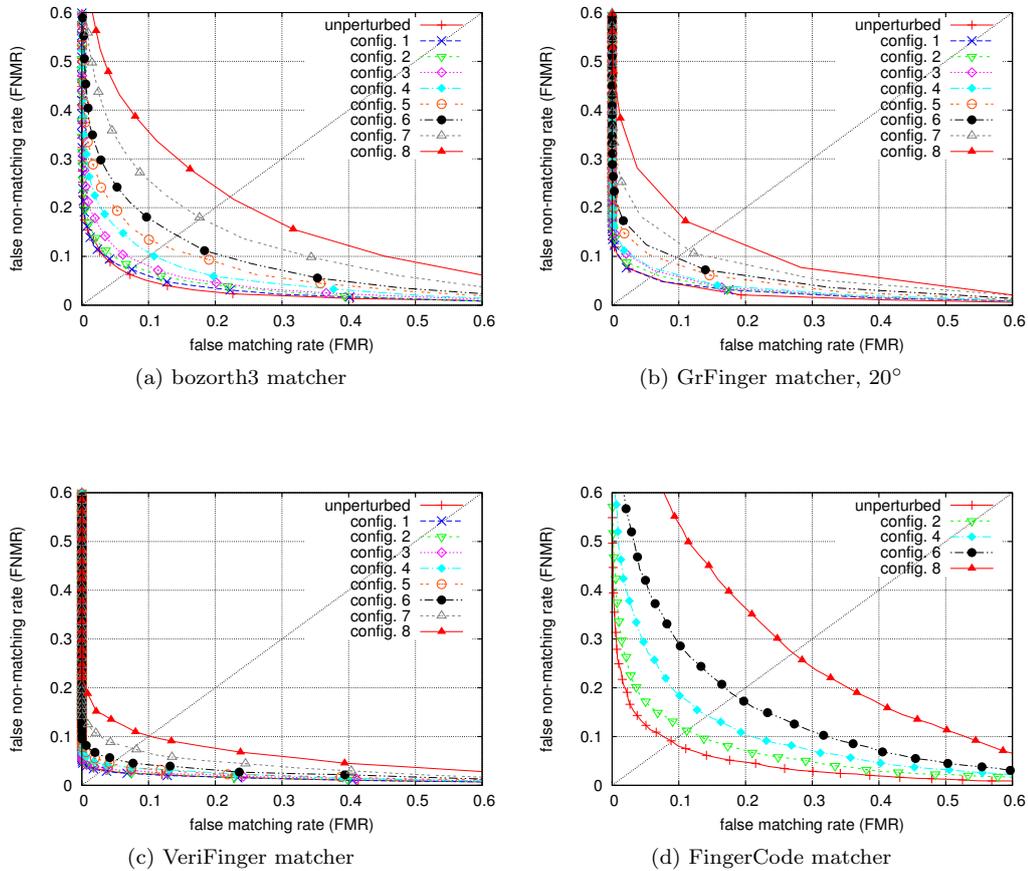


Figure 5.42: Receiver operating characteristics for StirMark test *Affine Transformations - Stretching in Y-Direction* on fingerprint images of DB3.

Comparing the two GrFinger configurations with each other, for fingerprint images of DB3 we see again the effect, that GF20 proves to be more robust to the introduced perturbations than GF180. Even though the amount by which GF180's results are stronger impaired than GF20's, is slightly smaller than it was in the tests for images of DB1 and DB2.

As for **bozorth3**, the matcher once more turns out to be the minutiae-based matcher, that is the most sensitive to the attacks of the *Stretching in Y-Direction* tests. As can be seen by comparing the exemplary ROC plots in Figure 5.42 and the equal error rates in Table 5.28 alike, the higher the effectiveness of the test-configuration, the higher the relative aggravation of the matching results, in comparison to the impairment witnessed in the results of the other minutiae-based matchers. For illustration: The equal error rates for unperturbed images are quite equal for bozorth3 and for GF180 – 6.68% and 6.90% respectively. For low-valued test-configurations they are still quite close, bozorth3's EER only slightly worse. Yet latest from configuration 4 onwards, bozorth3's matching performance drops rapidly, so that finally, for test-configuration 8, its EER is 4.87% worse than that of GF180.

Configuration	bozo3 (%)	VF (%)	GF180 (%)	GF20 (%)	FC (%)	POC (%)
unperturbed	6.68	3.60	6.90	5.82	8.98	15.07
1	7.38	3.57	7.23	5.75		
2	7.90	3.52	8.02	6.39	11.23	18.22
3	8.80	3.61	8.73	6.89		
4	10.40	4.35	8.89	7.20	14.25	23.99
5	12.23	4.70	10.53	8.82		
6	14.29	5.83	11.21	9.72	18.40	29.69
7	17.83	7.86	14.41	11.42		
8	22.22	10.20	17.57	15.03	27.12	37.67

Table 5.28: Equal error rates for *Affine Transformations – Stretching in Y-Direction* test conducted on sample image database DB3.

5.12 Affine Transformations Shearing in Y-Direction (affineYshear)

Parameter Configurations

Table 5.29 lists the parameter configurations for the inverse transformation matrix shown in Figure 5.34 as used for the StirMark test *Shearing in Y-Direction* in the experiments of present work.

Configuration	a	b	c	d	e	f
1	1	0	0.05	1	0	0
2	1	0	0.10	1	0	0
3	1	0	0.15	1	0	0
4	1	0	0.20	1	0	0
5	1	0	0.25	1	0	0
6	1	0	0.30	1	0	0

Table 5.29: Configurations of the parameters of inverse transformation matrix stated in Figure 5.34 as applied in the StirMark test *Affine Transformations – Shearing in Y-Direction*

Examples images for configurations 1, 4 and 6 can be found in Figure 5.43.

Results and Discussion

Regarding the results of the various fingerprint matchers in response to the perturbations introduced into fingerprint images of the FVC databases DB1 to DB3, we can make the following general observations:



Figure 5.43: Examples for the *Affine Transformations – Shearing in Y-Direction* test, applied to an image from DB1 (ID 91.2).

Overall the fingerprint matchers are surprisingly robust to the image-manipulations of the *Shearing in Y-Direction* tests. When regarding the exemplary ROC plots per database, in Figures 5.44, 5.45 and 5.44, said behavior becomes very apparent, as within each ROC plot, despite the relatively close zoom – the ranges for FMR and FNMR in the plots are set to $[0:0.35]$ – the individual ROC curves representing the matching performances per test-configurations, lie comparatively close together.

Another point that, as well, can easily be recognized in said ROC plots, but that is also likewise obvious in the listings of equal error rates in Tables 5.30, 5.31 and 5.32, is, that undoubtedly **VeriFinger** is the matcher, whose matching performance exhibits the least negative influences by the image-perturbations caused by the *Shearing in Y-Direction* tests. Not only do the EER corresponding to the various test-configurations show only a very small difference to the respective EER for unperturbed images, but also in the ROC we can observe three major aspects, that indicate VeriFinger’s superior matching performance under current test conditions: For one, we can clearly see, how exceptionally little the ROC curves for the individual test-configurations are separated from each other. The same can then also be said about the points of ZeroFMR. Furthermore, regarding the area of $FMR > EER$, we can note how comparatively faster the FNMR values in the various results for perturbed images trend to the FNMR level related to unperturbed images, occasionally even under-run it.

Almost the opposite of what can be said about VeriFinger’s matching performance, can be said about the performance of the **Phase Only Correlation** matcher. In the *Shearing in Y-Direction* tests for fingerprint images of DB1, DB2 and DB3 alike, POC clearly shows the worst – i.e. the most influenced by the image-perturbations – results of all matchers tested. The actual magnitude of the impairment varies from database to database, but can be as much as three times the amount observed in the other fingerprint matchers.

DB1

As mentioned in the general comments, **VeriFinger** is the matcher, that is the most robust to the perturbations introduced by the *Shearing in Y-Direction* in the fingerprint images of DB1. Regarding the listing of equal error rates in Table 5.30, we find, that even for the most effective test-configuration, number 6, the EER is only 2.44% worse, that the EER for

unperturbed images. Taking a look at Figure 5.44c, we likewise see, that the ROC curves produced by the matching results for test-configurations 1 to 3, lie almost on top of the ROC curve for the original, unperturbed images. Only from configuration 4 onward, the corresponding ROC curves lie in increasing, yet – compared to the other matcher’s plots – small distance.

Comparing the two GrFinger configurations – **GF20** and **GF180** – with each other, we find in the corresponding equal error rates, that from test-configuration 2 onwards, GF180 exhibits slightly more sensitivity to the perturbations in the fingerprint images, than GF20. When additionally comparing the ROC plots, we find another interesting behavior: In the area of (very) low FMR (< 7% about) and accordingly high(er) FNMR (> 13% about), GF180 seems to be more robust than GF20 – not only are the points of ZeroFMR relatively closer together, but in this section, the ROC curves corresponding to configurations 1 and 2 partially even indicate better results than for unperturbed images. A behavior we do not find as articulated in the ROC plot of GF20.

bozorth3 shows a more diverse reaction to the image-manipulations of the *Shearing in Y-Direction* tests: For fingerprint images of test-configurations 1 and 2 the matching results turn out better than those for the original, unperturbed images. A glance at Table 5.30 confirms, that the respective equal error rates are about 1% lower than the one for unperturbed images. For test-configuration 3 then, the matching performance is quite similar to that for the original images, which can also be seen in Figure 5.44a, as the two corresponding ROC curves get to lie almost on top of each other. However, with test-configuration 4 to 6 then, the matching performance decreases comparatively fast, so that for level 6 bozorth3’s results are the second most impaired ones. The only matcher showing worse results then, is – as already mentioned in the general comments – the **Phase Only Correlation** matcher.

As for **FingerCode**’s matching performance in comparison to that of the other matchers, FingerCode turns out to be surprisingly robust to the perturbations introduced in fingerprint images of DB1. Regarding for example the listing of equal error rates in Table 5.30, we find, that, except in test-configuration 03, the EER achieved by the FingerCode matcher for the individual perturbation levels are permanently less distant from the respective EER for unperturbed images, than is the case with the EER of GF20 and hence also GF180. For test-configurations 5 and 6, FingerCode’s matching results are also clearly better, than those of bozorth3. A behavior, that also becomes very apparent, when regarding the respective ROC plots in Figure 5.44.

Configuration	bozo3 (%)	VF (%)	GF180 (%)	GF20 (%)	FC (%)	POC (%)
unperturbed	14.81	5.87	11.41	13.61	12.54	22.60
1	13.85	5.58	11.24	13.79	12.57	22.64
2	13.88	5.77	12.12	13.92	12.76	24.79
3	14.88	6.01	13.23	14.28	13.30	27.43
4	16.26	6.74	13.64	15.60	14.15	29.90
5	17.96	7.26	14.56	16.16	14.71	37.73
6	21.46	8.31	16.37	17.33	15.82	40.22

Table 5.30: Equal error rates for *Affine Transformations – Shearing in Y-Direction* test conducted on sample image database DB1.

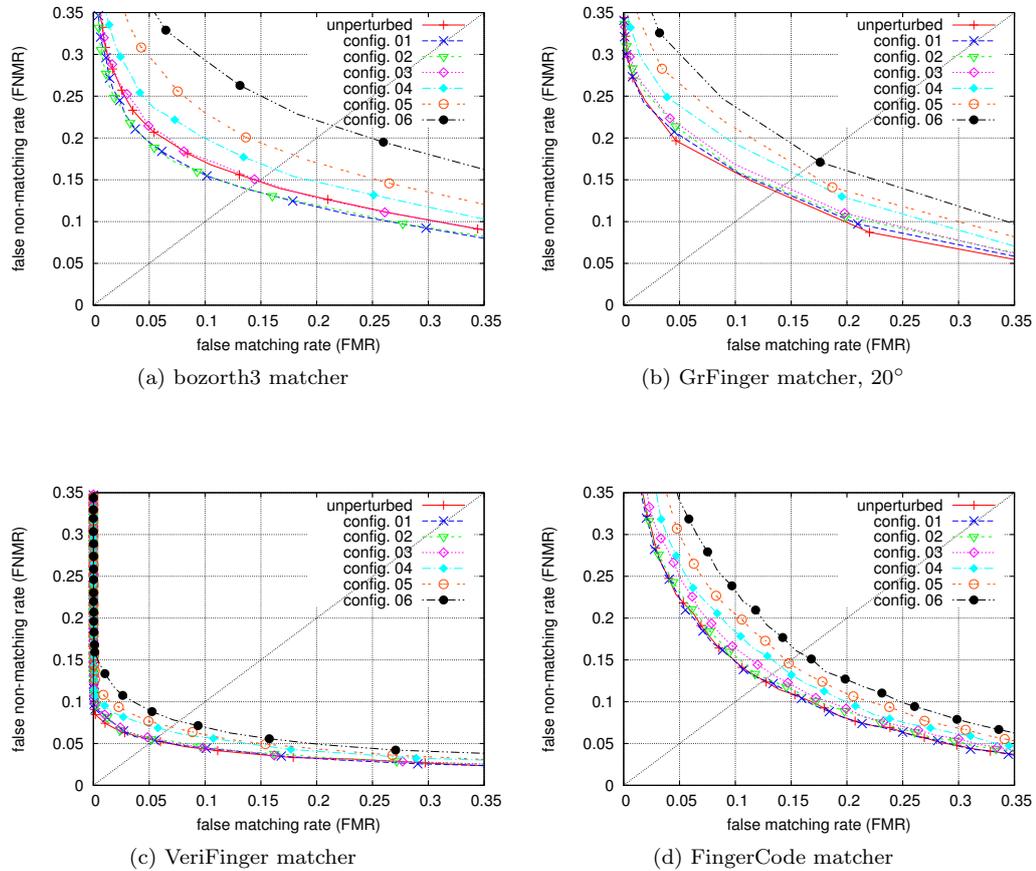


Figure 5.44: Receiver operating characteristics for StirMark test *Affine Transformations - Shearing in Y-Direction* on fingerprint images of DB1.

DB2

A very interesting point in the matching results of **VeriFinger**, **GF20** and **GF180** is, that the relative impairment, caused by the perturbations in the fingerprint images, is remarkably similar between the respective equal error rates related to the *Shearing in Y-Direction* tests for DB1 and those related to the tests for DB2. In other words, regarding the listing of equal error rates for the current tests in Table 5.31: When per matcher – VeriFinger, GF20 and GF180 – calculating the differences between the EER corresponding to the individual test-configurations and the respective EER for unperturbed images, then we will find the same values ($\pm 0.8\%$), when calculating described differences based on the equal error rates for the DB1-related tests (see Table 5.30).

However, that does not necessarily mean, that therefore the overall matching performances of the three matchers for fingerprint images of DB2 is identical to that for images of DB1. For instance, when comparing the respective ROC plots for the **two GrFinger configurations** (as representative example see the ROC plots for GF20 in Figures 5.44b and 5.45a), we will find several small differences: In the area of (very) low FMR ($< 10\%$ about) and accordingly high(er) FNMR ($> 13\%$ about), in the ROC plots for current tests, the ROC

curves related to the different test-configurations are somewhat spread-out farther, than they are in the ROC plots for the DB1-based tests, which indicates a higher influence of the image-perturbations in current results. On the other hand, in the area of high(er) FMR (past the EER) the respective FNMR values of the DB2 test-results trend faster towards the FNMR-level achieved for unperturbed DB2-images, than is the case in the DB1-related ROC plots.

Also the comparison of the ROC plots for **VeriFinger**'s results in the tests of DB1 and DB2 reveals small differences: For example does it become quite obvious in the plot related to DB1, that the points of ZeroFMR in the results for the diverse test-configurations, are relatively farther distanced from the point of ZeroFMR in the results for unperturbed images, than we can observe in the plot related to DB2. Another example for the differences between the two respective ROC plots is, that in the plot related to DB1, in the area of $FMR < 5\%$, the ROC curves generated for the results of test-configurations 1, 2 and 3 lie notably closer to the ROC curve for unperturbed images than in the plot related to DB2.

Inspecting the matching results of **bozorth3** for the *Shearing in Y-Direction* tests on images of DB2, we find that here the perturbations in the fingerprint images have overall a higher impact on bozorth3's matching performance, than they have in images of DB1. In comparison to the other minutiae-based matchers then, we can state, that for test-configurations 1 to 3, the impairment in the results is quite comparative to that observed in the other matchers' results. Yet, as was the case in the tests on DB1 images, the relative matching performance worsens considerably for the more effective test-configurations 4 to 6. For illustration: while the difference between the EER for unperturbed images and the EER related to test-configuration 6 is only 2.26% for VeriFinger and 4.48% for GF180, this difference is 8.07% for bozorth3.

In the previous *Shearing in Y-Direction* tests on images of DB1 the **FingerCode** matcher exhibits a surprisingly high robustness to the perturbations in the fingerprint images. For images of DB2 on the other hand, the negative influence caused by the image-perturbations is clearly noticeable in the respective matching results. This behavior becomes most apparent, when regarding the corresponding ROC plot in Figure 5.45d: As is very obvious, the ROC curves relating to the various test-configurations lie clearly above the ROC curve representing FingerCode's matching performance for the original, unperturbed fingerprint images of DB2. A very interesting observation though is, that for test-configurations 1 to 4 the individual ROC curves for the perturbed images lie comparatively close together, which is especially true for the area of $FMR > 12\%$. Also the ROC curves related to the most effective test-configurations, 5 and 6, gradually trend towards said set of ROC curves for the lesser perturbed images and from a FMR of about $> 27\%$ onwards, even fall together with it. So in this regard one might almost say, that after an initial level of deterioration in the results, introduced already by test-configuration 1, an increase in the tests's effectiveness only brings minor additional impairment.

When comparing FingerCode's matching results in current tests on fingerprint images of DB2, to those of the minutiae-based matchers, we have to state, that here now both GrFinger configurations clearly outperform the FingerCode matcher. However, regarding bozorth3's results, we find, that the FingerCode matcher once again turns out to be comparatively less influenced by perturbations of the two highest intensity levels (test-configurations 5 and 6).

Inspecting the matching performance of the **Phase Only Correlation** matcher in current *Shearing in Y-Direction* tests for fingerprint images of DB2, we find that the perturbations created by test-configurations 1 to 4 lead to a relative aggravation of the matching results,

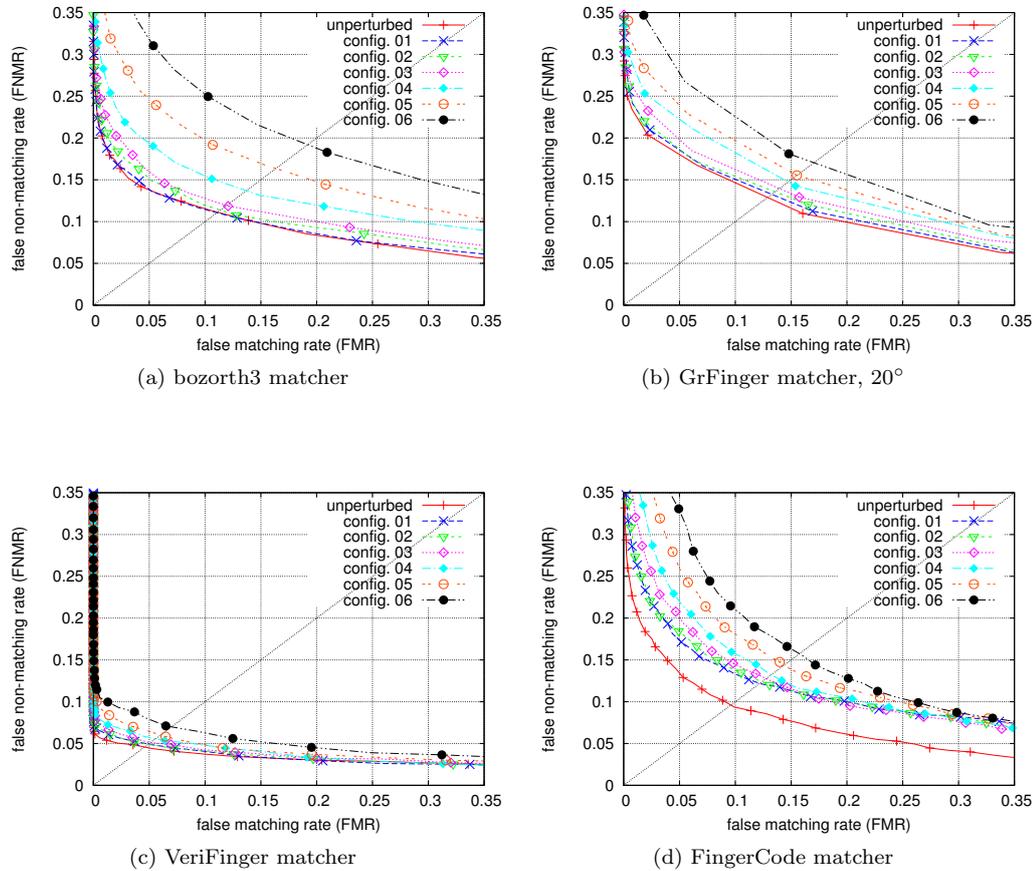


Figure 5.45: Receiver operating characteristics for StirMark test *Affine Transformations - Shearing in Y-Direction* on fingerprint images of DB2.

when compared with POC’s performance in the previous tests on images of DB1. For illustration we can regard the listing of equal error rates in Table 5.31: While in the case of DB1, the EER in the results for test-configuration 1 is quasi equal with the EER generated for unperturbed image, in case of DB2 we already find a difference of 1.54%. This performance gap then widens with increasing level of effectiveness of the test-configuration, so that in the respective EER values for configuration 4 we find a difference to the respective EER value for unperturbed images of “only” 7.31% in case of DB1 images, but already of 11.58% in case of DB2-images. However, comparing the matching performances for the two most-effective test-configurations, 5 and 6, we discover, that the opposite is true: Here for fingerprint images of DB2 the perturbation’s influence is comparatively less than we witness it for images of DB1. Accordingly also the corresponding EER in the results for DB2 are 0.47% and 1.13% less distant from the respective EER for unperturbed images.

DB3

Of all the matchers regarded, **VeriFinger** shows once again the least sensibility to the perturbations introduced by the *Shearing in Y-Direction* tests into the fingerprint images

Configuration	bozo3 (%)	VF (%)	GF180 (%)	GF20 (%)	FC (%)	POC (%)
unperturbed	11.12	5.01	11.72	12.89	9.60	9.69
1	11.07	5.27	12.18	13.26	12.19	11.23
2	11.38	4.98	11.91	13.57	12.39	14.26
3	11.92	5.55	12.77	13.95	12.77	17.84
4	13.70	6.04	13.83	14.74	13.22	21.27
5	16.35	6.24	14.16	15.53	14.39	24.35
6	19.20	7.27	16.56	17.04	15.74	26.19

Table 5.31: Equal error rates for *Affine Transformations – Shearing in Y-Direction* test conducted on sample image database DB2.

of DB3. When further comparing VeriFinger’s matching performance in current tests with that exhibited in the tests for DB1 and DB2 we see, that in the matching results for images of DB3 the negative influence of the image-perturbations is the least apparent. Inspecting VeriFinger’s equal error rates in Table 5.32, we find, that for test-configurations 1 to 3 the corresponding EER even lie slightly below that for the original, unperturbed images. For the most effective test-configuration, number 6, the distance between the corresponding EER and the EER for unperturbed images is just 1.44%.

Like for VeriFinger, also the results of both GrFinger configurations, **GF20** and **GF180**, for the current tests on images of DB3 exhibit clearly less impact by the perturbations in the fingerprint images, than is the case in the previous tests for DB1 and DB2. When comparing the corresponding ROC plots, it becomes very apparent, that in the plots related to current tests, the ROC curves representing the results for the individual test-configurations lie notably closer to the ROC curve for unperturbed images, than they do in the plots related to the tests on DB1 and DB2 images (see also Figure 5.46b in comparison to Figures 5.44b and 5.45b). Also the corresponding equal error rates in Table 5.32 support this observations. Further we can find, that for fingerprint images of DB3, GrFinger produces better matching results for test-configuration 1 than for the original, unperturbed images.

The behavior of **bozorth3** in current tests is somewhat contrary to that of the other minutiae-based matchers: judging by the listing of equal error rates in Table 5.32, the introduction of the perturbations created by the *Shearing in Y-Direction* tests, into fingerprint images of DB3, causes a stronger deterioration in the matching results, than we observe in the tests for fingerprint images of DB1 or DB2. Still, one aspect where bozorth3’s matching performance in current tests is superior to that exhibited in the test for DB1 and DB2, becomes apparent, when regarding the ROC plot in Figure 5.46a: In the area of high(er) FMR (> 20%) the FNMR of all test-results trend clearly faster towards the FNMR levels achieved for the original, unperturbed images, than we can observe it in the corresponding plot of the previous tests (see Figures 5.44a and 5.45a).

As was the case in the tests for fingerprint images of DB2, the matching results of the non-minutiae-based matchers for the current tests on images of DB3 generally exhibit a stronger influence of the *Shearing in Y-Direction* perturbations, than the results of the minutiae-based matchers. However, and also in line with the previous observations for images of databases DB1 and DB2, the **FingerCode** matcher once more shows a higher robustness to the perturbations introduced by the two most effective test-configurations, 5 and 6, than does

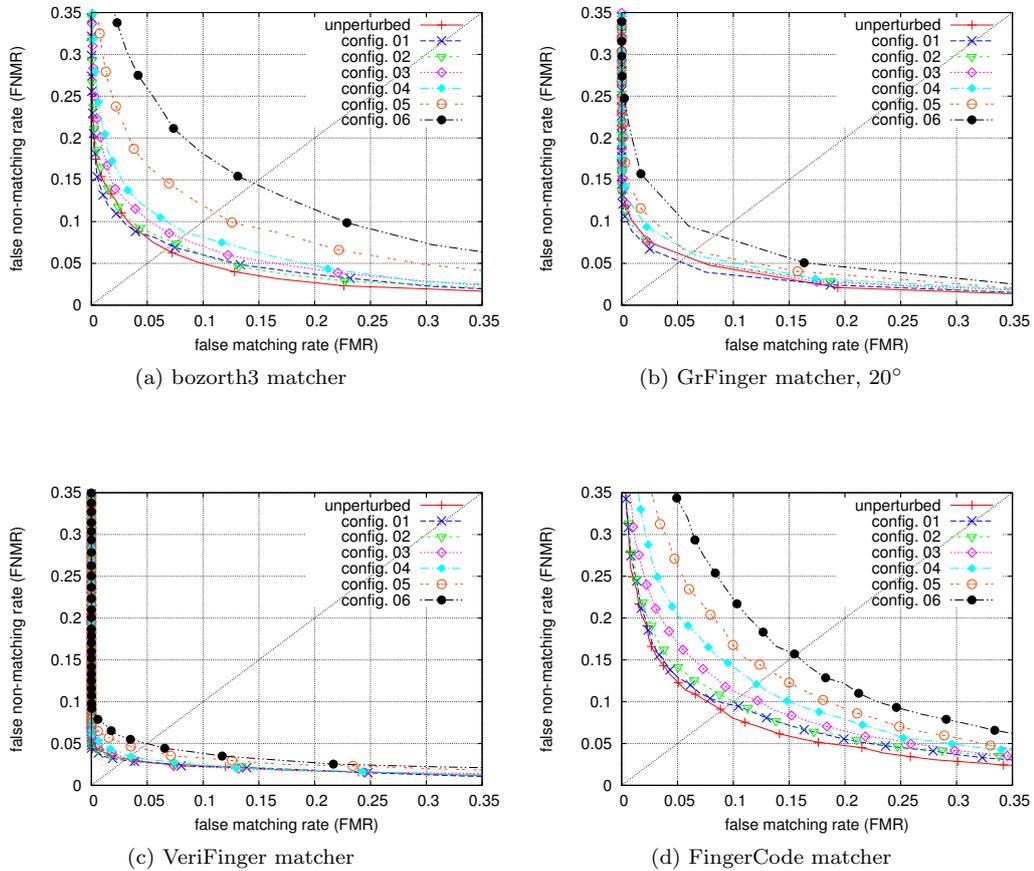


Figure 5.46: Receiver operating characteristics for StirMark test *Affine Transformations - Shearing in Y-Direction* on fingerprint images of DB3.

the bozorth3 matcher. Not only are the corresponding EER in FingerCode’s matching results less different from the respective EER related to the unperturbed fingerprint images (please refer to Table 5.32 for details), but also a glance at the ROC plots in Figure 5.46 reveals, that FingerCode’s ROC curves, displaying the matching performance for test-configurations 5 and 6, trend faster towards the FNMR levels achieved for unperturbed images, than do the corresponding ROC curves of bozorth3.

The **Phase Only Correlation** matcher, on the other hand, is once more the matcher with the highest sensitivity to the perturbations, with its results for DB3-images displaying an even stronger impairment, than is the case for images of DB1 or DB2.

Configuration	bozo3 (%)	VF (%)	GF180 (%)	GF20 (%)	FC (%)	POC (%)
unperturbed	6.68	3.60	6.90	5.82	8.98	15.07
1	7.13	3.27	6.20	5.22	9.67	16.30
2	7.47	3.42	7.09	5.80	10.03	20.30
3	8.03	3.37	7.06	5.80	10.80	25.21
4	8.62	4.05	7.88	6.27	12.09	30.02
5	11.07	4.85	8.26	6.46	13.51	35.73
6	14.56	5.04	9.64	8.43	15.59	37.97

Table 5.32: Equal error rates for *Affine Transformations – Shearing in Y-Direction* test conducted on sample image database DB3.

5.13 Affine Transformations

Shearing in X- and Y-Direction (affineXYshear)

Parameter Configurations

Table 5.33 lists the parameter configurations for the inverse transformation matrix shown in Figure 5.34 as used for the StirMark test *Shearing in X- and Y-Direction* in the experiments of present work.

Configuration	a	b	c	d	e	f
1	1	0.05	0.05	1	0	0
2	1	0.10	0.10	1	0	0
3	1	0.15	0.15	1	0	0
4	1	0.20	0.20	1	0	0
5	1	0.25	0.25	1	0	0
6	1	0.30	0.30	1	0	0

Table 5.33: Configurations of the parameters of inverse transformation matrix stated in Figure 5.34 as applied in the StirMark test *Affine Transformations – Shearing in X- and Y-Direction*

Examples images for configurations 1, 4 and 6 can be found in Figure 5.47.

Results and Discussion

As expected, the combination of the two image-manipulations *Shearing in X-Direction* and *Shearing in Y-Direction* leads to a clearly stronger deteriorating effect on the matching performances of the individual fingerprint matchers regarded, than the application of, for example, just the *Shearing in Y-Direction* manipulation alone, as in the previously discussed StirMark-test.

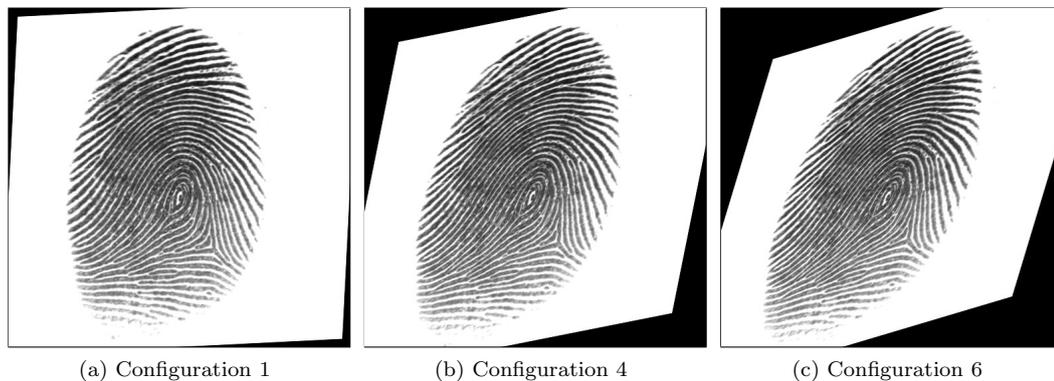


Figure 5.47: Examples for the *Affine Transformations – Shearing in X- and Y-Direction* test, applied to an image from DB1 (ID 91.2).

DB1

Regarding the matching performance of matcher **bozorth3** in response to the perturbations of tests *Shearing in X- and Y-Direction* in fingerprint images of DB1, we can make the following statements: For test-configuration 1, the StirMark test does not have any noteworthy influence on bozorth3's matching results: In the ROC plot in Figure 5.48a, we find, that the ROC curve corresponding to test-configuration 1 lies almost on top of the ROC curve representing the matching results for the original, unperturbed images of DB1 – in the area around the equal error rate, even slightly below. For test-configuration 2, the negative effect of the image-perturbations on bozorth3's performance is already clearly detectable, as not only the corresponding ROC curve in the ROC plot lies decidedly above that for unperturbed image, but also as the corresponding equal error rate rises from 14.56% to 17.44%. From test-configuration 3 onward then, the test results of bozorth3 show a stronger impairment, than the results of any other regarded fingerprint matcher but the Phase Only Correlation matcher. Inspecting the listing of equal error rates in Table 5.34, we can find for example, that for test-configuration 3 bozorth3's EER is already 7.56% higher than it's EER for unperturbed images, while for the second most impaired minutiae-based fingerprint matcher, GF180, this difference is only 5.75%.

When comparing both GrFinger configurations, **GF20** and **GF180**, in regard to the amount of influence the perturbations, introduced by the *Shearing in X- and Y-Direction* tests into the fingerprint images of DB1, have on the individual matching performances, we find, that throughout all test-configurations, GF20 clearly outperforms GF180. This behavior becomes most apparent, when comparing the corresponding ROC plots, as in case of GF20 all the ROC curves related to the various test-configurations lie clearly closer to the ROC curve for unperturbed images than the do in the plot for GF180. The listing of equal error rates in Table 5.34 likewise supports the observation: Already for test-configuration 1 the EER for GF180 is 0.63% more different from the respective EER for unperturbed images, than is the case for GF20. And this gap in performance even grows with increasing effectiveness of the test-configuration, so that finally, for configuration 6, the relative impairment in the EER for GF180 is 8.16% higher than that in the EER for GF20.

The **VeriFinger** matcher demonstrates a somewhat more diverse reaction to the perturbations in fingerprint images of DB1: Not only in the ROC plot in Figure 5.48c, but also when

regarding the respective equal error rates in Table 5.34 we find, that for test-configurations 2 to 4, of all the matchers regarded, VeriFinger exhibits the least impairment in its matching results. For test-configurations 5 and 6 though, VeriFinger’s matching performance deteriorates considerable, so that in terms of robustness to the image-perturbations, it falls behind both GrFinger Configurations and FingerCode. For illustration let us regard the listing of equal error rates and compare the differences between the EER corresponding to specific test-configurations and the respective EER for unperturbed images: Looking at test-configuration 4, VeriFinger’s EER here is 5.69% different from its EER for unperturbed images. For GF20 this difference is 5.82%, for FingerCode 7.44% and for GF180 it is even 10.41%. Now turning to test-configuration 6, the situation changes as described above and accordingly said difference values get 23.19% for VeriFinger, while only 14.54% for GF20, 16.43% for FingerCode and 22.71% for GF180.

As already indicated in the previous paragraphs, the **FingerCode** matcher turns out to be surprisingly robust to the perturbations introduced into the fingerprint images of DB1 by the *Shearing in X- and Y-Direction* tests. Based on the equal error rates in Table 5.34, and backed up by a comparison of the corresponding ROC plots (please refer to Figure 5.48), we can determine, that the only fingerprint matcher that exhibits less influence in its results, over the whole range of test-configurations, is GF20. On the other hand, as stated before, bozorth3’s and VeriFinger’s matching performances appear only in the first two, respectively first four levels of perturbation intensity less impaired, than FingerCode’s performance. The two remaining fingerprint matchers, GF180 and Phase Only Correlation, then produce clearly stronger influenced matching results in all regarded test-configurations.

The **Phase Only Correlation** matcher is decidedly the one matcher, exhibiting the highest sensitivity to the *Shearing in X- and Y-Direction* perturbations introduced in the fingerprint images of DB1. The related ROC plot shows, that the ROC curves, corresponding to the individual test-configurations, are undoubtedly farther distant from the ROC curve for unperturbed images, than is the case in the ROC plot of any of the other matchers. Accordingly also the equal error rates in Table 5.34 show, that the relative distances between the EER achieved for the original, unperturbed images and the EER related to the individual test-configurations are constantly the higher than those in the results of the other matchers.

Configuration	bozo3 (%)	VF (%)	GF180 (%)	GF20 (%)	FC (%)	POC (%)
unperturbed	14.81	5.87	11.41	13.61	12.54	22.60
1	14.56	6.15	12.13	13.71	13.22	24.92
2	17.44	6.57	14.24	15.17	14.34	29.98
3	22.38	8.94	17.15	16.76	16.52	34.37
4	30.14	11.57	21.82	19.42	19.97	45.31
5	37.21	19.44	28.46	24.30	24.37	49.96
6	40.71	29.06	34.11	28.15	28.97	52.82

Table 5.34: Equal error rates for *Affine Transformations – Shearing in X- and Y-Direction* test conducted on sample image database DB1.

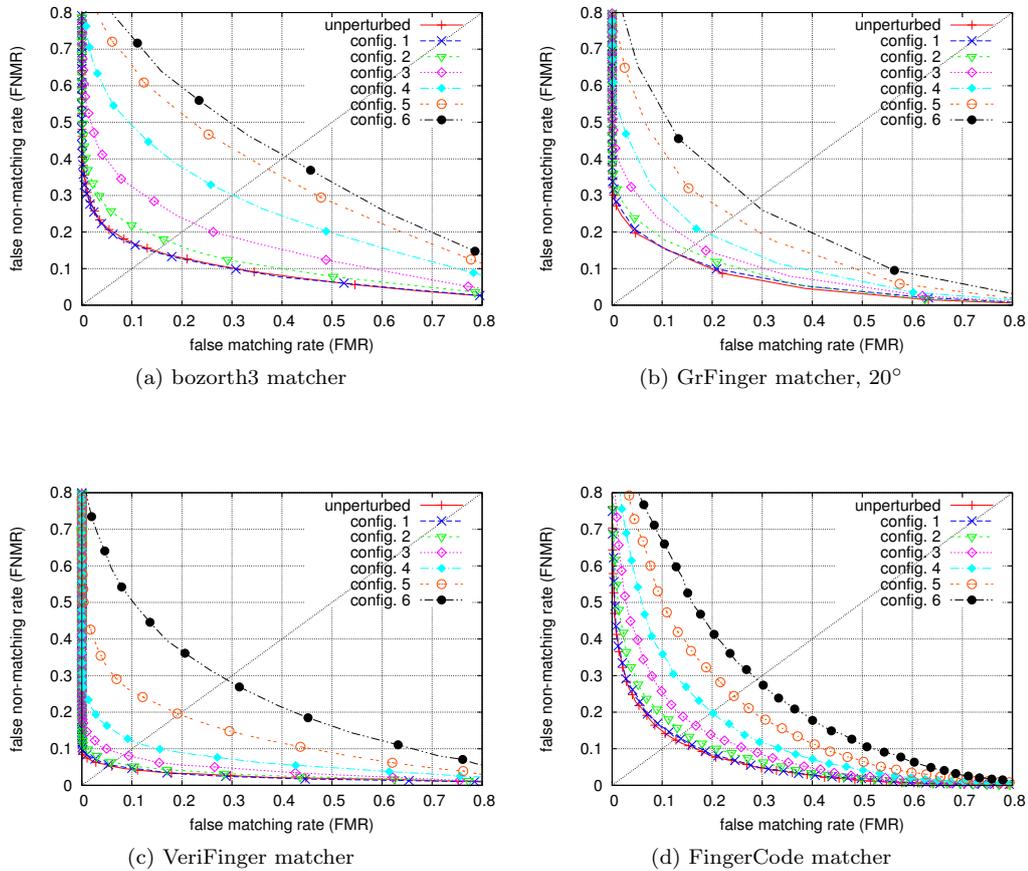


Figure 5.48: Receiver operating characteristics for StirMark test *Affine Transformations – Shearing in X- and Y-Direction* on fingerprint images of DB1.

DB2

The reaction of **bozorth3**, when the perturbations created by the *Shearing in X- and Y-Direction* tests are introduced into fingerprint images of DB2, resembles largely that, exhibited by bozorth3 in the tests for images of DB1: For test-configuration 1, the influence of the perturbations is barely noticeable in the matching results and also the calculated equal error rate is quasi similar to that established in the results for unperturbed images of DB2. However, for test-configurations 2 to 6, bozorth3’s matching performance is decidedly affected by the image-perturbations. An inspection of the equal error rates listed in Table 5.35 reveals, that for test-configurations 2 to 6 bozorth3 is subject to the strongest impairment among all minutiae-based matcher and from test-configuration 3 on, is also outperformed by the FingerCode matcher. A comparison of the ROC plots in Figure 5.49 supports this observation.

Regarding both GrFinger configurations, **GF20** and **GF180**, and comparing their respective matching results for current tests with those, generated during the previous set of tests on fingerprint images of DB1, we can state, that GF20 generally shows a somewhat higher sensitivity to the perturbations in images of DB2, while in contrast GF180 appears less

affected, than it was in the tests for DB1. Still, when regarding the listing of equal error rates in Table 5.35, we can see clearly, that also in current tests GF20 clearly outperforms GF180. For example the difference between the EER in the matching results for test-configuration 1 and the EER for the unperturbed images is 1.12% for GF180 and only 0.90% for GF20. Per more-effective test configuration again also the manifestations of the negative influences increase faster in GF180's matching results than in GF20's results. For configuration 6 finally, we find, that GF180's EER is 19.84% higher than its EER for unperturbed images, while for GF20 this difference is only 15.53%.

The **VeriFinger** matcher behaves in the tests for fingerprint images of DB2 very much like it did in the previous tests for images of DB1. In the results for the test-configurations 4 to 6 though, we see a certain improvement – i.e. a higher robustness to the introduced perturbations – compared to the results for DB1-images. Due to this improvement, when comparing the matching performances of the individual fingerprint matchers for current *Shearing in X- and Y-Direction* tests on images of DB2, VeriFinger now exhibits the least impaired results of all matchers for the test-configurations 2 to 5 (while for images of DB1 this was “only” the case for configurations 2 to 4). Moving from configuration 5 to configuration 6 though, VeriFinger's matching performance, once again, degrades essentially – the corresponding equal error rate dropping from 11.30% to 21.66% – revealing a higher sensitivity to the image-perturbations in this test-configuration, than both GrFinger configurations or the FingerCode matcher.

When regarding the listing of equal error rates in Table 5.35 and comparing **FingerCode**'s overall matching performance, in response to the perturbations introduced into the fingerprint images of DB2, to that of the other fingerprint matchers tested, we find already for test-configuration 1, a degradation of 2.78%, which, at this level of test effectiveness, is only surpassed by a degradation of 4.43% in case of the Phase Only Correlation matcher. Also at the next level of perturbation intensity the results of the FingerCode matcher are still more impaired than those of the minutiae-based fingerprint matchers. However, from test-configuration 3 on, the bozorth3 matcher turns out to be clearly more influenced by the perturbations in the fingerprint images, than the FingerCode matcher. Further for the most effective test-level, configuration 6, as said before, FingerCode then even manages to outperform the VeriFinger matcher: Regarding the listing of equal error rates again and calculating the differences between the EER in the respective results for test-configuration 6 and the corresponding EER achieved for the original, unperturbed images, we witness an aggravation of 20.33% for FingerCode, of 22.09% for VeriFinger and even 26.80% for bozorth3.

Also in the *Shearing in X- and Y-Direction* tests on images of DB2 the **Phase Only Correlation** matcher displays overall the strongest impaired matching results of all fingerprint matchers regarded in the experiments. However, when comparing the matching performances for the various perturbation intensities in present tests for images of DB2 with those exhibited in the previous tests on fingerprint images of DB1, we can make an interesting observation: Regarding the particular listings of equal error rates in Tables 5.34 (DB1) and 5.35 (DB2), we find, that for the first three test-configurations, 1 to 3, in present tests on DB2-images the matching performance deteriorates clearly faster than it does for images of DB1. Yet for the higher effective test-configurations, 4 to 6, the opposite is true, as here the perturbations, introduced in images of DB1, clearly have a stronger negative influence on the matching results, than when they are introduced in the fingerprint images of DB2.

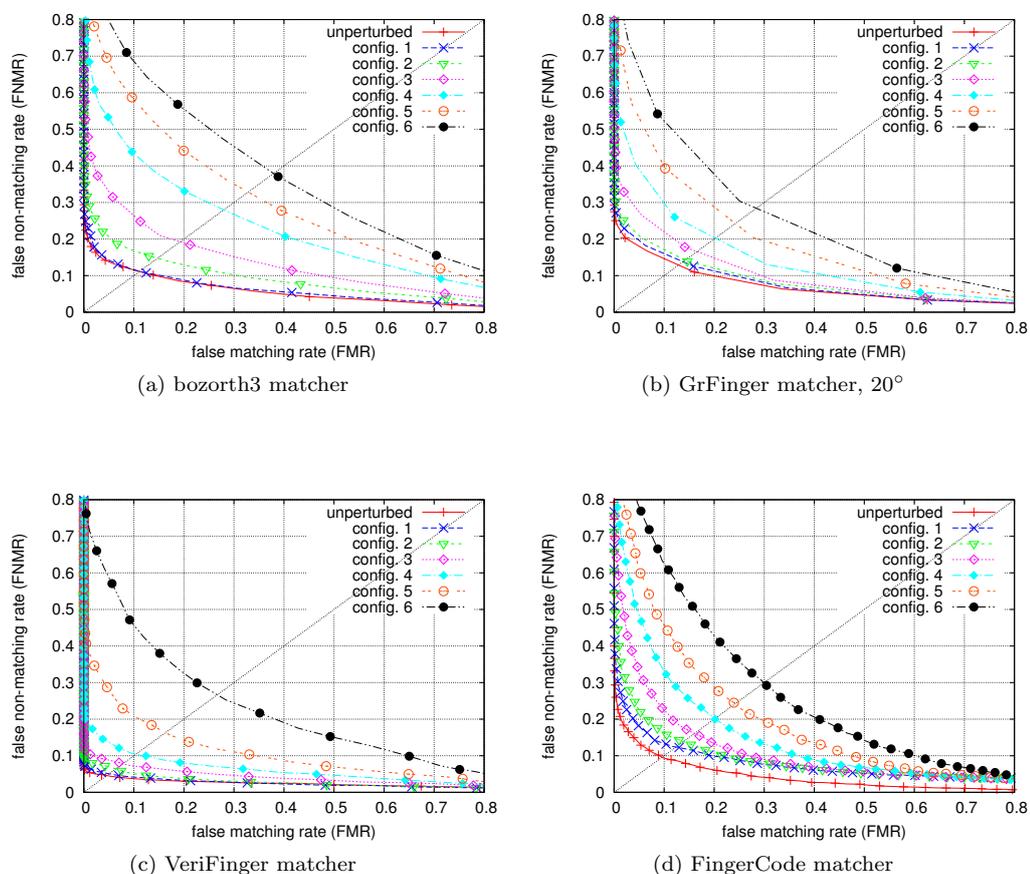


Figure 5.49: Receiver operating characteristics for StirMark test *Affine Transformations – Shearing in X- and Y-Direction* on fingerprint images of DB2.

DB3

Regarding the results for the *Shearing in X- and Y-Direction* tests on fingerprint images of DB3 and comparing the individual matching performances of the examined fingerprint

Configuration	bozo3 (%)	VF (%)	GF180 (%)	GF20 (%)	FC (%)	POC (%)
unperturbed	11.12	5.01	11.72	12.89	9.60	9.69
1	11.18	5.29	12.85	13.79	12.37	14.12
2	14.75	6.53	14.09	14.22	13.85	21.40
3	19.26	8.10	16.85	16.58	16.06	28.08
4	27.97	10.45	21.57	20.25	20.10	30.72
5	32.85	16.31	26.00	24.24	24.20	34.03
6	37.92	26.67	31.56	28.42	29.93	36.96

Table 5.35: Equal error rates for *Affine Transformations – Shearing in X- and Y-Direction* test conducted on sample image database DB2.

matchers with each other, we find a general behavior, that is largely comparable to the one observed in the previous tests for images of DB2:

The GrFinger configuration **GF20** exhibits notably less influence by the perturbations in the fingerprint images, than the configuration **GF180**. It is worth noting though, that for test-configuration 1, GF20 and GF180 both manage to produce matching results that are (very) slightly better than those generated for unperturbed images and here GF180 is able to reduce the EER by 0.41%, while GF20 “only” by 0.26%. Overall, comparing GrFinger’s matching performance relating to fingerprint images of DB3 to that presented so far, for images of DB1 and DB2, GF180 clearly produces the least impaired results for DB3, while for GF20 a likewise clear statement can not be made (For further details please refer to the respective equal error rates in Tables 5.34, 5.35 and 5.36).

Of all fingerprint matchers regarded in current tests, **VeriFinger**’s matching performance is, once again, the most robust to the negative influences by the perturbations in the fingerprint images of DB3 – apart from two exceptions: For one, for test-configuration 2 GF20 manages to produce mildly less impaired results. But second and more significant, for test-configuration 6, the results are impaired as such, that here both GrFinger configurations, GF20 and GF180, exhibit a better performance at this level of test-effectiveness. However, contrary to the observations in the previous tests on images of DB1 and DB2, the FingerCode matcher no longer manages, to outperform the VeriFinger matcher at this highest level of perturbation intensity. The described behavior becomes not only apparent, when inspecting the corresponding equal error rates in Table 5.36, but is especially obvious when comparing the related ROC plots in Figures 5.50b, 5.50c and 5.50d.

Among the set of minutiae-based fingerprint matchers, **bozorth3** is, the most sensitive to the perturbations in fingerprint images of DB3, which – and this is a difference to bozorth3’s behavior for the perturbed images of DB2 – is also true for the results for test-configuration 1. When further comparing bozorth3’s matching performance in regard to all three image databases DB1 to DB3, we can state that for images of DB1 the perturbations generated by the *Shearing in X- and Y-Direction* tests have the least impact on the results and for images of DB3 the strongest. An observation that can be verified in the corresponding equal error rates and the related ROC plots alike.

As was the case in the previous tests for fingerprint images of DB2, also here, in current test for images of DB3, **FingerCode**’s matching results are strongly impaired by the perturbations introduced in the fingerprint images. For test-configurations 1 and 2 the extend of the negative influence on the matching results of FingerCode is larger than witnessed in the results of any minutiae-based fingerprint matcher. From test-configuration 3 on thought, bozorth3 turns out to be more influenced by the perturbations than the FingerCode matcher – a behavior that basically falls in line with the observations made in the results for the previous tests on the fingerprint image databases DB1 and DB2. For illustration we can once again consider the differences between the respective EER achieved for the original, unperturbed images and the EER in the results for the individual test-configurations: Re-grading for example configuration 3, we find, that for FingerCode said difference is 7.13%, while for bozorth3 it amounts already to 8.19%. At the highest perturbation level, in test-configuration 6, the discrepancy in the individual susceptibility to the influences of the perturbations becomes even more noticeable, as the differences then amount to 16.43% for FingerCode and 25.90% for bozorth3.

The **Phase Only Correlation** matcher is once more the matcher displaying clearly the strongest impaired results in all *Shearing in X- and Y-Direction* tests on images of DB3.

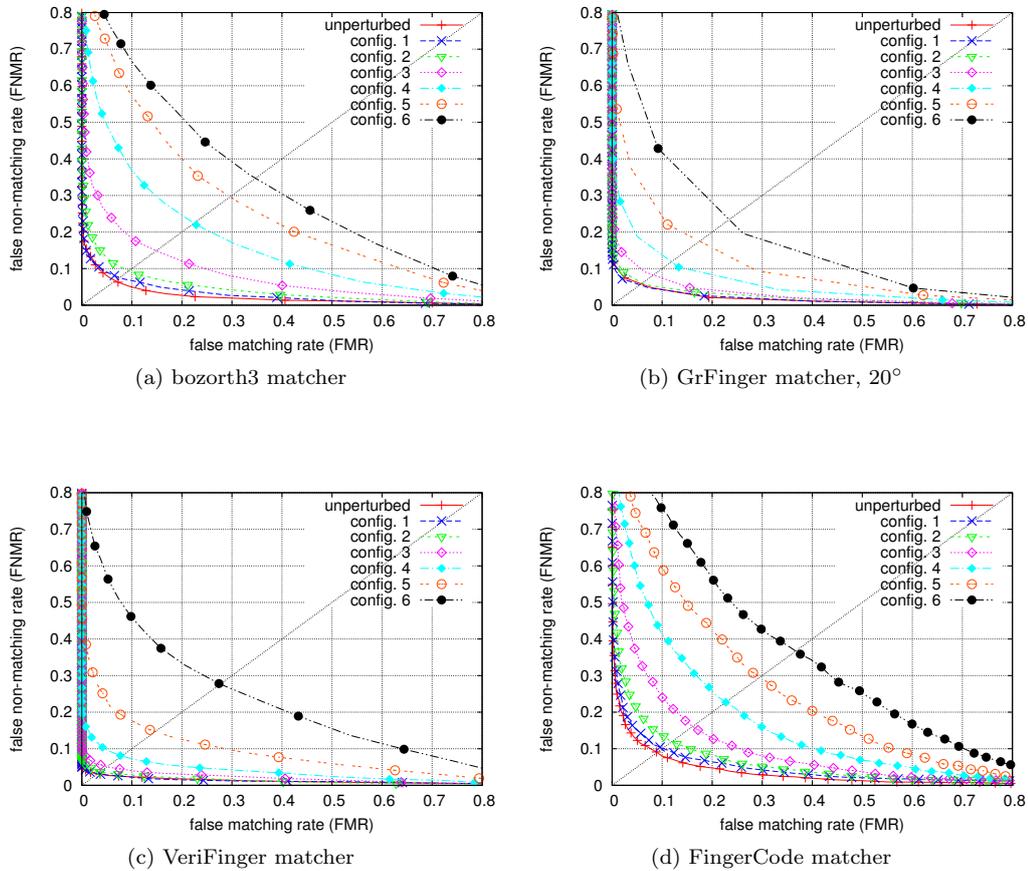


Figure 5.50: Receiver operating characteristics for StirMark test *Affine Transformations – Shearing in X- and Y-Direction* on fingerprint images of DB3.

Regarding the listing of equal error rates in Table 5.36, we can establish, that already for test-configuration 1 POC's EER shows a difference to the EER for unperturbed images, that most other matchers do not even reach at test-configuration 3. Comparing POC's matching performance in the previous tests for fingerprint images of databases DB1 and DB2 to that, exhibited in current tests then, we can state, that the results for the tests on DB3 are stronger impaired by the image-perturbations, than the results of both previous tests for fingerprint images of DB1 and DB2.

5.14 Small Random Distortions (rnddist)

This constitutes *the* StirMark test. It is a combination of the aforementioned basic image manipulations and aims "*in its simplest version*" [27] to simulate a resampling process. In other words it generates perturbations, that typically appear when printing an image and thereafter scanning it again.

The image manipulation happens in the following four consecutive processing steps (more

Configuration	bozo3 (%)	VF (%)	GF180 (%)	GF20 (%)	FC (%)	POC (%)
unperturbed	6.68	3.60	6.90	5.82	8.98	15.07
1	7.56	3.48	6.49	5.56	10.14	20.83
2	9.41	4.01	7.57	6.03	11.96	29.82
3	14.87	5.10	10.25	8.25	16.11	35.88
4	22.39	8.24	14.88	11.86	22.72	43.52
5	29.65	14.74	19.95	17.50	29.65	45.85
6	34.71	28.06	27.61	23.41	36.70	48.96

Table 5.36: Equal error rates for *Affine Transformations – Shearing in X- and Y-Direction* test conducted on sample image database DB3.

detailed information can be found in [27], here I just present an excerpt to facilitate understanding of the influence the StirMark test has on fingerprint images):

1. *Minor geometric distortions* – The images are (slightly) stretched, sheared, shifted and/or rotated by a random amount, followed by a resampling applying bi-linear or Nyquist interpolation. The distortion itself is determined by the function: $M = \alpha[\beta A + (1 - \beta)D] + (1 - \alpha)[\beta B + (1 - \beta)C]$, where A, B, C, D are the corners of the image, M is a point, whose location is to be distorted and $0 \leq \alpha, \beta \leq 1$ are the coordinates of M, relative to the corners.
2. *Transfer function* – A not otherwise specified transfer function, aimed to emulate analog/digital converter imperfections, introduces a small and smoothly distributed error into each image value.
3. *Global “Bending”* – A (slight) deviation is applied to each pixel, which is greatest at the center of the image and decreasing towards the borders.
4. *Higher Frequency Displacement* – Additional displacement of the image values is performed, based on the function $\lambda \sin(\omega_x x) \sin(\omega_y y) + n(x, y)$, with n being a random number.
5. *JPEG compression* – As final step a medium JPEG compression is applied to the image.

Relation to Fingerprints

In its character of being a combination of several different image distortions, by applying *the* StirMark test on fingerprint images, I aim to simulate an interaction of various naturally occurring image perturbations: Foremost a random warping of the ridge lines, that in real life would for example be caused by unevenly distributed pressure exercised on the contact area during acquisition, or if this contact area were to be uneven by itself. Also inaccuracies, errors introduced by the fingerprint scanner can be a source for this type of deformation (two corresponding examples are shown in Figure 5.51).



Figure 5.51: Two examples for warping effects in fingerprint images. In this case introduced by the fingerprint scanner.

As is stated in above listing of the StirMark internal processing steps, said image-warping will be performed both on a global, as well as on a very local level, adding even more to the “natural”, “coincidental” character of the output fingerprint images.

Parameter Configurations

The intensity of *the* StirMark perturbations can be adjusted via a single parameter. As the corresponding source code reveals, “*the attack parameter is simply a factor that multiplies the default parameters*”. In the experiments of present work, we will be applying the *Small Random Distortions* test with following set of parameter values: {0.6, 1.0, 1.4, 1.8, 2.2, 2.6, 3.0, 3.4, 3.8, 4.2} and the limited number of tests, run on the non-minutiae-based fingerprint matchers, cover the set {0.6, 1.0, 1.8, 2.6, 3.4, 4.2}. Figure 5.52 shows examples generated for parameter values 0.6, 2.6 and 4.2.

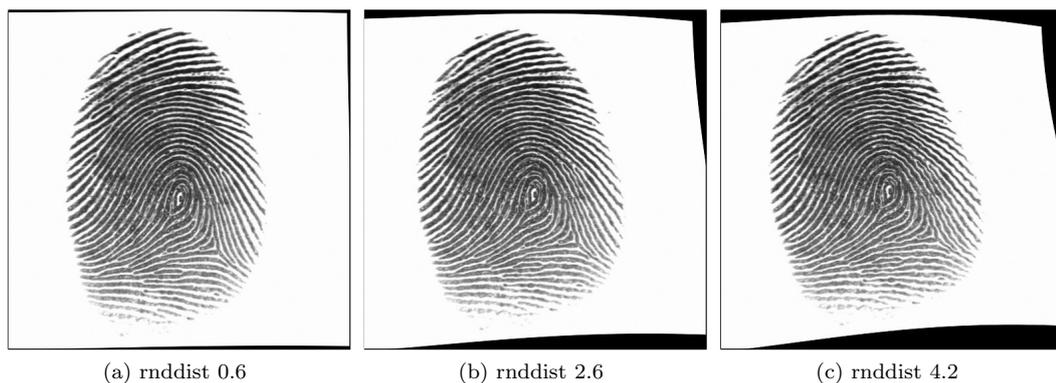


Figure 5.52: Examples for the *Small Random Distortions* test, applied to an image from DB1 (ID 91.2).

For values < 0.6 the influence of the attack is barely noticeable. On the other hand, when the parameter is set to values > 4.2 , the distortions reach a level, that will hardly be found in any real-life fingerprint applications. One effect in particular that causes a decidedly “unnatural”

appearance in the fingerprint images, is, that for high parameter values the ridge lines adopt a periodic, wave-shaped structure. This behavior seems to be a manifestation of the employed sine-functions in the processing steps “Global Bending” and “Higher Frequency Displacement”. Said wave-structure can partially already be witnessed for parameter value 4.2 – see Figure 5.52c for a corresponding example.

Results and Discussion

Overall, when regarding the influences of the StirMark tests, *Small Random Distortions*, on the matching performances of the individual fingerprint matchers, we can establish, that for all three databases, DB1 to DB3, the perturbations, which are introduced into the fingerprint images, have only a very limited negative impact on the respective matching results. In fact, for some matchers certain perturbation-levels even lead to an improved matching performance, in comparison to the results achieved for the original, unperturbed fingerprint images.

DB1

Of all the fingerprint matchers regarded, **VeriFinger** is clearly the one matcher, whose results are the least influenced by the perturbations, introduced into fingerprint images of DB1 in course of current *Small Random Distortions* tests. Inspecting the listing of equal error rates in Table 5.37, we find, that in VeriFinger’s results, the maximum difference between an EER related to a test on perturbed images and the EER for unperturbed images, is only 1.33%. Further we can also note, that for test-levels 0.6 to 1.8 the matching results turn out to be better, than those generated for unperturbed images. When regarding the according ROC plot in Figure 5.53c, we can observe this behavior in particular in the area of FMR greater than the respective EER.

The response of the **bozorth3** matcher to the perturbations, introduced in the fingerprint images of DB1, is in so far very interesting, as the matching results for the first six test-levels, rnddist 0.6 to 2.6, are considerably better, than the matching results for unperturbed images. Regarding Table 5.37, we can establish, that the EER in the results for rnddist 0.6 to 2.6 are up to 1.15% lower than the EER relating to the original fingerprint images. Accordingly, when referring to the ROC plot in Figure 5.53a, we find the ROC curves corresponding to the first six test-levels to lie notably below the ROC curve portraying bozorth3’s matching performance for unperturbed images of DB1.

Comparing the matching performances of both employed GrFinger configurations, **GF20** and **GF180**, in regard to their individual sensitivity to the image manipulations by current *Small Random Distortions* tests, we can state, that in all cases but rnddist 0.6, the matching results of GF20 are clearly less influenced, than those of GF180. This becomes for example very apparent, when examining the respective equal error rates in Table 5.37 and calculating per GrFinger matcher, how far the EER related to the individual test-levels are separated from the EER for unperturbed images. For GF180 this difference value is in average 1.05% higher than for GF20.

When evaluating GF20’s and GF180’s matching performance relative to the other two minutiae-based fingerprint matchers, it becomes obvious, that overall GF180 is the minutiae-based matcher exhibiting the highest sensitivity to the perturbations in fingerprint images of DB1. As for GF20, it depends on the individual level of test-effectiveness, if the results

of GF20 are more impaired than those of bozorth3 or VeriFinger – no simple, clear verdict can be found here (for details, please refer to the listing of equal error rates in Table 5.37).

Turning to the non-minutiae-based fingerprint matchers, we find for example, that the **FingerCode** matcher exhibits a rather interesting reaction to the perturbations introduced into the fingerprint images of DB1: Already with the first and least-effective test, rnddist 0.6, the EER rises from originally 12.54% for the unperturbed images to 13.60% (a difference of 1.06%). However, for the following three tests-levels of the reduced set of tests performed on the non-minutiae-based matchers, rnddist 1.0, 1.8 and 2.6, the increased extend of the image-perturbations does not seem to have an accordingly larger effect on FingerCode's matching performance, as the corresponding EER values stay relatively constant: 13.58%, 13.67% and 13.79% respectively. Only for the two higher effective test-levels, rnddist 3.4 and 4.2, then, the matching performance gradually decrease again. The ROC plot in Figure 5.53d supports this observation: We can see, how – especially for $FMR > 10\%$ about – the ROC curves, corresponding to the four least effective levels of the *Small Random Distortions* tests conducted on the FingerCode matcher, lie very close to each other and in a certain, almost fixed distance to the ROC curve representing the matching performance for unperturbed images of DB1. The ROC curves corresponding to tests rnddist 3.4 and 4.2 though, each lie clearly separated from the other curves.

When comparing the matching results of the FingerCode matcher in current tests, to the results of the minutiae-based fingerprint matchers regarded, we can generally state, that the FingerCode matcher shows clearly a higher sensitivity to the *Small Random Distortions* perturbations in the fingerprint images. The only two exceptions are, that in the tests rnddist 2.6 and rnddist 4.2 FingerCode manages, to appear more robust than GF180. For illustration we can refer to the listing of equal error rates in Table 5.37: When calculating the differences between the EER in the results for the individual test-levels and the corresponding EER for the original, unperturbed images, we find, that in FingerCode's results for tests rnddist 2.6 and rnddist 4.2 said differences are lower by 0.46% and 0.75% respectively, than those in GF180's results.

Also the **Phase Only Correlation** matcher displays a somewhat unexpected matching performance in the *Small Random Distortions* tests of the reduced test-set. For tests rnddist 0.6 and 1.0, the generated matching results are notably better than the results POC originally achieves for the unperturbed images of DB1: The equal error rates relating to said tests are 1.03% and 1.14% lower than the EER for the unperturbed images (see Table 5.37 for details). Accordingly also the POC matcher's ROC plot shows, that the ROC curves corresponding to rnddist 0.6 and 1.0 lie clearly below the ROC curve for the original database images.

For the tests rnddist 1.8 and 2.6 then, the equal error rates exhibit a certain deterioration in the matching results – almost neglectable for rnddist 1.8, already larger for rnddist 2.6 (the difference between the EER related to test rnddist 2.6 and the EER for unperturbed images amounts to 1.28%). Yet an interesting behavior of the Phase Only Correlation matcher in regard to these medium-effective test configurations is revealed, when inspecting the ROC plot again: In the area of $FMR < 23\%$ the ROC curves representing the matching results for rnddist 1.8 and 2.6 lie clearly above the ROC curve which indicates the POC's matching performance for unperturbed DB1-images. However, at a FMR of about 23%, the ROC curve for rnddist 1.8 falls below the curve for the unperturbed images and trends further downward, until a FMR of 39% about, from which point on the ROC curve for rnddist 1.8 nearly falls into line with the curves for the two less effective test-configurations, rnddist 0.6 and 1.0. In addition, at quasi the same level of FMR, also the ROC curve for rnddist 2.6 falls below that for unperturbed images and stays there, until we finally reach the point of

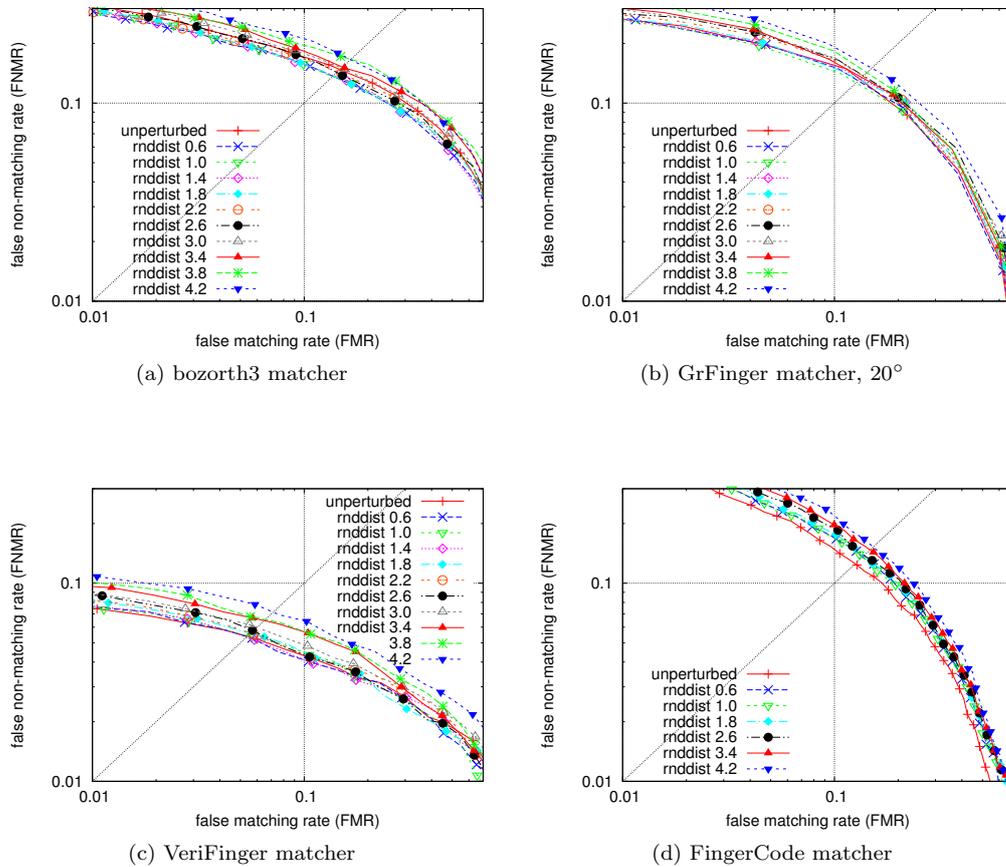


Figure 5.53: Receiver operating characteristics for the StirMark test *Small Random Distortions* on fingerprint images of DB1. (Plots apply a logarithmic scale. Ranges are set to FMR [1%,70%] and FNMR [1%,40%]).

100% FMR and 0% FNMR.

Turning to the two most-effective test-levels within the reduced set of *Small Random Distortions* tests, rddist 3.4 and 4.2, we can make another interesting observation in the results of the Phase Only Correlation matcher: From rddist 2.6 to rddist 3.4 we witness a rather drastic drop in the matcher’s performance – the equal error rates differ by 6.52% – while the gap between rddist 3.4 and rddist 4.2 again is comparatively small, with a relative deterioration of the respective equal error rate by “only” 1.75%.

DB2

When comparing **VeriFinger**’s matching performance in current *Small Random Distortions* tests on fingerprint images of DB2 with its performance in previous tests on images of DB1, we can note a slight aggravation in the overall matching results: For example, in Table 5.38 we find, that the equal error rates related to the individual levels of test-effectiveness are in average 0.50% farther distant from the equal error rate for unperturbed DB2-images, than they are in case of DB1-images. However, in comparison to the other fingerprint matchers

Factor	bozo3 (%)	VF (%)	GF180 (%)	GF20 (%)	FC (%)	POC (%)
unperturbed	14.81	5.87	11.41	13.61	12.54	22.60
0.6	13.69	5.71	11.18	13.40	13.60	21.57
1.0	13.67	5.72	11.87	13.00	13.58	21.45
1.4	13.82	5.77	12.04	13.74		
1.8	13.76	5.82	12.03	13.46	13.67	22.64
2.2	14.42	6.06	12.76	14.13		
2.6	14.31	6.03	13.12	14.20	13.79	23.88
3.0	15.06	6.26	13.24	14.16		
3.4	15.20	6.73	13.49	13.97	14.75	30.40
3.8	16.66	6.85	14.23	14.78		
4.2	16.67	7.21	14.85	15.48	15.24	32.14

Table 5.37: Equal error rates for *Small Random Distortions* test conducted on sample image database DB1.

regarded in current tests, VeriFinger is still the one matcher, whose matching results are, in general the, the least influenced by the perturbations introduced in the fingerprint images during the tests.

Like the VeriFinger matcher, also the **bozorth3** matcher exhibits a deteriorated matching performance, when working on the perturbed fingerprint images of DB2. While in the tests for DB1, the matching results of the first six levels of test-effectiveness are better even, than the results for unperturbed images, in case of DB2-related tests, the same can only be said for the weakest test, rnddist 0.6. Overall – and this point, as well, is different to bozorth3’s behavior in the previous tests – we can observe, that with increasing effectiveness of the applied *Small Random Distortions* tests, also the matching results aggravate accordingly. The ROC plot in Figure 5.54a clearly illustrates this observation, as does the corresponding listing of equal error rates in Table 5.38. Another aspect, that we can find when regarding the equal error rates, is, that for the lower test-levels, up until and including rnddist 2.2, bozorth3 shows a higher robustness to the image perturbations in DB2-images, than both GrFinger configurations, GF20 and GF180. For test-levels rnddist 2.6 to 3.8 then, the matching performances of GrFinger and bozorth3 are quite comparable. For the highest test-level, rnddist 4.2, though, bozorth3’s matching results exhibit clearly the strongest impairment of all minutiae-based matchers’ respective results.

Inspecting the matching results of the GrFinger matcher, we can establish also for the *Small Random Distortions* tests on images of DB2, that **GF180** once again shows a higher sensitivity to the perturbations introduced in the fingerprint images, than **GF20**. However, the respective differences to the relative impairments found in the test-results of GF20 are not as distinct anymore. This correlates to the fact, that in comparison to the results achieved in previous tests for DB1-images, the current matching performance of GF20 is clearly deteriorated, while in case of GF180 only 5 of 10 test-results for DB2-images are relatively stronger impaired, than the corresponding results for images of DB1.

For images of DB2 now, we can decidedly state, that both **non-minutiae-based** fingerprint matchers, FingerCode and Phase Only Correlation, are more sensitive to the perturbations

of the *Small Random Distortions* tests, than any of the minutiae-based fingerprint matchers – at least this is true for the reduced set of tests conducted on the non-minutiae-based matchers, even though the matcher’s results so far suggest a rather similar behavior for the remaining levels of test-effectiveness.

The overall reaction of the **FingerCode** matcher to the perturbations in fingerprint images of DB2, generated by the *Small Random Distortions* tests, is very much comparable to its reaction in the previous tests on DB1-images: Already at the first test-level, rnddist 0.6, we can witness a relatively strong impairment in the matching results – a glance at Table 5.38 reveals, that the EER for rnddist 0.6 is already 2.82% higher, than the EER in the results for unperturbed images. Any further increase in the tests’ effectiveness though, causes only a comparatively small additional aggravation in the respective test results. Aside from the listing of EER, the described behavior can also be observed in Figure 5.54d, where we can see, how comparatively close the ROC curves corresponding to the individual test-levels are to each other (especially in the area of FMR > 15% about), yet how clearly this group of curves is separated from the ROC curve representing the FingerCode’s matching results for the original, unperturbed DB2-images. Inspecting the developing of the individual curves more closely, we can also make a further, quite interesting observation in FingerCode’s ROC plot: In the area of FMR < 20% about, the curves portrait the expected behavior, that with each increasing level of perturbation-intensity in the fingerprint images, the matching performance gradually deteriorates, while from a FMR of about 20% onwards, it turns out, that FingerCode produces obviously lower FNMR for the higher effective test-levels rnddist 2.6 to 4.2, than it does for the lesser effective levels rnddist 0.6 to 1.8.

Comparing now the **Phase Only Correlation** matcher’s performance on the perturbed images of current tests to that of FingerCode, the respective ROC plots as well as the related equal error rates in Table 5.38 depict the following situation: For tests rnddist 0.6 and 1.0, POC’s matching results are clearly less influenced by the image-perturbations than FingerCode’s – while, for example, for test rnddist 0.6 the EER in the results of FingerCode is 2.82% higher than the respective EER for unperturbed images, this difference amounts to only 1.20% in the corresponding results of POC. However, for the more effective tests rnddist 1.8 to rnddist 4.2 then, it is the FingerCode matcher, that undoubtedly exhibits the greater robustness. For instance, regarding the listing of equal error rates again, we find, that POC’s EER in the results for rnddist 1.8 to rnddist 4.2 are up to nearly three times as far separated from its EER for unperturbed images, than are the corresponding EER of the FingerCode matcher.

DB3

In regard to the *Small Random Distortions* tests on fingerprint images of DB3, the matching performances of all examined fingerprint matchers, but the Phase Only Correlation matcher, exhibit a general improvement – i.e. a smaller manifestation of the negative influences by the generated image perturbations in the respective matching results – compared to the performances displayed in the previous section for tests on images of DB2. Furthermore for VeriFinger and the GrFinger configuration GF180, the current test results also show less impairment in comparison to those, generated in the tests on fingerprint images of DB1.

Of all the fingerprint matchers regarded in current *Small Random Distortions* test for DB3-images, **VeriFinger** once more turns out to be the matcher, the most robust to the influences of the image perturbations created by the tests. Inspecting the equal error rates of VeriFinger’s matching results in Table 5.39, we can observe two interesting aspects: For one,

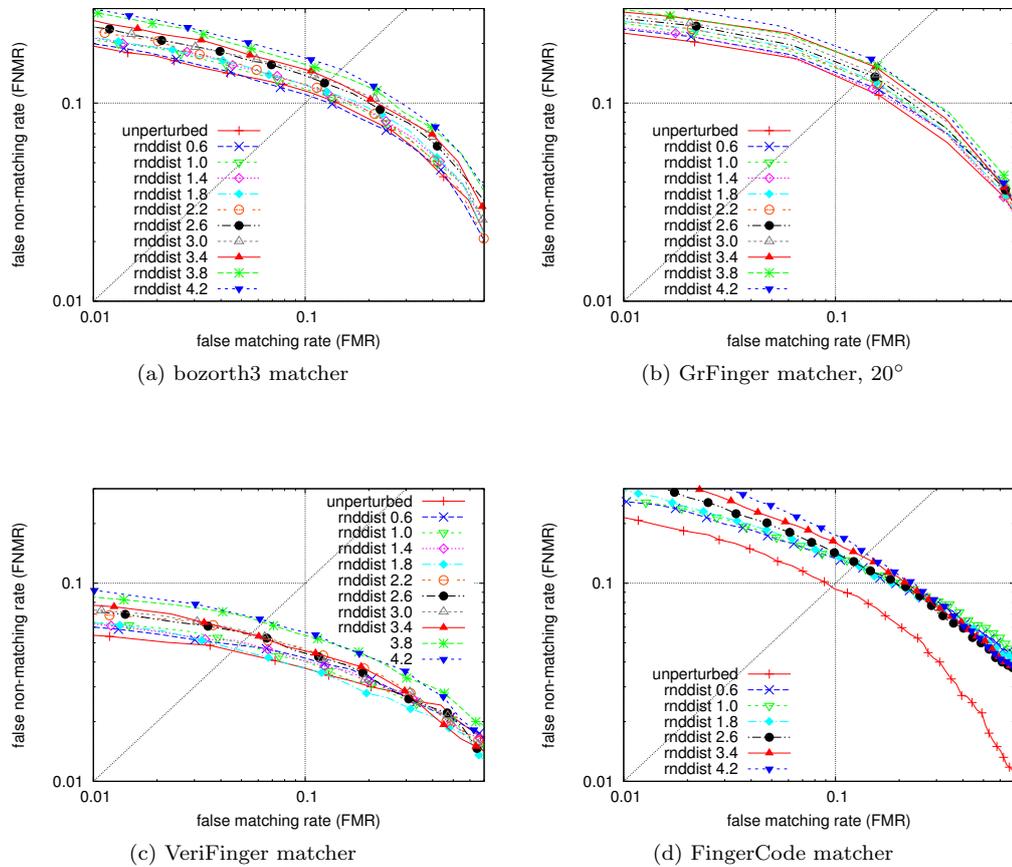


Figure 5.54: Receiver operating characteristics for the StirMark test *Small Random Distortions* on fingerprint images of DB2. (Plots apply a logarithmic scale. Ranges are set to FMR [1%,70%] and FNMR [1%,40%]).

we notice, that in the first six levels of increasing test-effectiveness, rddist 0.6 to 2.6, the corresponding EER lie a little below the EER in the results for the original, unperturbed fingerprint images of DB3, which indicates a slightly better matching performance for those test-levels. Second – and this reveals just how little VeriFinger’s matching performance is affected by the perturbations introduced in images of DB3 – all test-related EER lie within a distance of between -0.19% and 0.51% from the EER related to the unperturbed images.

As mentioned before, the results of current tests show, that also **GF180** is less sensitive to the *Small Random Distortions* perturbations in images of DB3, than was the case in the previous tests for fingerprint images of DB1 and DB2. Yet still – and this behavior is in line with the previously exhibited results – also for fingerprint images of DB3, **GF20** shows a higher robustness to the image-perturbations than GF180, even though, when regarding the listing of equal error rates in Table 5.39, we find, that the differences between the relative matching performances of GF20 and GF180 are marginally at best. Another observation that we can make in the respective equal error rates and that is likewise discernible in GF20’s ROC plot in Figure 5.55b, is, that for the less-effective test-levels rddist 0.6 to 2.2, both GrFinger configurations produce slightly better matching results, than they do for the original unperturbed images of DB3. Interesting to note: Among denoted set of tests,

Factor	bozo3 (%)	VF (%)	GF180 (%)	GF20 (%)	FC (%)	POC (%)
unperturbed	11.12	5.01	11.72	12.89	9.60	9.69
0.6	10.78	5.17	12.27	13.32	12.42	10.89
1.0	11.35	5.58	12.62	13.76	12.34	11.49
1.4	11.66	5.83	12.17	13.01		
1.8	11.75	5.51	12.92	13.89	12.40	13.23
2.2	11.79	5.54	12.94	13.76		
2.6	12.57	5.78	13.01	14.29	12.61	16.34
3.0	13.07	6.16	13.26	14.68		
3.4	13.23	5.69	14.22	15.32	13.57	19.00
3.8	14.05	6.43	14.71	15.57		
4.2	14.82	6.77	15.00	16.14	14.05	21.96

Table 5.38: Equal error rates for *Small Random Distortions* test conducted on sample image database DB2.

a higher intensity of the image-perturbations does not necessarily cause stronger impaired matching results. A statement, that, by the way, can also be made about the results of VeriFinger for the first six test-levels (see above for details).

As for the matching performance of the **bozorth3** matcher in current tests on fingerprint images of DB3: When regarding the corresponding ROC plot in Figure 5.55a it becomes quite apparent, that bozorth3 is the one minutiae-based matcher, whose matching results are subject to the strongest impairment, caused by the *Small Random Distortions* perturbations in the fingerprint images. Also the listing of bozorth3's equal error rates in Table 5.39 supports this observation: While for the second most sensitive minutiae-based matcher, GF180, the maximum difference between the EER related to the unperturbed DB3-images and the EER of a *Small Random Distortions* test is 1.15%, bozorth3 surpassed this level of impairment already for test rnddist 3.0. For tests rnddist 3.4 to 4.2 the negative effect of the image perturbations on the matching results increases further, so that finally the EER in bozorth3's results for test rnddist 4.2 is 3.37% higher than the EER for the unperturbed images.

Like in the *Small Random Distortions* tests for images of DB2, now also here, in current tests on images of DB3, both **non-minutiae-based fingerprint matchers** are clearly stronger influenced by the generated image-perturbations, than any of the minutiae-base matchers regarded. Furthermore, and also in line with the observations in previous *Small Random Distortions* tests for fingerprint images of databases DB1 and DB2, the results of the Phase Only Correlation matcher are notably stronger impaired, than those of the FingerCode matcher.

In the previous tests for images of DB1 and DB2 alike, we could observe in the matching results of the **FingerCode** matcher, that already with the perturbations created by the least effective test-level, rnddist 0.6, we induced a rather distinct "initial" impairment in the matcher's results – well observable in the corresponding ROC plots, in form of a clear gap between the ROC curve displaying the FingerCode's performance for rnddist 0.6 and the ROC curve related to the original, unperturbed fingerprint images. In present tests for

images of DB3, this effect can not be witnessed anymore. However, judging by the equal error rates in Table 5.39, we can establish, that for images of DB3 the negative influence of the image-perturbations, caused by the *Small Random Distortions* tests, on FingerCode’s overall matching performance, is clearly less, than is the case for images of DB2. When comparing with the results of the tests on images of DB1 on the other hand, the same can only be said for test-levels rnddist 0.6 to 1.8, while for the more intensive perturbations introduced by rnddist 2.6 to 4.2, the results for fingerprint images of DB3 appear notably stronger impaired.

Despite the fact, that the **Phase Only Correlation** matcher is once again the one fingerprint matcher regarded, whose matching performance is decidedly the most susceptible to the negative influences caused by the perturbations of the *Small Random Distortions* perturbations in the fingerprint images of DB3, the POC matcher is also the only matcher, whose results for current tests on images of DB3 do not show any relative improvement in robustness to the perturbations, when compared with its results generated in the previous tests for images of DB1 or DB2. As the listing of equal error rates in Table 5.39 displays, for the two least effective tests, rnddist 0.6 and 1.0, POC’s matching performance for DB3-images is very comparable to that, exhibited for fingerprint images of DB2. The more effective tests, rnddist 1.8 to 4.2, however, cause a deterioration in the matching results, that is clearly stronger, than previously observed for images of DB1 or DB2. As illustrative example we can regard the EER in the database-specific results for the most effective test-level, rnddist 4.2 and calculate its difference to the respective EER achieved for unperturbed images: Said difference amounts to 9.55% in the tests for DB1, to 12.27% in the tests for DB2 and finally to 15.10% in current tests for fingerprint images of DB3.

Factor	bozo3 (%)	VF (%)	GF180 (%)	GF20 (%)	FC (%)	POC (%)
unperturbed	6.68	3.60	6.90	5.82	8.98	15.07
0.6	6.74	3.41	6.70	5.61	9.45	16.28
1.0	6.95	3.33	6.92	5.57	9.61	16.80
1.4	7.04	3.30	6.77	5.29		
1.8	7.21	3.50	6.71	5.55	10.00	19.18
2.2	7.56	3.59	6.70	5.53		
2.6	7.58	3.31	7.47	5.89	10.60	22.43
3.0	8.15	3.62	7.68	6.38		
3.4	8.78	4.12	7.75	6.50	11.39	27.58
3.8	9.59	3.93	7.42	6.23		
4.2	10.06	3.99	8.05	6.50	12.87	30.17

Table 5.39: Equal error rates for *Small Random Distortions* test conducted on sample image database DB3.

5.15 Latest Small Random Distortions (lrnddist)

This test represents the latest modification of *the* StirMark test, the way it was introduced with version 4.0 of the benchmark (as the comments in the source code suggest). It differs

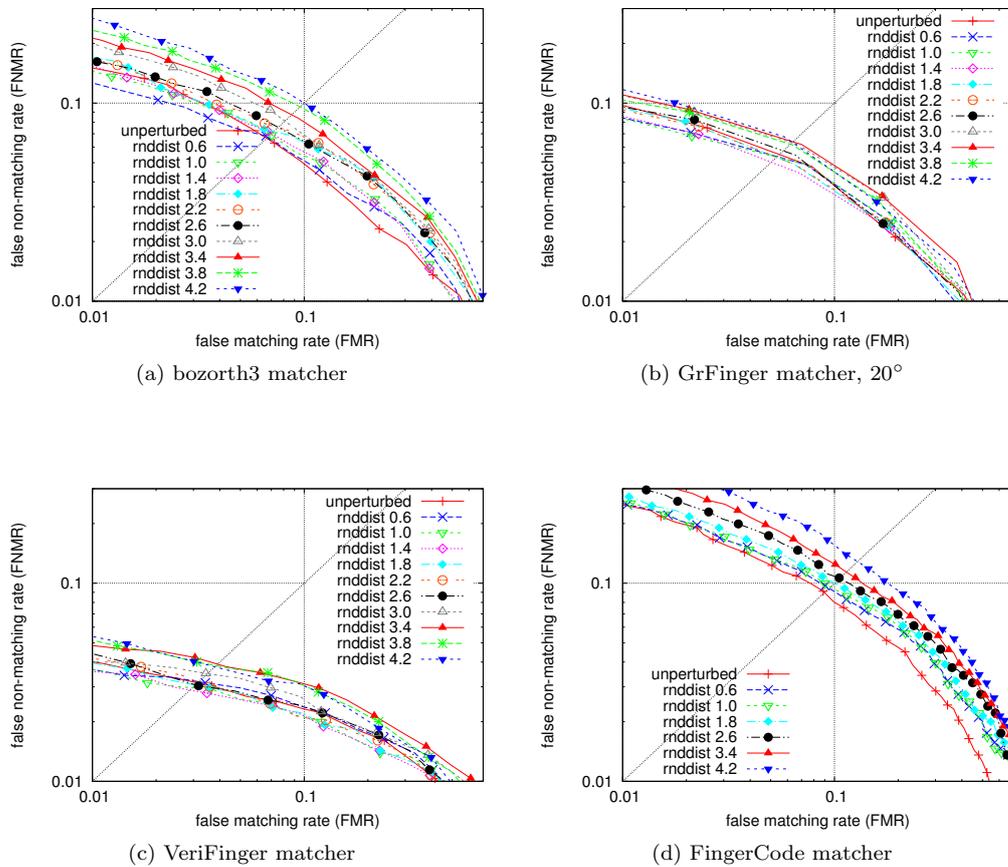


Figure 5.55: Receiver operating characteristics for the StirMark test *Small Random Distortions* on fingerprint images of DB3. (Plots apply a logarithmic scale. Ranges are set to FMR [1%,70%] and FNMR [1%,40%]).

from the *Small Random Distortions* test in so far, as it replaces the two processing steps *Global Bending* and *Higher Frequency Displacement* with a single distortion procedure:

The two original steps strongly rely on sine-functions to determine the individual distortion of each point in the output image (which is clearly noticeable in a periodical pattern introduced in the perturbed images for high values of the attack parameter). In contrast, the new function employs two random *Distortion Maps* instead – one map influences the distortion of the x-coordinate and the other the distortion of the y-coordinate of each point in the output image.

According to a comment at line 241 in file `SMBDistortionMap.cpp` of the StirMark Benchmark source files, a distortion map is generated “*by specifying pseudo randomly the Fourier transform of this map*”. More information on the creation of the distortion maps and on the specifics of the *Latest Small Random Distortions* test in general, can likewise (only) be found in the source code of the related classes, downloadable at [25].

Relation to Fingerprints

Very much like with the previously listed *Small Random Distortions* attack, also by employing the *Latest Small Random Distortions*, I aim to simulate a random, haphazard combination and interaction of various natural distortions in fingerprint images. So while the intentions for the application of both “original” StirMark tests are the same, the perturbations generated by the tests differ distinct enough, for me to embrace this *Latest Small Random Distortions* attack as quasi “second type” of random combination of natural distortions.

Parameter Configurations

Like with above older version of *the* StirMark test, also here in *Latest Small Random Distortions* a single parameter, representing a multiplier for the default parameters, is used to adjust the intensity of the attack.

For the experiments of present work, I chose the same parameter sets as previously with the *Small Random Distortions* attack: {0.6, 1.0, 1.4, 1.8, 2.2, 2.6, 3.0, 3.4, 3.8, 4.2} for the complete analysis and {0.6, 1.0, 1.8, 2.6, 3.4, 4.2} for the limited number of tests, run on the non-minutiae-based fingerprint matchers. Figure 5.56 shows example images generated for parameter values 0.6, 2.6 and 4.2.

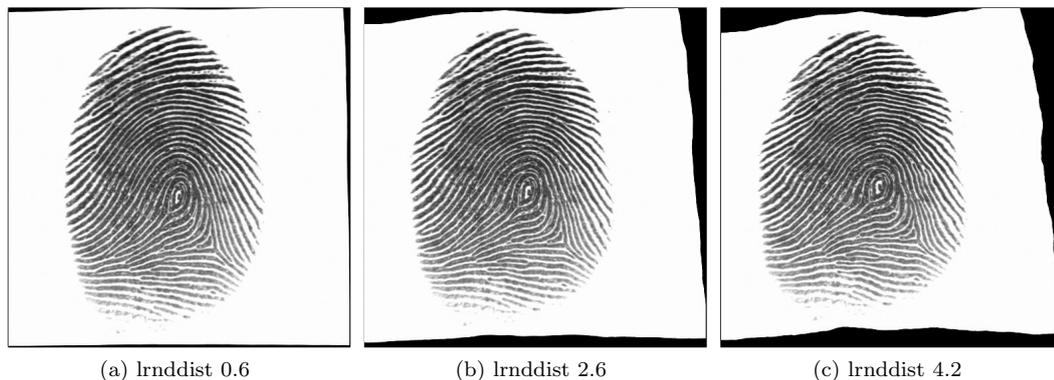


Figure 5.56: Examples for the *Latest Small Random Distortions* test, applied to an image from DB1 (ID 91.2).

Similar to the *Small Random Distortions* test, for parameter values < 0.6 the influence of the attack is barely noticeable in the output images. Yet due to the substitution of the two sine-function-based processing steps with random “Distortion Maps”, aforementioned wave-like structures in the ridge lines of the fingerprint do not appear anymore when selecting high parameter values.

Results and Discussion

Very much comparable to the observations in the previous test-series applying *the* original StirMark test *Small Random Distortions*, we can also state for the revisited StirMark test *Latest Small Random Distortions*, that the created image perturbations, introduced in fin-

gerprint images of databases DB1 to DB3, have only a very limited impact on the individual matching results of the fingerprint matchers employed in our experiments.

DB1

Regarding the matching results of the fingerprint matcher **VeriFinger** for the *Latest Small Random Distortions* tests on fingerprint images of DB1, we find, that in the four lesser effective test-levels, lrnddist 0.6 to 1.8, VeriFinger’s matching performance is nearly equal or in parts even better, than it is originally for the unperturbed images of DB1. A look at the listing of equal error rates in Table 5.40 confirms, that the corresponding EER are up to 0.65% lower than the EER in the matching results for unperturbed images. From test lrnddist 2.2 onwards then, with each increase in test-effectiveness, the relative impairment caused in the matching results, becomes likewise more apparent – a behavior that can also be observed in the related ROC plot in Figure 5.57c. Finally the EER in VeriFinger’s results for the most effective test, lrnddist 4.2, lies 2.00% above that related to the unperturbed images.

In comparison with the other fingerprint matchers regarded, we can establish, that overall VeriFinger’s matching performance exhibits the highest robustness to the influences caused by the perturbations, introduced during the *Latest Small Random Distortions* tests into the fingerprint images of DB1. However, this “lead” over the other fingerprint matchers is by far not as distinct, as we could witness it in the set of *Small Random Distortions* tests in previous section 5.14.

The fingerprint matcher, that displays the second least influence by the perturbations in DB1 images, is the GrFinger configuration **GF20**. As opposed to the results of VeriFinger, here, for GF20, only the matching results for lrnddist 0.6 are in parts better than the matcher’s results for unperturbed images of DB1 – the ROC plot in Figure 5.57b shows, that while the ROC curve corresponding to test lrnddist 0.6 is constantly very close to the ROC curve representing the matching results for the unperturbed images, it is only in the area of FMR < 2% and also the area of FMR short before the EER, that the test-related curve indicates the smallest FNMR. On the other hand, we further find in the plot, that also the ROC curves related to tests lrnddist 1.0 to 1.8 lie in a relatively close distance to the curve for unperturbed images (in the area of FMR > 45%, the curve corresponding to the results for lrnddist 1.4, even lies considerably below all others). This observation is supported by the respective equal error rates in Table 5.40: The EER for tests lrnddist 0.6 to 1.8 all lie within a range of $\pm 0.11\%$ around the EER for the original, unperturbed fingerprint images of DB1. For more effective test levels then, the negative influences of the image-perturbations on GF20’s matching results, become more discernible. In particular between the tests lrnddist 3 and 3.4 we experience a rather notable drop in matching performance, as also the respective EER “jump” from 14.30% to 15.05%.

When comparing **GF180**’s matching results to those of GF20, we can establish, that GF180 is clearly more susceptible to the perturbations introduced by the *Latest Small Random Distortions* tests into the fingerprint images of DB1. In fact, GF180 exhibits the worst – i.e. most influenced – overall matching performance of all minutiae-based matchers regarded: For the least effective test level, lrnddist 0.6 the matching results of GF180 are actually even slightly better than those, it achieves for the original, unperturbed images of DB1 – the respective EER lies 0.17% below that for unperturbed images and also in the ROC plot, the ROC curve related to test lrnddist 0.6 lies below all others. However, when increasing the level of test-effectiveness, the corresponding results deteriorate comparatively strong.

Regarding the listing of equal error rates in Table 5.40, we find, that already from lrnddist 0.6 to the following test, lrnddist 1.0, the EER increases by 1.07%. Although the results suggest, that with GF180, an amplification of the *Latest Small Random Distortions* perturbations in the fingerprint images does not necessarily cause an accordingly stronger deterioration of the matching performance, for test levels lrnddist 1.0 and above the cognizable impairment is permanently on a relative high level. For the most-effective test, lrnddist 4.2, finally, the corresponding EER is 4.42% higher than GF180's EER for unperturbed images.

The reactions of the **bozorth3** matcher to the *Latest Small Random Distortions* perturbations in fingerprint images of DB1 are very interesting in so far, as for tests lrnddist 0.6 to lrnddist 1.8 (and in parts even for test lrnddist 2.6) bozorth3 achieves decidedly better matching results, than it does for the original, unperturbed DB1-images: Regarding the respective ROC plot in Figure 5.57a, we can see, that the ROC curves, corresponding to said four least-effective test levels, lie clearly below the ROC curve indicating bozorth3's matching performance on the unperturbed images. The same can also be observed for the ROC curve related to test lrnddist 2.6, for $FMR > 14\%$. When additionally referring to the listing of equal error rates in Table 5.40, we find that the EER in the bozorth3's results for tests lrnddist 0.6 to 1.8 are between 0.78% and 1.21% lower than the EER for the unperturbed fingerprint images. Another aspect of bozorth3's matching performance, that becomes apparent in the ROC plot, is, how relatively far the ROC curves for the results of the two most-efficient *Latest Small Random Distortions* tests, lrnddist 3.8 and 4.2, are separated from the set of the other curves. This marks the comparatively large drop in matching performance between test levels lrnddist 3.4 and 3.8 – the corresponding EER differ by 2.36%.

The matching behavior, that the **FingerCode** matcher exhibits in current *Latest Small Random Distortions* tests on fingerprint images of DB1 is very comparable to its behavior in the previous *Small Random Distortions* tests on DB1-images (see page 175 for details): Already with the least effective test, lrnddist 0.6, the impairment in the matching results is as such, that the corresponding EER is 1.05% higher, than the EER achieved for unperturbed fingerprint images. For the more effective tests however, the according aggravations in the results appear rather limited – until test-level lrnddist 2.6 the further increase in the respective EER is no more than 0.41% and even the EER in the results for the most effective test, lrnddist 4.2, is only 1.66% higher, than that for lrnddist 0.6. For none of the other matchers regarded in current tests, we find the matching performances for the individual perturbation-levels to lie that close to each other – an observation that becomes also very apparent, when comparing the corresponding ROC plots in Figure 5.57.

A further statement that we can make, when comparing FingerCode's matching performances in current tests, with those of the minutiae-based matchers, is, that for tests lrnddist 2.6, 3.4 and 4.2 (the three most effective test-levels in the reduced set of tests, that were also conducted on the non-minutiae-based matchers) FingerCode turns out to be more robust to the perturbations in the fingerprint images, than GF180. For lrnddist 4.2 this can also be said in regard to bozorth3's performance (please refer to the listing of equal error rates in Table 5.40 for details).

In the results of the two least effective *Latest Small Random Distortions* tests we can see a surprisingly strong improvement in the **Phase Only Correlation** matcher's performance, relative to the results achieved for the original, unperturbed images of DB1. In the corresponding ROC plot, we find, that the ROC curve related to test lrnddist 0.6, as well as, from a $FMR > 10\%$ about onwards, also the ROC curve related to test lrnddist 1.0 lie clearly below the ROC curve displaying POC's matching performance for the unperturbed images.

The listing of equal error rates in Table 5.40 indicates, that the EER in the results for tests lrnddist 0.6 and 1.0 are 1.68% and 0.93% respectively lower than the EER for the original DB1-images.

Turning to the medium-effective tests, lrnddist 1.8 and 2.6, then, we can also observe an interesting matching behavior: In the corresponding matching results, the impairment caused by the perturbations in the fingerprint images is clearly noticeable – especially in the area of FMR < 35% about, the negative influences cause a comparatively strong deterioration of the matching performance, which can also be witnessed in the corresponding EER, that are up to 2.07% higher than the EER for unperturbed images. Yet when regarding the the ROC plot again, we find that from the point of 35% FMR about, the ROC curve related to test lrnddist 1.8 drops under the curve representing POC’s results for the unperturbed images. At FMR of 48% about, also the ROC curve related to test lrnddist 2.6 follows.

Inspecting the Phase Only Correlation’s results for the two most-effective test-levels regarded on this matcher, lrnddist 3.4 and 4.2, the influence by the perturbations in the fingerprint images becomes very apparent, as we witness a comparatively large drop in the matcher’s performance. For illustration we can once more refer to the listing of equal error rates in Table 5.40: While the EER in the results for the four less effective test-levels are all within a range of about $\pm 2\%$ from the EER for the original, unperturbed images, the EER related to the perturbations in lrnddist 3.4 has already a distance of 8.36% and that for lrnddist 4.2 a distance of 11.86%.

Factor	bozo3 (%)	VF (%)	GF180 (%)	GF20 (%)	FC (%)	POC (%)
unperturbed	14.81	5.87	11.41	13.61	12.54	22.60
0.6	13.60	5.88	11.24	13.49	13.59	20.92
1.0	14.00	5.24	12.31	13.71	13.75	21.67
1.4	13.88	5.72	12.25	13.72		
1.8	14.04	5.71	12.12	13.72	13.85	23.27
2.2	15.23	5.84	12.54	14.16		
2.6	14.76	6.37	13.49	14.00	14.00	24.63
3.0	15.75	6.22	13.12	14.30		
3.4	15.99	7.15	14.29	15.05	14.71	30.96
3.8	18.35	7.39	14.60	15.80		
4.2	18.13	7.88	15.82	15.53	15.24	34.46

Table 5.40: Equal error rates for *Latest Small Random Distortions* test conducted on sample image database DB1.

DB2

The overall matching performance of the *VeriFinger* matcher in the *Latest Small Random Distortions* tests on fingerprint images of DB2 is very much alike to that, exhibited by the matcher in the previous tests for images of DB1 – the extent by which the influence of the image-perturbations affects the matching results for each individual test-level is largely comparable. Further, when comparing the corresponding ROC plots (see Figures 5.57c and

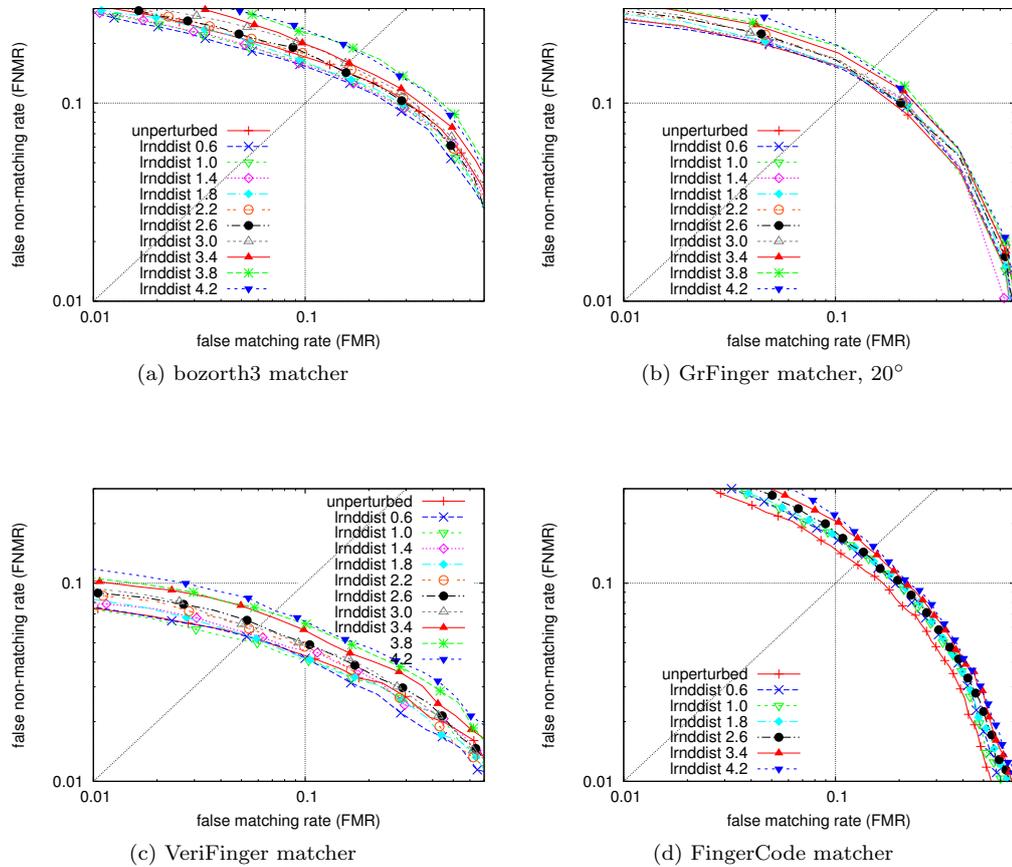


Figure 5.57: Receiver operating characteristics for the StirMark test *Latest Small Random Distortions* on fingerprint images of DB1. (Plots apply a logarithmic scale. Ranges are set to FMR [1%,70%] and FNMR [1%,30%]).

5.58c) we find in both plots a quite similar matching behavior: The degradation in the results for the higher-effective tests, lrnddist 3.4 to 4.2 (and in parts also lrnddist 3.0), is decidedly stronger than in the results of the remaining, less effective tests. Accordingly the ROC curves related to these tests, lrnddist 3.0 to 4.2, permanently lie above all other ROC curves. Now regarding the ROC curves representing VeriFinger’s matching performance for the less effective tests, lrnddist 0.6 to 2.6, for $FMR < 3.5\%$ they all lie above the ROC curve related to the original, unperturbed images. For $FMR > 3.5\%$ though, one after the other, the ROC curves corresponding to lrnddist 0.6 to 2.6 start to under-run that for unperturbed images – starting with the curve for lrnddist 1.8, followed by lrnddist 1.4 at a FMR of 13%, etc., until a FMR of 33%, from which point on even the ROC curve corresponding to lrnddist 2.6 indicates a relatively lower FNMR per FMR, than does the ROC curve for VeriFinger’s original results for unperturbed DB2-images. Altogether, like in previous tests on fingerprint images of DB1, also in current tests VeriFinger turns out to be the matcher most robust to the generated images-perturbations.

Regarding the matching performance of both GrFinger setups, **GF20** and **GF180**, in the current tests on fingerprint images of DB2 we can make several interesting observations: For one, in both setups the matching results of the least-effective test, lrnddist 0.6, appear

comparatively strong influenced by the *Latest Small Random Distortions* perturbations in the DB2-images (for illustration: the respective EER are 0.89% (GF20) and 0.79% (GF180) higher, than the EER for unperturbed images), while in case of previous DB1-related tests, the corresponding results turn out to be even slightly better, than those for unperturbed fingerprint images. Generally the matching results of GF20 for all current tests, but lrnddist 3.8, exhibit a stronger negative impact by the image-perturbations than is the cases for images of DB1. The equal error rates in Table 5.41 show, that in average GF20's EER, related to current tests, lie 0.48% further distant from the EER for unperturbed fingerprint images of DB2, than they did in the previous tests for DB1. In contrast, the matching results of GF180 for all current tests, but lrnddist 0.6 (as mentioned above), are equally or even less affected by the perturbations introduced in DB2-images, than witnessed in DB1-related tests. Overall however, GF20's matching results are still less impaired than those of GF180.

Compared with its matching performance in previous tests for images of DB1, in current tests for DB2, the **bozorth3** matcher exhibits clearly a higher sensibility to the *Latest Small Random Distortions* perturbations introduced in the fingerprint images. For illustration we can, for example, examine the respective equal error rates in Table 5.41: In average, the EER in bozorth3's matching results for the individual lrnddist tests on DB2-images are 0.96% further distant from the corresponding EER for the original, unperturbed images, than we find it in the results for the tests on DB1-images (please see also Table 5.40 for details). A further interesting observation in regard to the listing of equal error rates is, that the EER for test-levels lrnddist 0.6 and 1.4 indicate a relatively better matching performance of bozorth3 in those tests, than for the unperturbed DB2-images. When additionally regarding the ROC plot in Figure 5.58a, we find, that this is only true in the area of FMR between 6.5% and 22% about. In the complementary area the ROC curves related to lrnddist 0.6 and 1.4 lie rather clearly above the ROC curve representing the matching results for the original, unperturbed fingerprint images. Instead then, from the same point of 22% EER on, the matching results of lrnddist 1.8 produce the relatively lowest FNMR values.

All in all the observations lead to the verdict, that for fingerprint images of DB2, bozorth3 appears to be the minutiae-based fingerprint matcher, whose matching performance is the most influenced by the image perturbations generated by the individual *Latest Small Random Distortions* tests. Overall however, the **non-minutiae-based fingerprint matchers** exhibit a decidedly higher sensitivity to the influence of the perturbations in the fingerprint images of DB2, than the minutiae-based matchers. Further, the results of the FingerCode matcher and the Phase Only Correlation matcher alike, show, that in current tests on images of DB2 the impairment is also relatively stronger than in the previous tests on images of DB1.

Regarding the equal error rates for the **FingerCode** matcher in Table 5.41, we see, that already for the least effective test-level, lrnddist 0.6, the EER is 2.56% higher than the EER achieved for the original, unperturbed DB2-images. The EER related to the other tests then stay at such a high level and even increase, so that finally for the most effective test, lrnddist 4.2, the difference to the EER for unperturbed images amounts to 5.29%. The corresponding ROC plot in Figure 5.58d supports our observations: The most obvious aspect is, that the ROC curve, displaying FingerCode's matching performance on unperturbed images, permanently lies very clearly under the ROC curves related to the individual *Latest Small Random Distortions* tests. For FMR > 10% about, this gap even widens considerably. The test-related ROC curves then have an interesting behavior as well: For FMR < 15% the individual curves lie fairly separated. For FMR > 15% though, they move comparatively close together – all, except the ROC curve related to test lrnddist 0.6, which gradually

exchanges “positions” with the other test-related curves, until finally, for $FMR > 25\%$, it indicates the highest FNMR per FMR in all of FingerCode’s test results.

Concerning the matching performance of the **Phase Only Correlation** matcher in current tests on fingerprint images of DB2, it is rather sufficient to regard the listing of respective equal error rates alone, as the ROC plot supports the observation of the quite distinct differences in the relative impairment of the matching results and does not add any further essential details. So when regarding Table 5.41, we can establish the following aspects: The EER related to the least-effective test, lrnndist 0.6, is 1.0% higher, than the EER for the unperturbed DB2-images. This difference is smaller, than we witness it for the same test-level in the results of the FingerCode matcher (there: 2.56%, see above for details). In other words, the perturbations introduced by test lrnndist 0.6 into the fingerprint images of DB2, deteriorate FingerCode’s matching performance notably more, than that of the Phase Only Correlation matcher. A similar statement can be made, when comparing both non-minutiae-based matcher’s results for the following test, lrnndist 1.0. Also here the POC matcher appears less influenced by the image perturbations, even though the difference is not as clear as before. However, for the remaining tests-levels from the reduced set of tests conducted on the non-minutiae-based fingerprint matchers, the situation changes considerably: For test lrnndist 1.8 the respective EER of POC is already 4.16% higher, than the EER for unperturbed images, while the corresponding difference is only 3.01% for FingerCode. This gap in both matcher’s performances even widens, until finally for lrnndist 4.2, which introducing the strongest *Latest Small Random Distortions* perturbations considered in present tests, POC’s EER lies 14.69% above the EER it achieves for unperturbed images, while in FingerCode’s results this difference is “only” 5.29%.

Factor	bozo3 (%)	VF (%)	GF180 (%)	GF20 (%)	FC (%)	POC (%)
unperturbed	11.12	5.01	11.72	12.89	9.60	9.69
0.6	10.50	5.40	12.52	13.78	12.16	10.69
1.0	11.20	5.35	12.24	13.22	12.12	11.78
1.4	10.68	4.78	12.70	13.59		
1.8	11.71	4.97	12.52	13.94	12.61	13.85
2.2	12.96	5.64	13.13	13.76		
2.6	12.52	5.10	13.32	13.80	13.59	18.70
3.0	13.16	5.58	13.52	14.04		
3.4	13.44	6.22	13.65	14.84	13.83	21.45
3.8	14.84	6.39	14.96	14.75		
4.2	15.40	6.91	15.15	15.38	14.89	24.38

Table 5.41: Equal error rates for *Latest Small Random Distortions* test conducted on sample image database DB2.

DB3

Regarding the impact of *Latest Small Random Distortions* perturbations in fingerprint images of database DB3 on the matching performances of the fingerprint matchers examined

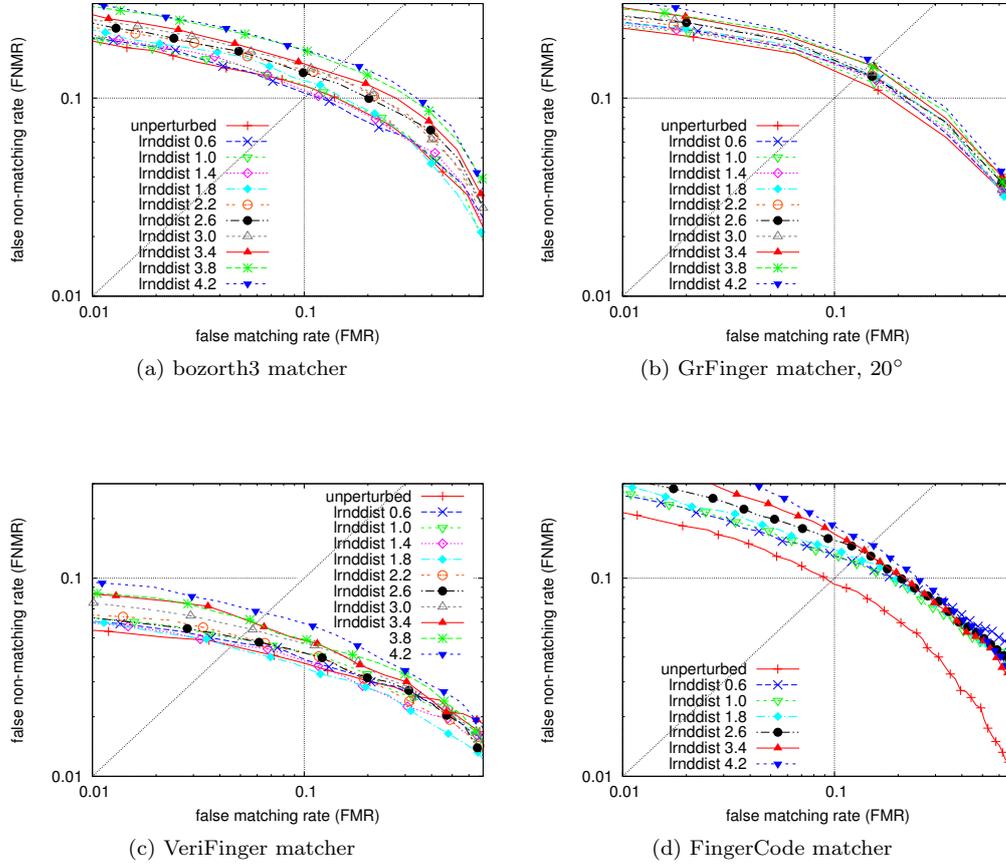


Figure 5.58: Receiver operating characteristics for the StirMark test *Latest Small Random Distortions* on fingerprint images of DB2. (Plots apply a logarithmic scale. Ranges are set to FMR [1%,70%] and FNMR [1%,30%]).

in our experiments, and comparing these observations to those made in previous tests for fingerprint images of databases DB1 and DB2, we get to following, quite interesting distinction: In case of matchers VeriFinger, GF20, GF180 and FingerCode the perturbations have clearly less influence on the respective matching results, when introduced in fingerprint images of DB3, than when introduced into images of DB1 or DB2. For matchers bozorth3 and Phase Only Correlation the opposite is true: Here the image perturbations in the fingerprint images of DB3 affect the respective matching results clearly stronger, than is observable in the previous tests for images of DB1 or DB2.

Inspecting the matching results of matcher **VeriFinger** closer, we find, that the cognizable effect of the *Latest Small Random Distortions* perturbations is only very limited. The corresponding equal error rates in Table 5.42 show, that the maximum difference between any one test-related EER and the EER for the original, unperturbed images of DB3 is only 1.08% (in case of the most effective test, lrnddist 4.2). In fact, 7 of 10 test-related EER are within a range of $\pm 0.55\%$ of the EER for the unperturbed images. Another observation we can make in the listing of equal error rates and which is also clearly evident in the corresponding ROC plot in Figure 5.59a, is, that for the 3 least-effective tests, lrnddist 0.6 to 1.4 and in parts also for the tests lrnddist 1.8 and 2.2, VeriFinger produces better matching results than it

achieves for the original, unperturbed fingerprint images.

Despite its extraordinary good – i.e. unimpaired – results, in current tests for images of DB3, VeriFinger is not the one matcher, exhibiting the highest robustness to the influences of the image perturbations. Even though its overall matching performance also includes an “outlier”, in average the matching results of **GF20** turn out to be slightly less influenced, than VeriFinger’s. Regarding, for example, the equal error rates in Table 5.42, we find, that the differences between GF20’s EER for the unperturbed fingerprint images and the EER corresponding to the individual *Latest Small Random Distortions* tests lrnndist 0.6 to 3.8, lie within the range of [-0.36%, 0.72%]. The only exception, aforementioned “outlier”, occurs in the results for the most effective test, lrnndist 4.2: The respective EER is 2.02% higher than that for the unperturbed images. A further observation we can make, is, that for the four least-effective test-levels, lrnndist 0.6 to 1.8, GF20, in parts, exhibits a better matching performance, than it does for the original, unperturbed fingerprint images of DB3 – this especially holds for the area of FMR < 10% about. GF20’s ROC plot in Figure 5.59b, confirms aforesaid statements: The two aspects that are the most obvious in the plot are, for one, how comparatively close the ROC curves for tests lrnndist 0.6 to 3.8 lie to each other and second, how relatively far the ROC curve corresponding to the matching results for test lrnndist 4.2 is separated from the others.

As mentioned before, comparing the matching performances of **GF180** in the *Latest Small Random Distortions* tests for all three databases, DB1 to DB3, the level of robustness to the image perturbations, that we witness in current tests for images of DB3 is higher, than it is for images of DB1 or DB2. In case of GF180, this relative increase in performance – or for that matter: decrease in impairment – is clearly more distinct, than we can observe it in matchers VeriFinger or GF20. (As for the respective behavior of the FingerCode matcher: In results for the reduced set of tests that were conducted on the non-minutiae-based matcher, we find, that the gain in robustness from the tests for DB1-images to the tests for DB3-images is even more distinct than in the corresponding results for GF180, yet clearly less distinct from tests for DB2-images to tests for DB3-images).

All in all however, also in the current tests for fingerprint images of DB3, GF180’s matching results appear slightly stronger influenced by the generated image-perturbations, than those of GF20 and VeriFinger. “Stronger influenced” thereby applies in both ways: For one, GF180 exhibits a better matching performance for test lrnndist 0.6 than for unperturbed images. The difference between the respective EER amounts to 0.70% – in case of GF20 and VeriFinger, which exhibit a comparable performance for test-level lrnndist 0.6, the differences between the test-related EER and that for unperturbed images are just 0.36% and 0.34% respectively. Regarding the other end of the scale, test lrnndist 4.2 for example, GF180’s EER is 2.92% higher than its EER for unperturbed images, while the corresponding differences 2.02% for GF20 and only 1.03% for VeriFinger.

Of the set of minutiae-based fingerprint matchers considered, **bozorth3** is the matcher, whose performance on fingerprint images of DB3 is the most impaired by the *Latest Small Random Distortions* perturbations introduced in the images. Regarding the corresponding equal error rates in Table 5.42, we find, that for the least-effective test, lrnndist 0.6, the related EER is 0.19% higher, than that for the original, unperturbed DB3-images. This difference then increases with each successive level of test-effectiveness, until finally the EER for test lrnndist 4.2 is 5.52% higher than the EER for the unperturbed images. In the respective ROC plot in Figure 5.59a, we can well observe said continuous increase in the EER rates. Additionally we can make two quite interesting observations: For one, we see that the three ROC curves representing the matching results for tests lrnndist 0.6, 1.0

and 1.4 and further also the three ROC curves related to tests lrnddist 1.8, 2.2 and 2.6 lie relatively close to each other. This finding holds especially in the area of FMR between 2% and 20% about and can also be witnessed in the respective equal error rates. Moreover we find in the ROC plot, that for $FMR < 4.5\%$ about, as well as for $FMR > 30\%$ about, the matching results for lrnddist 1.0 and 0.6, in parts also lrnddist 1.4 exhibit a better matching performance, than bozorth3 achieves for the original, unperturbed fingerprint images.

Comparing the matching results of both **non-minutiae-based matchers**, FingerCode and Phase Only Correlation, for the reduced set of *Latest Small Random Distortions* tests conducted on them, with the corresponding results of the minutiae-based fingerprint matchers, we can establish also in the current case of DB3-images, that in general the non-minutiae-based matchers are notably more susceptible to the negative influences caused by the perturbations in the fingerprint images. The only exceptions we find, are in relation to bozorth3's matching performance, as for tests lrnddist 1.8, 3.4 and 4.2 the FingerCode matcher exhibits a comparatively higher robustness to the perturbations. This behavior can best be observed, when inspecting the listing of the equal error rates in Table 5.42: Considering for example the matcher's results for test lrnddist 4.2, then FingerCode's respective EER value lies 4.18% higher, than the EER, it achieved for the original, unperturbed fingerprint images of DB3, while in case of bozorth3, the corresponding difference amounts to 5.52%.

When regarding the ROC plot for the **FingerCode** matcher in Figure 5.59d, the relative aggravation in the matching results, caused by the *Latest Small Random Distortions* perturbations in the fingerprint images of DB3 becomes apparent, as the ROC curves representing FingerCode's matching performance for the individual tests are fairly separated. However, the clear distance between the group of test-related ROC curves and the ROC curve indicating the matching results for the original, unperturbed images, that can be seen in the ROC plots for the previous tests for fingerprint images of DB1 and DB2 (see Figures 5.59d and 5.59d), is not as clearly observable in the plot for present matching results. When further referring to Table 5.42, we can establish, that the differences between the EER in FingerCode's matching results for the *Latest Small Random Distortions* tests and the EER for unperturbed images range between 0.49% (lrnddist 1.0) and 4.18% (lrnddist 4.2).

As already mentioned before, the matching results of the **Phase Only Correlation** matcher for fingerprint images of DB3 exhibit a decidedly larger impairment, caused by the generated perturbations introduced into the images, than they do for the tests on images of databases DB1 or DB2. Besides having the ROC plot depict a rather "unambiguous" image, also the equal error rates in Table 5.42 attest the comparatively strong impairment in POC's matching results: The EER corresponding to the least-effective test lrnddist 0.6 is already 1.43% higher, than POC's EER for unperturbed images of DB3. This is a level of impairment, that for example for the bozorth3 matcher – the least robust minutiae-based matcher in current tests – we only observe for test lrnddist 2.2. The Phase Only Correlation matchers's results for the successive tests then show, that with increasing intensity of the *Latest Small Random Distortions* perturbations introduced in the fingerprint images also the level of deterioration of the results grows. For the most effective test, lrnddist 4.2, we then accordingly establish an EER of 34.34%, which is sizable 19.27% higher, than the EER for the original, unperturbed fingerprint images.

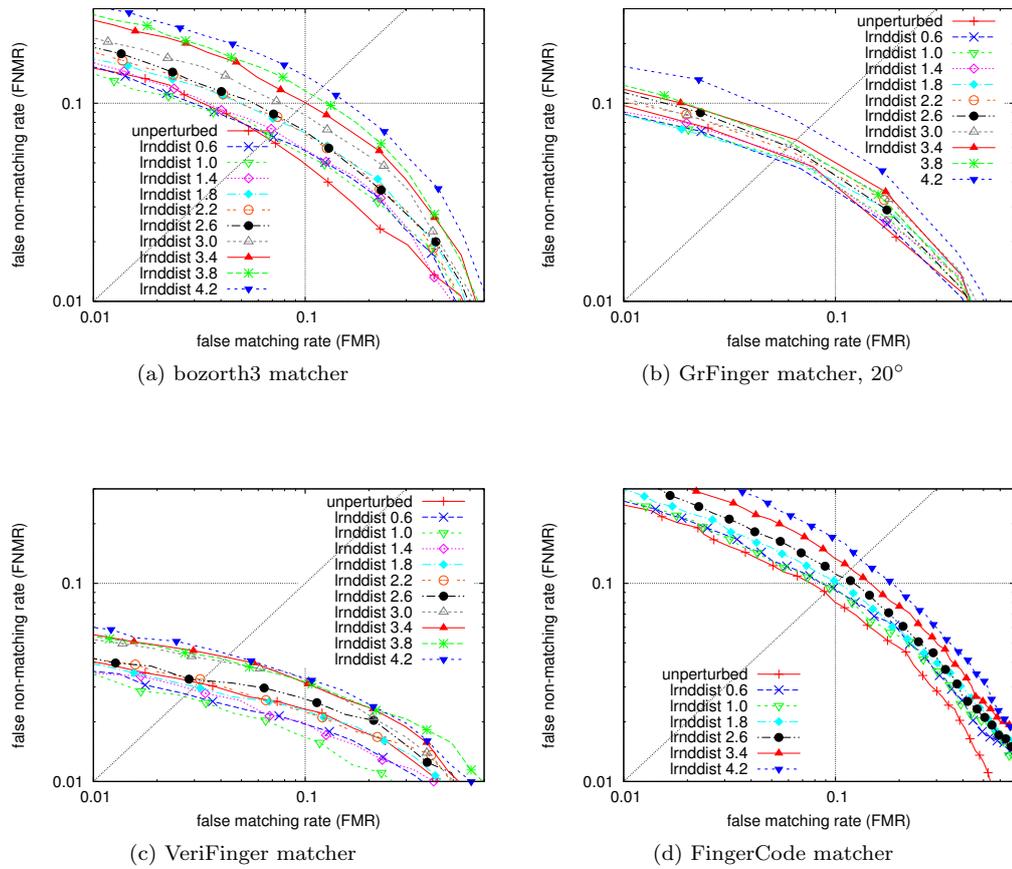


Figure 5.59: Receiver operating characteristics for the StirMark test *Latest Small Random Distortions* on fingerprint images of DB3. (Plots apply a logarithmic scale. Ranges are set to FMR [1%,70%] and FNMR [1%,30%]).

Factor	bozo3 (%)	VF (%)	GF180 (%)	GF20 (%)	FC (%)	POC (%)
unperturbed	6.68	3.60	6.90	5.82	8.98	15.07
0.6	6.87	3.26	6.20	5.46	9.56	16.50
1.0	6.86	3.13	6.68	5.59	9.47	17.21
1.4	7.24	3.05	6.98	5.79		
1.8	7.96	3.19	7.12	5.60	10.15	21.25
2.2	8.09	3.49	7.59	5.92		
2.6	8.20	3.63	7.30	6.14	10.76	25.21
3.0	9.18	4.10	7.49	6.17		
3.4	10.06	4.48	8.78	6.53	11.86	31.40
3.8	10.97	4.37	8.22	6.39		
4.2	12.21	4.63	9.82	7.84	13.16	34.34

Table 5.42: Equal error rates for *Latest Small Random Distortions* test conducted on sample image database DB3.

Chapter 6

Summary

In the introduction to this thesis, I have stated three major goals and posed several additional questions, which were intended, to be examined in course of my studies in present work. Consequently I will now also commence this summary by referring to these statements and thereby trying to get to conclusive answers, based on the experiences and findings gathered in the experiments.

The main goal was to *Investigate the influence of different kinds of natural perturbations in fingerprint images on the matching performance of diverse types of fingerprint matchers.*

At first I will briefly recapitulate the experiments that were conducted in this thesis: Five fingerprint matchers – three minutiae-based, one ridge-feature based and one correlation-based – were being employed. The fundamental test data was adopted from parts of the sample data of the Fingerprint Verification Contest 2004. It consisted of three individual sets, each containing 800 fingerprint images – 100 different fingers à 8 imprints. Each particular set had been acquired with a different type of fingerprint scanner, under varying finger- and positioning-related conditions. The “natural perturbations” (in terms of definition on page 6), that I set out to investigate, were simulated and introduced into the sample images, by the StirMark Benchmark, a tool that originates from the field of digital image watermarking robustness analysis. Thus, based on the initial three sample data sets, “new” sets of perturbed fingerprint images were created for altogether 12 different types of perturbations, with up to 10 varying levels of intensity per type. Using aforementioned five fingerprint matchers on the original, unperturbed fingerprint images, initial matching results were created, that determined each matcher’s fundamental matching performance on the sample images and henceforth served as basis of comparison. The thereby employed test procedure followed the regulations, defined in the Fingerprint Verification Contest 2004 (and 2006 alike), resulting in 7750 individual matching operations per test-run (2800 genuine tests and 4950 impostor tests). In the same manner, the experiments were then conducted on the five fingerprint matchers, using the various sets of perturbed fingerprint images. Per perturbation type, the results generated by a matcher were mainly analyzed in regard to two aspects: For one, “How does the matchers’ performance change over the various levels of perturbation-intensity, viewed in comparison to its initial performance for unperturbed images?” and second “How extensive is the manifestation of the particular perturbation in the matchers’s results, in comparison to that observed in the results of the other fingerprint matchers?”.

The detailed analysis and discussion of the experimental results for each type of simulated natural perturbation can be found in the respective sections in chapter 5. In present summary I want to take a more generalized view on the various findings:

One fundamental observation that we can state is, that the robustness of an individual fingerprint matcher to perturbations in fingerprint images, is very much dependent on two factors: the particular type of perturbations, as well as on the respective type of fingerprint images concerned: When regarding the results of all experiments conducted and comparing the individual matching performances with each other, we can establish, that in terms of robustness to the perturbation-related influences, none of the considered fingerprint matchers was able to continuously outperform the others (or at least a certain group thereof). On the contrary, for each matcher we can find at least one combination of perturbation type and fingerprint image type, for which its matching results are less affected by the respective perturbations, than the corresponding results of all other matchers. Likewise also exists at least one set of tests, for which the respective matcher is among the two most influenced matchers.

How strong a particular image type can affect a matchers susceptibility to the influences of the image perturbations, can best be seen in a concrete example: Regarding VeriFinger's matching performance in the *Additive Noise* tests (see section 5.3 for details), we find, that for fingerprint images of DB1, VeriFinger exhibits clearly the least sensitivity to the perturbations of all the matchers regarded. For fingerprint images of DB2 the opposite is true and VeriFinger generates the most-impaired results. Finally, for fingerprint images of DB3 then, VeriFinger shows once again a very high level of robustness. Additionally, for DB3 images we can also observe another very interesting aspect in the matcher's response to the image perturbations, which I will soon address in more detail: For almost all tested levels of additive noise, VeriFinger produces (slightly) better matching results, than it does for the original, unperturbed fingerprint images of DB3.

So, inspecting the results of the experiments, we are not able to establish a "global", generally applicable ranking of the fingerprint matchers, based on their specific robustness to the different kinds of perturbations in the various kinds of fingerprint images. Even when limiting the scope of the analysis to the particular types of fingerprint images (in our case, the single databases) we still find too high variations in the individual, perturbation-type dependent performances of the fingerprint matchers, as that we would be able to establish an order on the set of matchers, per image type. One statement, that we can make though, is, that for databases DB1 and DB3, the bozorth3 matcher and the GrFinger configuration GF180 never assume the rank of the most robust matcher, Yet on the other hand, for all three databases, they also never turn out to be the most impaired one.

Taking hitherto noted observations into account, for an overall summary it would thus be very desirable, to arrive at a tabulation, stating for each specific perturbation type in combination with each type of fingerprint images, which fingerprint matcher exhibits, in its matching results, the strongest influences caused by the perturbations, which matcher exhibits the second strongest influences and so on. However, the experiments show, that also at this level of detail, oftentimes it is not easily possible to establish such a clear ranking. When regarding the entirety of matching results, generated for a particular combination of perturbation type and fingerprint image type, we frequently find, that despite gradually increasing intensity of the perturbations across the tests within the test set, the amount of influence, these perturbations have on the fingerprint matchers' results, does not always grow accordingly, nor equally strong for every matcher. Mostly these variations happen on a small scale, from one test-level to the other – quasi: "in the first test matcher A appears stronger

influenced, in the second test matcher B, in the third test matcher A again...”. Still, we can also find examples where two matchers exchange “positions” rather continuously, over the whole set of tests – for instance in the *Latest Small Random Distortion* tests for fingerprint images of database DB2 (see also page 189), between matchers bozorth3 and GrFinger in configuration GF180: At first bozorth3’s matching performance is clearly less influenced by the perturbations than GF180’s. With increasing perturbation-intensity though, this relative distance gradually lessens, until from test-level lrnddist 3.0 on, GF180 then exhibits increasingly more robustness than bozorth3.

As we see, the experimental results make it quite difficult, to derive general conclusions about the fingerprint matchers’ sensitivity to the various perturbations, rather a detailed analysis is encouraged. Regarding more specific aspects however, we still get to a series of further interesting findings:

Starting off, for example, with a fairly broad comparison of the specific results of the different types of fingerprint matchers examined in this thesis – the ridge-feature-based FingerCode matcher, the correlation-based Phase Only Correlation matcher and the three minutiae-based fingerprint matchers bozorth3, VeriFinger and GrFinger. (Detailed information on the respective matcher types and on the individual matchers is given in chapter 4). In the following I will have to separate the analysis into the three sample image databases, to be able to make more distinct statements. Also, please bear in mind, that the non-minutiae-based fingerprint matchers were only tested on a reduced set of test-levels per StirMark test set, so the related findings presented, are also based on extrapolations of the existing data and assumptions about the further matching behavior:

DB1 The level of robustness to the perturbations in the fingerprint images of DB1, exhibited by the non-minutiae-based fingerprint matchers in the first four StirMark-tests presented, *Additive Noise*, *Median Cut Filtering*, *Mean Filtering* and *Modified Gauss Filtering* (please refer to sections 5.3 – 5.6 for details) is surprisingly high. While the matching results, generated by the set of minutiae-based fingerprint matchers, appear strongly impaired (especially for higher levels of perturbation-intensity), also the ROC plots confirm a comparatively low amount of perturbations-related influence in the results of the FingerCode matcher and further also the Phase Only Correlation matcher.

While the first four StirMark tests so far only affect the visual quality of a fingerprint image, all the other tests also affect the shape and orientation of the respective finger imprint. With that, the response of the non-minutiae based matchers changes drastically: While for the fifth test, *Remove Lines*, in those test-levels, that were also examined on the non-minutiae-based fingerprint matchers, FingerCode appears to be the least influenced matcher of all matchers regarded, the Phase Only Correlation matcher now exhibits the highest amount of influence in its matching results. In the *Rotations* test, only the GrFinger configuration GF20 shows a worse performance, than the non-minutiae-based matchers. In the subsequent four StirMark tests, which employ affine transformations to generate the perturbations, POC is once again the most influenced fingerprint matcher, while FingerCode competes quite well with the individual minutiae-based matchers. Finally, in the StirMark tests, *Small Random Distortions* and *Latest Small Random Distortions*, the set of minutiae-based matchers clearly shows more robustness to the perturbations in the fingerprint images of DB1, than any of the non-minutiae-based fingerprint matchers.

DB2 For the first two perturbation types, both non-minutiae-based fingerprint matchers

once again exhibit the highest level of robustness to the perturbations in the fingerprint images of DB2 – in the *Additive Noise* test the FingerCode matcher appears less influenced, in the *Median Cut Filtering* test the Phase Only Correlation matcher. In the following two tests, based on convolution filtering, POC generates still the least impaired results of all regarded fingerprint matchers, while FingerCode competes with VeriFinger for the rank of the most-influenced matcher. Next, we get to the tests that, as mentioned before, affect the shape and orientation of the finger imprints. Accordingly henceforward, compared to all the other matchers, the Phase Only Correlations matcher constantly displays the greatest amount of perturbations-related influence in its matching results – with one exception: the *Rotations* test, where again the GrFinger configuration GF20 appears clearly more sensitive to the deliberately rotated fingerprint images, than any other matcher. As for the FingerCode matcher, in the *Remove Lines* test, together with VeriFinger, it exhibits the comparatively highest robustness to the perturbations. For the remaining tests though (with exception of the *Rotations* test) the relative level of impairment in its matching results lies generally between that of the minutiae-based matchers and that of the POC matcher.

DB3 Already for the first StirMark test, *Additive Noise*, the overall situation presents itself somewhat different, than we witnessed it for fingerprint images of DB1 and DB2: The matcher, the least affected by the additional random noise in the images of DB3 is VeriFinger, followed by the Phase Only Correlation matcher. FingerCode, on the other hand, turns out to be the matcher, whose matching results are the strongest impaired. In the next two StirMark tests, *Median Cut Filtering* and *Mean Filtering*, FingerCode stays the matcher, the most influenced by the perturbations, while POC's matching results once exhibit the highest level of robustness and once a rather average one. In all of the following StirMark tests though, the Phase Only Correlation matcher shows clearly the highest sensitivity to the perturbation-related influences of all matchers regarded – with exception of the *Rotations* test, where once again GF20 appears the most influenced. Like was the case for fingerprint images of DB1 and DB2, FingerCode's matching results for the *Remove Lines* test are, like those of VeriFinger, only slightly affected by the perturbations in the fingerprint images. For the tests *Modified Gauss Filtering* and *Rotations* then, FingerCode's level of robustness is generally comparable to that of the minutiae-based matchers (except, of course, GF20 in the *Rotations* test). However, for the last six perturbation types then – the four, generated using affine transformations and the two “original” StirMark tests – FingerCode's sensitivity to the perturbations lies again between the levels witnessed in the minutiae-based fingerprint matchers and that witnessed in the Phase Only Correlation matcher.

One thing, that we can state in regard to both non-minutiae-based matchers: In those test-sets, where they appear more susceptible to the image perturbations, than the entirety of the minutiae-based matchers, the differences in the respective impairment, found in the matching results, are almost always very distinct – for the Phase Only Correlation matcher even more so, than for the FingerCode matcher. For example the degradation observed in the respective EER rates for the non-minutiae-based matchers is up to several times stronger, than that in the EER rates for the minutiae-bases matchers. Also in the related ROC plots, the differences in the relative separations of the ROC curves, which correspond to the various test-levels, is clearly perceptible.

Another observation, that we can make in the experimental results, concerns the two configurations of the fingerprint matcher GrFinger, that we regarded in this thesis - GF20, which only considers a limited angular range of $\pm 20^\circ$ for its rotational alignment procedure

and GF180, which considers an angular range of $\pm 180^\circ$. When comparing the matching performances, both configurations exhibited in the 36 individual test-sets employed in the experiments (12 perturbation types \times 3 fingerprint data sets), we find, that in 29 cases GF20 turns out, to be more robust to the influences of the image-perturbations than GF180. As for the remaining 7 cases, there is just one single perturbations type, for which GF180 appears less influenced than GF20 for all three fingerprint image types: the *Rotations* tests – which does not come as a big surprise. One thing, that is also interesting in respect to GF20’s performance in the *Rotations* tests though, is, how clearly more impaired GF20’s matching results are, in comparison to those of any other matcher regarded – especially to those of the non-minutiae-based matchers, as their rotational alignment functionality was likewise limited to the angular range of $\pm 20^\circ$ (for details please refer to section 5.8).

As to why GF20 performs so different from GF180 and especially, in most of the cases performs better than GF180, I can merely make assumptions, as no declarative information concerning this matter could be found in GrFinger’s documentation nor do the results gathered in present experiments suffice, to draw significant-enough conclusions about the specific functionalities of the matcher. One plausible explanation for GF20 behavior would be, that with the limitation of the angular range, that GF20 inspects, to find the best matching set of minutiae, it is also less prone (“has less chances”), to find misleading minutiae-combinations, that it might ultimately identify as actual match.

A further interesting observation, that we can make in individual matching performances in most of the employed StirMark test-sets, is, that occasionally fingerprint matchers achieve better matching results for perturbed images, than they initially do for the corresponding original, unperturbed fingerprint images. For certain perturbation types, this behavior is quite easily understandable – for example, when regarding the *Modified Gaussian Filtering* tests on fingerprint images of database DB3: As noted in the corresponding discussion of the results on page 119, for this particular type of fingerprint images, the *Modified Gaussian Filtering* perturbations lead to a certain “clearing” of the finger imprints, making the ridge-structure better perceptible. Hence one would expect, that it should be easier for the fingerprint matchers to match such “perturbed” images and at least the results of VeriFinger, both GrFinger configurations and FingerCode clearly confirm this assumption. On the other hand, there are several examples, for which the observed improvement in matching performance is definitely harder to comprehend. A quite explicit example therefor, that was also mentioned earlier, can be found in VeriFinger’s matching performance in the *Additive Noise* test for fingerprint images of DB3: For almost all test levels VeriFinger produces better matching results than for the unperturbed images. Even though the relative differences are comparatively small – regarding the corresponding equal error rates in Table 5.5, we find, that differences between the EER relating to the individual noise levels and the EER for the unperturbed images range between 0.06% and 0.7% – per se it seems to be against logic, that random noise introduced into fingerprint images, improves the results of a fingerprint matcher. At this point further experiments would be necessary, in order to find the reasons for this matching behavior.

Overall, in the results of the experiment of present work, we are able to observe a broad variety of aspects, regarding the reactions of the diverse fingerprint matchers and matcher types in respect to the individual kinds of perturbations in the different types of fingerprint images. However, I have to state, that a genuine understanding, of why a particular fingerprint matcher or matcher type reacts to a certain test-setup (fingerprint image type, perturbation type and perturbation intensity) the way it does, is beyond of what can be establish in the experimental results and thus would be subject for further in-depth investigations.

Now let us consider the next goal defined in present thesis: *Evaluate in general, if the experiment procedures applied, could form the basis for a common fingerprint matcher comparison and analysis benchmark.*

When inspecting the experimental results, we find, that already with this first try for a benchmark setup, we are actually able to identify differences in how the individual fingerprint matchers respond to the various types of perturbations – as already noted on several occasions, we can observe, how for a single perturbation type, the respective amount of influence in a matcher’s results changes with increasing perturbation-intensity, we can further compare these observations between the various regarded fingerprint matchers and we are even able to derive certain statements concerning the specific types of fingerprint matchers. So overall it is possible to gather a broad range of information, at various levels of detail.

Thus I can state, that yes, I am very convinced, that the experimental approach, employed in present thesis, would serve well as basis for a common generic benchmark architecture for comparison and analysis of fingerprint matchers. A benchmark, that to my believe, would enable a user, to systematically examine the matching performances of individual fingerprint matchers and further establish their particular strong points and weaknesses in regard to their ability to cope with certain kinds of perturbations in fingerprint images.

The results, generated with this benchmark, could then, for example, be used to make informed decisions about which fingerprint matcher best suites the specific needs of particular application scenarios. Further, one might also think about combining two different types of matching algorithms to compensate specific weaknesses - one could choose two matchers/matcher-types as such, that one is robust to those perturbation types, to which the other one is sensitive and vice versa.

Finally, let us now turn to the last remaining goal, defined for the experiments of present thesis: *Determine the applicability of the well-known digital image watermarking robustness-testing tool StirMark Benchmark for simulation and induction of natural perturbations.*

The particular reasons for employing the StirMark Benchmark in present studies have already been stated on several occasions and are further summarized in section 2.3.

As the StirMark Benchmark itself was developed with the intention, to establish a standard for robustness-tests in the field of digital image watermarking and to provide the architecture for a corresponding generic benchmark, as such it already features a broad range of aspects, that make it very well suited for a benchmark architecture compliant to our requirements (please refer to section 2.3 for details). Further the StirMark Benchmark provides a series of image manipulation operations, that, as is also established in the individual StirMark test related sections in chapter 5, can very well be employed to simulate a variety of image-altering effects, that can be related to perturbations actually occurring in fingerprint images of real-life application scenarios. Moreover, via the set of given parameters we are able to precisely determine individual characteristics and the intensity of the perturbations to be simulated and thus have the possibility, to tailor the generated perturbed fingerprint images, to match the specific objectives we would like to investigate.

Still, ultimately I deem the StirMark Benchmark unfit for application in connection with a benchmark for evaluating fingerprint matching performances. This verdict is due to a single reason, which, as important as it is for the original purpose of the StirMark Benchmark, is as troublesome to our intended application: Its function, which automatically embeds a certain pre-defined watermark into the processed images. As established in section 5.2, the watermark itself, introduced in the fingerprint images, most likely does not have any

effect on our experimental results (despite the fact, that we could actually nullify the watermark of even adjust it to our needs). However, as we determined, said functionality also slightly increases the brightness of the processed images. Thereby it not only severely limits the possibility to combine several perturbation types, by repeated application of individual StirMark operations to the fingerprint images, but more important, this also influences the actual matching results of the individual fingerprint matchers. The effect is comparatively small (see results in section 5.2), still it distorts the perception of the effectively achievable matching performances in a very unpredictable manner, thus it can not be tolerated in the intended common fingerprint matching benchmark.

Thus for the simulation of natural perturbations in fingerprint images, in our case we would need a tool, that for the most part provides the same functionalities as the StirMark Benchmark, yet takes great precautions, to avoid introducing any unintended effects into the processed images.

Further it would be very interesting, to extend the amount of available image manipulation operations by perturbations, that have been specifically designed, to simulate fingerprint-application typical challenges in a given set of fingerprint images – for example a test, that simulates residues from ridge-lines of “previously” scanned fingerprints. Thereby it might be possible, to establish even more distinct, even more application-specific differences in the matching performances of various fingerprint matchers. Which particular problems these perturbation types should cover and how to simulate the desired effect, is of course subject to further investigations.

Bibliography

- [1] R. Bolle, J. Connell, S. Pankanti, N. K. Ratha, and A. Senior, *Guide to Biometrics*. Springer Verlag, 2003.
- [2] Q. Chen, M. Defrise, and F. Deconinck, “Symmetric phase-only matched filtering of Fourier-Mellin transforms for image registration and recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 12, pp. 1156–1168, 1994.
- [3] J. G. Daugman, “Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters,” *Journal of the Optical Society of America*, vol. 2, no. 7, pp. 1160–1169, July 1985.
- [4] —, “High confidence visual recognition of persons by a test of statistical independence,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 11, pp. 1148–1161, 1993.
- [5] G. de Sá and R. de Alencar Lotufo, “Improved fingercod matching function,” in *SIB-GRAPI*, 2006, pp. 263–272.
- [6] J. Fierrez-Aguilar, Y. Chen, J. Ortega-Garcia, and A. K. Jain, “Incorporating image quality in multi-algorithm fingerprint verification,” in *Proc. IAPR Intl. Conf. on Biometrics, ICB*, ser. LNCS, vol. 3832. Springer Verlag, 2005, pp. 213–220.
- [7] “Fingerprint Verification Competition 2004.” [Online]. Available: <http://bias.csr.unibo.it/fvc2004>
- [8] “Fingerprint Verification Competition 2006.” [Online]. Available: <http://bias.csr.unibo.it/fvc2006>
- [9] Griaule Biometrics, “Fingerprint SDK 2009,” 2009. [Online]. Available: http://www.griaulebiometrics.com/page/fingerprint_sdk
- [10] N. Hiroshi, K. Koji, K. Masayuki, A. Takafumi, and H. Tatsuo, “Pattern collation apparatus based on spatial frequency characteristics,” United States Patent #5,915,034, June 1995.
- [11] L. Hong, Y. Wan, and A. K. Jain, “Fingerprint image enhancement: Algorithm and performance evaluation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 777–789, Aug. 1998.
- [12] A. K. Jain, L. Hong, and R. Bolle, “On-line fingerprint verification,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 302–314, 1996.

- [13] A. K. Jain and D. Maltoni, *Handbook of Fingerprint Recognition*. Secaucus, NJ, USA: Springer Verlag New York, Inc., 2003.
- [14] A. K. Jain, S. Prabhakar, and L. Hong, "A multichannel approach to fingerprint classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 4, pp. 348–359, 1999.
- [15] A. K. Jain, S. Prabhakar, L. Hong, and S. Pankanti, "Filterbank-based fingerprint matching," *IEEE Transactions on Image Processing*, vol. 9, no. 5, pp. 846–859, 2000.
- [16] A. K. Jain, A. Ross, and S. Prabhakar, "Fingerprint matching using minutiae and texture features," in *ICIP*, 2001, pp. 282–285.
- [17] K. Karu and A. K. Jain, "Fingerprint classification," *Pattern Recognition*, vol. 29, no. 3, pp. 389–404, 1996.
- [18] T. Kenji, A. Takafumi, S. Yoshifumi, H. Tatsuo, and K. Koji, "High-accuracy subpixel image registration based on phase-only correlation," *IEICE transactions on fundamentals of electronics, communications and computer sciences*, vol. 86-A, no. 8, pp. 1925–1934, 2003. [Online]. Available: <http://ci.nii.ac.jp/naid/110003221273/en/>
- [19] I. Koichi, N. Hiroshi, K. Koji, A. Takafumi, and H. Tatsuo, "A fingerprint matching algorithm using phase-only correlation," *IEICE Transactions on Fundamentals*, vol. E87-A, no. 3, pp. 682–691, Mar. 2004.
- [20] M. Kutter and F. A. P. Petitcolas, "A fair benchmark for image watermarking systems," in *Proceedings of the 11th SPIE Annual Symposium, Electronic Imaging '99, Security and Watermarking of Multimedia Contents*, vol. 3657, San Jose, CA, USA, Jan. 1999.
- [21] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, and A. K. Jain, "FVC2004: Third Fingerprint Verification Competition," in *ICBA*, ser. LNCS, vol. 3072. Springer Verlag, 2004, pp. 1–7.
- [22] "Pattern Recognition and Image Processing Laboratory," Michigan State University. [Online]. Available: <http://biometrics.cse.msu.edu/>
- [23] "NIST Biometric Image Software," National Institute of Standards and Technology, 2010. [Online]. Available: <http://fingerprint.nist.gov/NBIS/>
- [24] Neurotechnology, "VeriFinger SDK 6.2," 2010. [Online]. Available: <http://www.neurotechnology.com/verifinger.html>
- [25] F. A. P. Petitcolas, "StirMark Benchmark 4.0." [Online]. Available: <http://www.petitcolas.net/fabien/watermarking/stirmark/>
- [26] —, "Watermarking schemes evaluation," *IEEE Signal Processing*, vol. 17, no. 5, pp. 58–64, Sept. 2000.
- [27] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Attacks on copyright marking systems," in *Information Hiding*, ser. LNCS, vol. 1525. Springer Verlag, 1998, pp. 218–238.

- [28] F. A. P. Petitcolas, M. Steinebach, F. Raynal, J. Dittman, C. Fontaine, and N. Fates, "A public automated web-based evaluation service for watermarking schemes: StirMark benchmark," in *Electronic Imaging 2001, Security and Watermarking of Multimedia Contents*, ser. Proceedings of the SPIE, vol. 4314. San Jose, CA, U.S.A.: Society for Imaging Science and Technology (IS&T) and International Society for Optical Engineering (SPIE), Jan. 2001, pp. 575–584, ISSN 0277-786X.
- [29] N. K. Ratha, S. Chen, and A. K. Jain, "Adaptive flow orientation based feature extraction in fingerprint images," *Pattern Recognition*, vol. 28, pp. 1657–1672, Nov. 1995.
- [30] A. Ross, A. K. Jain, and J. Reisman, "A hybrid fingerprint matcher," *Pattern Recognition*, vol. 36, no. 7, pp. 1661–1673, 2003.
- [31] A. Ross, J. Reisman, and A. K. Jain, "Fingerprint matching using feature space correlation," in *Biometric Authentication*, ser. LNCS, vol. 2359. Springer Verlag, 2002, pp. 48–57.
- [32] "Biometric Test Center," San Jose State University. [Online]. Available: <http://www.engr.sjsu.edu/biometrics>
- [33] R. Santos, "Java Advanced Imaging Stuff." [Online]. Available: <https://jaistuff.dev.java.net/>
- [34] "Biometrics Research Lab - ATVS," Universidad Autonoma de Madrid. [Online]. Available: <http://atvs.ii.uam.es/>
- [35] "Biometric System Laboratory," University of Bologna. [Online]. Available: <http://biolab.csr.unibo.it/>
- [36] C. I. Watson, M. D. Garris, E. Tabassi, C. L. Wilson, R. M. McCabe, S. Janet, and K. Ko, *User's Guide to NIST Biometric Image Software Export Control (NBIS-EC)*, National Institute of Standards and Technology. [Online]. Available: http://fingerprint.nist.gov/NBIS/nbis_export_control.pdf
- [37] —, *User's Guide to NIST Biometric Image Software (NBIS)*, National Institute of Standards and Technology. [Online]. Available: http://fingerprint.nist.gov/NBIS/nbis_non_export_control.pdf
- [38] "Java Advanced Imaging (JAI) API." [Online]. Available: <http://java.sun.com/javase/technologies/desktop/media/jai/>
- [39] "Programmer's Guide: Programming in Java Advanced Imaging." [Online]. Available: http://java.sun.com/products/java-media/jai/forDevelopers/jai1_0_1guide-unc/index.html
- [40] "teneighty.org - Fast Fourier Transform Package." [Online]. Available: <http://www.teneighty.org/software/index.html?f=fft&c=e98b8>
- [41] "ImageMagick." [Online]. Available: <http://www.imagemagick.org/>