ABSTRACT

Title of dissertation: AUTHENTICATION OF
FINGERPRINT SCANNERS

Vladimir Iankov Ivanov
Doctor of Philosophy, 2012

Dissertation directed by: Professor John S. Baras
Department of Electrical
and Computer Engineering

To counter certain security threats in biometric authentication systems, particularly in portable devices (e.g., phones and laptops), we have developed a technology for automated authentication of fingerprint scanners of exactly the same type, manufacturer, and model. The technology uses unique, persistent, and unalterable characteristics of the fingerprint scanners to detect attacks on the scanners, such as detecting an image containing the fingerprint pattern of the legitimate user and acquired with the authentic fingerprint scanner replaced by another image that still contains the fingerprint pattern of the legitimate user but has been acquired with another, unauthentic fingerprint scanner. The technology uses the conventional authentication steps of enrolment and verification, each of which can be implemented in a portable device, a desktop, or a remote server. The technology is extremely accurate, computationally efficient, robust in a wide range of conditions, does not require any hardware modifications, and can be added (as a software add-on) to systems already manufactured and placed into service. We have also implemented the technology in a demonstration prototype for both area and swipe scanners.

AUTHENTICATION OF FINGERPRINT SCANNERS

by

Vladimir Iankov Ivanov

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2012

Advisory Committee:
Professor John Baras, Chair/Advisor
Professor Jonathan Simon
Professor Gang Qu
Professor Carol Espy-Wilson
Professor Radu Balan, Dean's Representative

Dedication

To the memory of my father

Acknowledgments

First and foremost I would like to thank my advisor, Professor John Baras, for his vision, support, and mentorship. His immense energy and enthusiasm have always inspired me and gave me the confidence to aspire high and challenge seemingly unsolvable problems. His diverse interests and expertise made me look beyond by technical background and venture into completely new for me areas. A critical ingredient for the success of this project was the freedom I had to try and to fail. Initially very difficult, this eventually made me develop leadership skills and improve my organization, something I believe would have never happened in a deterministic, all-guided setting. And yet even more importantly, Professor Baras gave me the rarely offered opportunity to actually solve real problems in a practical way – to do the research, then design the algorithms, and finally implement them in a working system.

I would like to also thank Professor Jonathan Simon for the insightful discussions we had, for his valuable comments, both at the time of the PhD research proposal and the dissertation defense, and for the extremely careful review of the manuscript and his suggestions for improving it. I am also grateful to Professor Gang Qu, Professor Carol Espy-Wilson, and Professor Radu Balan for serving on my committee, for their critical comments and challenging questions, and for reviewing this rather voluminous document.

Many thanks go Mrs. Kimberly Edwards who was a dependable collaborator throughtout all my study and work and on whom I relied for my most critical and urgent needs, be it with conferences, travel arrangements, or purchasing various equipment for my lab. This has saved me precious time when I needed it most and I greatly appreciate it. Perfectly organized, responsive, and patient, she always made time for me, giving me advice and helping me even on short notice and over weekends.

I am very grateful for the financial support for my graduate studies and for my research provided by the U.S. Army Research Laboratory under the Collaborative Technology Alliance Program via Cooperative Agreement DAAD19-01-2-0011; by the U.S. Army Research Office award MURI W911-NF-0710287; by the University of Maryland at College Park and the University of Maryland Foundation; by the Department of Electrical and Computer Engineering at the University of Maryland; by the Maryland Center for Hybrid Networks (HyNet) (previously known as the Center for Satellite and Hybrid Communication Networks (CSHCN)) and the Institute for Systems Research at the University of Maryland; and by the National Aeronautics and Space Administration (NASA) via Cooperative Agreement NCC8-235. Any opinions, findings, and conclusions or recommendations expressed in this dissertation, however, are mine and should not be interpreted as reflecting the official views or policies of the U.S. Army Research Laboratory, the U.S. Army Research Office, the University of Maryland at College Park, or the National Aeronautics and Space Administration.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ACF	Autocorrelation Function
CDF	Cumulative Distribution Function
DFT	Discrete Fourier Transform
EDF	Empirical Distribution Function
EER	Equal Error Rate
FAR/FMR	False Acceptance/Match Rate
FPN	Fixed-pattern Noise
FRR/FNMR	False Rejection/Non-Match Rate
IQR	Interquartile Range
MTF	Modulation Transfer Function
PCA	Principal Component Analysis
PDF	Probability Density Function
PSD	Power Spectral Density
PRNU	Photo-response Non-uniformity (Noise)
ROC	Receiver Operating Characteristic
ROI	Region of Interest
SFR	Spatial Frequency Response
STD	Standard Deviation
SVM	Support Vector Machines
TC	Trusted Computing
TPM	Trusted Platform Module

Chapter 1

Introduction

Authentication verifies the claim about the identity of an entity. Biometric technologies measure unique personal characteristics which can be used to identify individuals with a high degree of certainty and thus have the potential to certify the connection between people and the systems they are authorized to use.

Using biometrics for authentication of people to systems provides convenience. When authenticating to portable devices, such as smartphones and laptops, however, security problems may arise because this authentication usually takes place in unsupervised environments (e.g., at home). Since a portable device can be easily stolen, an attacker with physical access to it can launch a powerful attack by manipulating the data which is acquired and transmitted by the biometric scanner. Furthermore, the biometric information has a low degree of secrecy as it can be captured by an unintended recipient and even without user's consent. Since the biometric characteristics are difficult to change and cannot be revoked, their compromise may lead to more serious consequences than, for example, a compromise of a password. Finally, regardless of all effort to keep user's biometrics private, the widespread use of biometric technologies are set to make the biometric information essentially publicly available, with the face photos being public even today.

To counter some of these security threats, we have developed a technology for automated authentication of fingerprint scanners. Fingerprint scanners have unique patterns that can be used to distinguish one scanner from another one. The pattern, which we call *scanner pattern*, stems from the variability of device characteristics at silicon level and is caused by imperfections of the conversion from the input to the scanner (i.e., the object applied to it) to its output (i.e., the digital image). The scanner pattern is a sufficiently unique, persistent, and unalterable intrinsic characteristic of the fingerprint scanners even to those of exactly the same technology, manufacturer, and model. Our technology is able to distinguish the pattern of one scanner from the pattern of another scanner of exactly the same model by extracting the pattern from a single image, acquired with each scanner.

In this way, the scanner pattern can be used to enhance the security of a biometric system by authenticating the scanner, used to acquire a particular fingerprint image, and thus detect attacks on the scanner, such as detecting an image containing the fingerprint pattern of the legitimate user and acquired with the authentic fingerprint scanner replaced by another image that still contains the fingerprint pattern of the legitimate user but has been acquired with another, unauthentic fingerprint scanner. The technology uses the conventional authentication steps of enrolment and verification, each of which can be implemented in a portable device, a desktop, or a remote server. The technology is extremely accurate, computationally efficient, robust in a wide range of conditions, does not require any hardware modifications, and can be added (as a software add-on) to systems already manufactured and placed into service. We have also implemented the technology in a demonstration prototype for both area and swipe fingerprint scanners.

Combining the biometric authentication with a scanner authentication leads to a two-part authentication, which we call *bipartite authentication*, that verifies both the identity of the user and the “identity” of the fingerprint scanner. Therefore, we see the main, although not necessarily the ultimate, application of this technology as a method to improve the security of portable devices with an additional layer. Strong security is required to gain the confidence of the users in their devices as secure universal terminals for online access to virtually all consumer services: from bank applications to mobile commerce to access to health care anywhere and at any time, for access to medical records, etc.

Chapter 2

Motivation and Background

2.1 Biometric Authentication and Biometric Systems

The objective of authentication of people is establishing their identity. The three general methods for authentication are based on: (1) what you know, e.g., a password, (2) what you have, e.g., a token, and (3) what you are. In the third method, the identification is based on person's physiological or behavioral traits, called collectively *biometrics*. Using biometrics for identification is not new — it has been around for centuries. Today biometrics is also used for authentication to systems and automated verification of identity.

The advantages of using biometrics for authentication over the first two methods are [Jain et al. 2006]:

- Biometrics cannot be lost or forgotten;
- Biometrics are difficult to copy, share, and distribute;
- Biometrics are difficult to forge;
- Biometric authentication requires presence of the person at the time and point of authentication.

Unfortunately, today's electronic technologies to great extent diminish these advantages and even create other types of challenges.

The conceptual diagram of a generic system that uses biometrics, as specified by [ISO/IEC SD 11], is shown in Figure 2.1. Although the figure describes a typical biometric system for authentication, its subsystems and processes are common for other applications that use biometrics. These conceptual subsystems need not correspond to physical subsystems in a real biometric system.

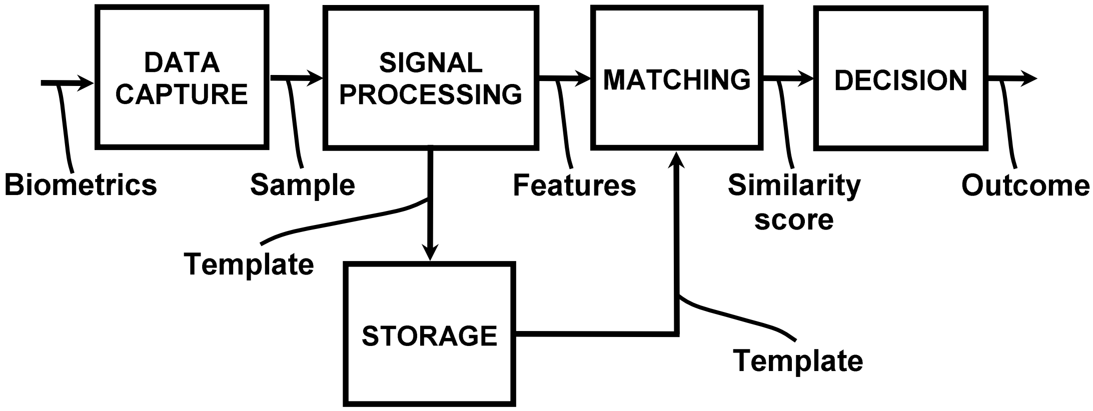


Figure 2.1: Generic biometric system

The Data Capture Subsystem (see Figure 2.1) takes a signal/image of the biometric identifier that the individual has presented to the biometric sensor and produces a biometric sample. This sample may be further compressed and/or encrypted and transmitted to the Signal Processing Subsystem. The latter extracts the distinguishing features from the sample, which may include segmentation, feature extraction, and quality control. In case of enrollment, it also creates one or more templates that may require several presentations. The template can be stored in the Storage Subsystem, within a biometric capture device, on a portable medium (e.g., in a smartcard), or on a personal computer/local server. The Matching Subsystem compares the features against one or more templates and outputs similarity scores that measure the degree of similarity between them. Based on the scores, on a pre-defined threshold, and on the type of operation (verification or identification), the Decision Subsystem produces a decision outcome. It is also possible to have a multi-biometric system that uses different biometric identifiers (as samples and templates), where the separate scores are combined and evaluated by the Decision Subsystem.

A Secure Biometric System, as defined in [INCITS M1/06–0424], may also include defense against biometric sensor attacks, biometric sample modification and/or injection attacks, a liveness detection to make sure that the captured biometric sample comes from a live person, etc.

The subsystems of a biometric system may be implemented in different physical systems: central/distributed (server), local (client), device (peripheral), and on-token. Depending on the location of the template storage and the location of the biometric match operations, [INCITS M1/06–0424] defines several architectural configurations. Of interest for us are "store on device/match on device," "store on token, match on device," and "store on token/match on token."

Due to several factors (discussed in section 2.2.2), rarely are the feature sets extracted from two biometric samples the same. The variability in the feature set of a person is referred to as *intra-class* variation, and the variability between the features sets of two different persons is referred to as *inter-class* variation [Jain et al. 2008]. Thus, a design objective is to find a feature set with small intra-class variation and large inter-class variation.

The event when the similarity score between the feature sets taken from two different persons exceeds the threshold and the Decision Subsystem produces a positive decision (match) is referred to as *false accept*. The probability of this event is termed *False Accept Rate (FAR)*. On the other hand, the event when the similarity score between the features sets taken from one and the same person is below the threshold and the Decision Subsystem produces a negative decision (non-match) is referred to as *false reject*. The probability of this event is termed *False Reject Rate (FRR)*. There is a specific and intrinsic tradeoff between FAR and FRR, and the function between the two is the *Receiver Operating Characteristic (ROC)*. The ROC is a primary performance metric of every biometric system [Jain et al. 2008]. The error rate when FAR = FRR is called *Equal Error Rate (FRR)*.

2.2 Biometric Identifiers and Their Characteristics

2.2.1 Biometric Identifiers

A physiological and/or behavioral characteristic that is sufficiently universal, distinct, permanent, collectable, and acceptable to be collected can be used as a biometric identifier [Maltoni et al. 2003]. Besides that, it is important that the biometric system, using this identifier, have high performance and be difficult to circumvent.

The biometric identifier of interest to our research is the fingerprint, which characteristics are summarized next. Other physiological biometric identifiers are the face, iris, hand geometry, palm vein patterns, DNA, retinal scans, ear shape, and body odor. Written signatures, voice, and typing patterns are typical behavioral biometrics ([NIST SP 800–32] and [Jain et al. 2006]).

Fingerprints

There are several reasons for choosing the fingerprint as the biometrics of primary interest in our research. The fingerprint systems have a very good balance between the desirable properties for biometric identifiers and systems as described above ([Maltoni et al. 2003, Jain et al. 2006]).

- Fingerprints are highly distinct and their pattern develops early in life;
- Fingerprint details are permanent. The features used by most matching algorithms are minutiae – the points of ridge and valley endings, lakes, points, and spurs;
- The use of fingerprints for identification of people is century-old long and quite mature, although it has been first systematized for law enforcement;
- Live-scan scanners can acquire high-quality fingerprint images;
- Low cost and small-sized implementations are available. This is a very important condition for their wide deployment, especially in portable electronic devices.

Nevertheless, changes in the environment (e.g., moisture), cuts and bruises, and changes due to ageing still pose certain challenges to the fingerprint technologies. Furthermore, the fingerprint algorithms are computationally intensive.

A typical fingerprint with its minutiae is shown in Figure 2.2.

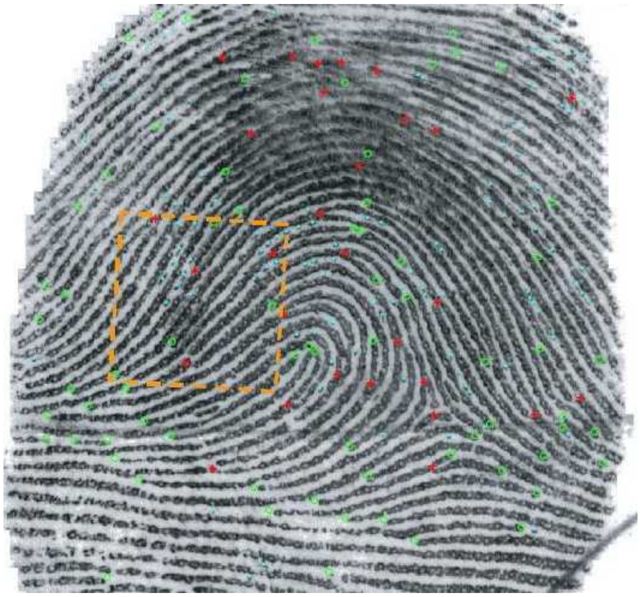


Figure 2.2: Fingerprint and its minutiae

2.2.2 Characteristics of Biometrics

2.2.2.1 Biometric Variance

In contrast to a password or a PIN code, the biometric information is not exact — it varies as a result of measuring a physical object (still or moving). Different captures of the biometrics of the same individual produce different biometric samples which, very often, produce different, however slightly, feature sets. The groups of factors that render the biometric information not uniquely repeatable are three [Jain et al. 2006]:

A. **Inconsistent presentation:** inconsistent user interaction with the sensor. For example, the 3D finger is projected onto the 2D surface of the sensor, the finger is not a solid object and therefore it gets deformed during this mapping, and each biometric acquisition may capture different parts of the finger;

B. **Irreproducible presentation:** changes in the biometric identifiers over time which, for example, can be due to aging, wear-and-tear, injuries, and pathological developments;

C. **Imperfect image/representational acquisition** such as imperfections and noise in the biometric sensor, nonuniform contact, environmental conditions (moisture, temperature, dirt), different illumination, imperfect feature extraction, etc.

The biometric variance results in imperfect user identification and authentication (FAR and FRR as discussed in Section 2.1).

2.2.2.2 Biometrics and Security

Besides its variance, using biometrics in security applications faces additional challenges because the biometric information:

- *has a low degree of secrecy*, i.e., it is difficult to be kept strictly private. [NIST SP 800–63] states that "biometrics do not constitute secrets suitable for use in the conventional remote authentication protocols;"

- *is not easily changeable.* Unlike passwords and PIN codes, once compromised, most biometrics cannot be changed (without surgical intervention) or be revoked;

- *can be counterfeited.* In unattended environments, an attacker can use models of the genuine biometric identifier, can make several attempts, or can even physically violate the integrity of the biometric sensor in order to substitute the legitimate information [INCITS M1/06–0424].

The low degree of secrecy of the biometric information was demonstrated in a protest by the Chaos Computer Club (a hackers organization) against the increasing use of biometrics. In the March 2008 edition of its magazine, the club published the fingerprint of the German Minister of the Interior (Home Secretary) Wolfgang Schäuble, warning that the fingerprints are not as safe as politicians claim and that "they should not be part of any critical security application." The hackers further included a thin film that can be taped over someone's finger to deceive fingerprint readers with Schäuble's fingerprint; they even created dummy fingers from his fingerprint [Heise Online 2008].

Even when the enrolled biometric templates are encrypted and stored in secure databases, function creep and owner abuse cannot be ignored as security threats. Furthermore, illegal access to private information in such "secure databases" is becoming a serious problem. Privacy Rights Clearinghouse estimated that since 2005 only in the US, over 500 million records containing personal information such as Social Security numbers, account numbers, and driver's license numbers, have been compromised (stolen or exposed) due to security breaches [PR Clearninghouse 2010]. Finally, an investigation [Robertson 2009] by The Associated Press revealed that "banks and other companies that handle your information are not being nearly as cautious as they could," which results in gambling with your personal data once you pay with a credit card. Under such circumstances, people's mistrust in the ability of systems and networks to protect their confidential information is completely justified.

In summary, regardless of all effort to keep user's biometrics private, the widespread use of biometric technologies are set to make the biometric information essentially publicly available, with the face photos being public even today.

Using biometric information for authentication, however, can provide several advantages over the systems that use other means of user identification because:

- Biometric samples contain more information than a PIN or a password that is does not need to be memorized and can be used for security purposes [INCITS M1/06–0424];
- Biometric information cannot be "guessed" in a dictionary attack; even brute force attacks are difficult;
- In conventional security systems, once the secret element is compromised, the methods using username–passcode allow the attacker to receive full access *and* cannot provide protection against repudiation and impersonating attacks because binding the identity with the claimant is not possible [INCITS M1/06–0424].

2.3 Biometric Scanners and Images

Since the biometric scanner plays the central role in our work, this section summarizes the characteristics of the fingerprint scanners and the images they produce.

The commercial fingerprint scanners directly sense the finger surface and acquire a *live-scan* digital image. They are typically more user-friendly, smaller, and cheaper than the scanners used for law enforcement and border control.

2.3.1 Fingerprint Scanners

A fingerprint scanner generally consists of (1) a sensor that reads the finger surface, (2) an A/D converter that converts the reading to digital, and (3) interface that connects the scanner to an external device. Here are the three families of sensing techniques [Maltoni et al. 2003], which we hereby summarize.

Optical Sensing

(a) *Frustrated Total Internal Reflection (FTIR).* The finger surface is illuminated by a bank of LEDs. The light gets reflected at the valleys (appearing bright) and absorbed at the ridges (appearing dark) and is focused through lens onto a CCD (or CMOS for lower cost) sensor. Because of their essentially 3D operation, these scanners cannot be deceived by a spoofing printed image, they also produce images with very high quality, but they may introduce geometric distortion, have problems with dry fingers, and are difficult to be miniaturized. Replacing their glass prism by a sheet prism reduces their size, which, however, also reduces the quality of the acquired images. The scanners using FTIR are the oldest and the most widely used today.

(b) *Optical fibers.* The reflected light from the finger surface is directly conveyed through a micro-optical guides to an array of CCD or CMOS pixels without the need of prism and lens. This makes the scanner more compact, but it also increases is cost as the pixel array has to cover the whole sensing area.

(c) *Electro-optical.* They have two layers: a light-emitting polymer and a photodiode array. In the places where the ridges touch the polymer, the potential is different than under the valleys, thus the amount of light varies and represents the finger pattern. The size of scanners is much smaller, but the images have lower quality.

(d) *Direct reading.* A high-quality camera is directly focused on the fingertip, thus the finger does not touch the surface. The challenge is obtaining well-focused and high-contrast images since a mechanical support keeps the finger at a uniform distance.

Solid-state Sensing

To reduce the size and cost of the optical scanners, in the solid-state scanners the user touches directly the surface of the sensor. The ridge pattern is captured by an array of sensor pixels by using one of the following effects.

(a) *Capacitive.* Micro-capacitive plates are embedded in a chip with the finger being the other plate. When the finger is placed, the capacitors get charged in function of the distance with the ridges and valleys producing different capacitive patterns. Since measuring the charges on the micro-capacitors is inaccurate, each sensor has a specific method for ensuring correct discrimination between ridges and valleys. A critical part of the capacitive sensors is the surface coating, and its thickness is determined as a tradeoff between protection and ability to distinguish between ridges and valleys. Although the capacitive sensors can adjust their parameters to compensate for non-ideal skin conditions, their performance depends on the cleanness of the surface (which requires frequent cleaning). Similarly to optical sensors, the capacitive sensors cannot be deceived by printed images of fingerprints. Capacitive sensors also work well with young, healthy, clean fingers, but their performance may considerably degrade with dry, sweaty, or damaged fingers and for fingers of elderly and people under stress and medication [Ratha and Bolle 2004].

(b) *Thermal.* As the ridges and valleys touch the sensor surface, a pyro-electric material generates current based on the temperature differentials as the valleys are away from the surface. The major problem is that, by reaching thermal equilibrium, this difference shorty disappears.

(c) *Electric field.* An RF signal is transmitted by a drive ring, gets modulated by the derma (subsurface of the skin) and is received by a matrix of antennas. This captures the ridge pattern beneath the skin surface.

(d) *Piezo-electric.* The different distances of ridges and valleys result in different mechanical pressure on pressure-sensitive sensors and produce different currents, which, unfortunately, are very small. These scanners also produce only binary images.

Ultrasound Sensing

Acoustic pulses, sent to the fingertip, get reflected and are captured by the receive sensor. The sound waves penetrate the skin surface and give partial echoes at impedance change, thus imaging the skin subsurface and making these sensors robust against dirt. These sensors produce images with good quality, but the scanner is large and expensive and also slow (takes several seconds to acquire an image), which makes them not very popular.

The most typical mode of fingerprint acquisition is when the finger touches (without moving on) the scanner surface [Maltoni et al. 2003]; these scanners are called area or touch scanners. This is simple but has several disadvantages: the sensor may become dirty, a latent fingerprint may remain on the surface that may impede the subsequent reading, and there are also hygienic concerns. Furthermore, the size of the sensor area (which is large) is directly related to its cost.

The other mode is by swiping the finger over the sensor and is used in the thermal and some optical scanners [Maltoni et al. 2003]; these scanners are called swipe, slide, or sweep scanners. Swiping overcomes the major disadvantages of the touching mode and can significantly reduce the cost as the sensor can have height of only several pixels. The full image is reconstructed by software. The major problems of this method is the training of the users to swipe their fingers properly and the errors that result from reconstructing the image from its slices.

2.3.2 Parameters of Fingerprint Images

The fingerprint image parameters of interest, as described in [FBI PIV 2006] and [Maltoni et al. 2003], are:

- **Resolution.** The minimum (spatial) resolution for FBI-compliant scanners is 500 dpi and it is typical for most commercial scanners today;
- **Area.** Area is an extremely important parameter as images with larger area contain more information (ridges and valleys). The FBI specifies 1"x1" as the minimum area that allows capturing sufficient part of the finger tip. However, to reduce the cost and minimize the device size, the commercial scanners typically capture considerably smaller area (one third to one half of that). This may lead to increased FRR as the overlap between the reference template and the match template is smaller;
- **Depth.** This is the number of bits per pixel. Almost all fingerprint scanners capture only scales of gray. The FBI requires 8 bits/pixel, but some commercial scanners use only 2 or 3 bits and then extrapolate to 8 bits.
- **Geometric accuracy.** This is the maximum geometric distortion and is given as a percentage. Most optical fingerprint sensors introduce geometric distortion.
- **Image quality.** This parameter cannot be defined precisely because it is also related to the finger quality, e.g., too wet or too dry fingers cannot be captured well by the scanners. In [FBI PIV 2006], the FBI has specified only

MTF (Modulation Transfer Function) and SNR which reflect the fidelity of reproduction with respect to the original pattern.

Chapter 3

Problem Statement

In this chapter we formulate the problem with attack models and assumptions and set forth the objectives and requirements for the solution.

3.1 Attack Models

The systems of interest for us are various portable devices that use fingerprint authentication, e.g., mobile phones (cell phones and smartphones), PDAs, laptops, hardware tokens, etc. The two main differences in the authentication to such devices in comparison with other systems and scenarios is that (a) the authentication here takes place in unsupervised environments and (b) portable devices are easily stolen and physical attacks on them are possible. We have identified three groups of possible attacks that are related to the biometric scanner (see Figure 3.1):

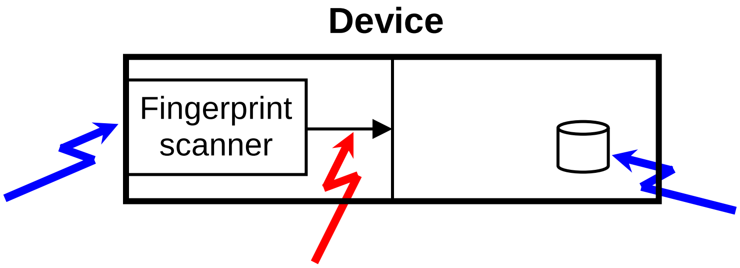


Figure 3.1: Attacks in a portable device related to the biometric authentication

1. **Fake fingertip.** The adversary provides physical reproductions, both as artificial fingers and as reproductions of fingerprint patterns, to the biometric scanner but does not manipulate the acquired image (after the scanner);

2. **Attack on the scanner.** The adversary injects fingerprint images at the output of the scanner (thus effectively bypassing it). This attack includes a malicious replacement of the authentic scanner and a replay of a stolen image of the authentic fingerprint;

3. **Attack on the data storage.** The adversary can access and manipulate the information pertaining to the biometric authentication that is stored in the system memory (e.g., in a RAM or on a hard disk). This includes overwriting results of the work of the system processor(s).

The target of our work is (2) – an attack on the scanner. Countering it essentially requires a verification of the authenticity of the scanner which acquired a particular fingerprint image, a process that we call *scanner authentication*.

3.2 Assumptions

The general assumption is that the biometric information is not secret. We further assume:

1. Possibly except for the biometric enrolment, the process of collecting biometric samples will be not be supervised by an agent or an officer;

2. The adversary has full physical access to all externals of the device, including the biometric scanner;

3. The adversary can replace the authentic scanner by other scanner (i.e., disconnect it and connect the other scanner in its place) without this being detected by the system by other means (other than our solution). The adversary can also at will connect the authentic scanner back to the system;

4. The adversary has access to the output of the biometric scanner and can replace the acquired digital signal, but cannot retrieve information from or modify the contents of the data storage (which can be, for example, protected by a TPM). The adversary also is unable to replace hardware components other than the biometric scanner or modify the software running in the system without this activity being detected by other means (e.g., with a TC technology);

5. The adversary has complete information about the biometrics but cannot produce the authentic live finger to the scanner of this particular device. This includes (a) possession of images of the same authentic finger but acquired with a different scanner (including a scanner of another type), possibly in ideal conditions and with very high resolution and (b) possession of extracted features (e.g., minutiae) or any complete or partial information of the authentic finger obtained from a latent fingerprint (i.e., retrieved from a solid surface) or otherwise;

6. The adversary is able to provide artificial fingers (e.g., made of clay or gelatine) and electronic reproductions obtained by synthesis or forging (including modifying) extracted features of the authentic finger;

7. The adversary can produce to the scanner live biometrics of someone else who is not the legitimate user of this device.

3.3 Objectives and Requirements

Our objectives and requirements from this technology are:

- **Scanners of the same model.** The scanner authentication has to be able to distinguish between canners of exactly the same acquisition technology, manufacturer, and model.

- **Accuracy.** Exact target error rates (FAR and FRR) are difficult to be specified at this stage as this is a research project and they will depend on the implementation constraints, such as time and memory; furthermore, the balance between FAR and FRR depends on the specific ROC and generally is an application-specific requirement. Nevertheless, as a generic metric and objective, we want to achieve an EER below 1%.

- **Efficiency.** Since our goal is to use the technology in portable devices, the computational (and to some extent memory) efficiency are of extreme importance. We believe that the computational efficiency and speed are key for the adoption of such a technology because the scanner authentication is only one part of the authentication process, which as a whole generally has to take very little time (about 1 second or even less) as to not annoy the user and thus discourage the use of the fingerprint authentication altogether. The problem is that the conventional fingerprint verification algorithms are typically very computationally intensive and take up considerable part of the time allowance. In addition to the fingerprint authentication, many companies also employ anti-spoof techniques (implemented in software and/or hardware) to detect fake fingers (i.e., the other type of attack), which techniques also add processing time. And finally, the problem can be particularly severe in portable devices which are constrained both in computational power and in energy. Therefore, the scanner authentication has to impose as little additional computational burden as possible. Although the time requirements for the scanner enrolment can be relaxed (i.e., users could tolerate longer time to enroll their biometrics and devices), the scanner verification should take very little time.

- **Cost.** A major system requirement is to use commercial off-the-shelf components as portable devices are most likely to be equipped only with such components. In particular, only general-purpose, low-cost fingerprint scanners may be assumed as available, i.e., scanners that do not provide any advanced functionality, such as very high resolution or high rate of frame capture. It can be assumed that the fingerprint images these scanners acquire have resolution of 500 dpi and represent sufficient part of the finger (for performing fingerprint verification). Next to the cost of the scanners, of importance for us is also the cost of integration of the technology in biometric systems. History has shown that the more complex a technology is, the higher is the cost of its implementation in products and test. Therefore, simplicity is another central objective.

- **Robustness.** To ensure broad deployment of the technology, it needs to be robust in a wide range of conditions – fingers with different patterns and sizes, different fingertip pressure on the scanner platen, changes in the fingertip skin (scratches, bruises, and wear), and environmental changes (temperature, moisture, dirt/grease, etc.).

- **Fixed-point implementations.** Another important objective is to have the algorithms implementable in precision-limited systems, in particular systems that employ only fixed-point arithmetic, i.e., in FPGAs and general-purpose microprocessors as floating-point coprocessors are usually not present in mobile phones. Therefore, numerically intensive and potentially unstable algorithms have to be avoided as implementing them in fixed-point arithmetic may prove particularly difficult and potentially unreliable.

- **Deployment in existing systems.** Since there are hundreds of millions fingerprint scanners already sold to customers, an optional yet very desirable objective is to be able to add the technology (as an "add-on") to systems already manufactured and put into service.

Chapter 4

Related Prior Work

4.1 Variability in Semiconductors and Devices

The central part of our study is what we call *scanner pattern* – the unique patterns of fingerprint scanners that can be used to distinguish one scanner from another one. We believe that the scanner pattern stems from the variability of element characteristics at semiconductor level and is caused by imperfections of the conversion from the input to the scanner (i.e., the object applied to it) to its output (i.e., the digital image). For this reason, we first review the prior work on variability in semiconductors and devices as it provides the basis from which, in first place, the existence of a scanner pattern can be inferred.

4.1.1 Variability in Semiconductors

We start by summarizing studies on variability at semiconductor level (i.e., in transistors) because, regardless of its specific acquisition method, any fingerprint scanner contains an array of minute sensor elements that can be the dominant factor determining the scanner pattern.

The process parameter variations in semiconductors fall into two general categories: variations among chips (lot-to-lot, wafer-to-wafer, and interdie variations) and intradie variations (due to gradient effects, random component variations, and across-field/across-chip linewidth variations) [Drennan 2002]. Lot-to-lot and wafer-to-wafer variations affect all devices manufactured in a lot/from a wafer (e.g., over-etching reduces the length of all transistors and process temperature and pressure differences induce variations), and they result in systematic variations in the device characteristics [Kinget 2005].

In the context of the analog design, *mismatch* is an intradie parameter variation and designates the difference in the electrical performance of two identical devices on the same chip; the first order effects that cause mismatch are geometry and bias [Drennan 2002]. This work also provides an overview, which we summarize here, of the device mismatch in BiCMOS technologies. For perimeter-dependent parameters such as the dimensions of a MOSFET gate and a BJT emitter, the length variability (i.e., its statistical variance) is inversely proportional to the device width and the width variability is inversely proportional to the device length. The variability of the area-dependent parameters, e.g., the BJT base dopant and MOSFET gate oxide thickness and channel dopant, is inversely proportional to the area. These geometric dependencies stem from the fundamental process parameters. The second cause for variability is the device sensitivity to the process parameter variations. For instance, the MOSFET drain current variation in function of the gate length is determined by the inverse dependence of the gate length variation from its width and the inverse dependence of the device sensitivity from the gate length.

The same work also discusses the mismatch gradients which are "spatially dependent, systematic changes in each process parameter across the die." Gradients superimpose with the random component variations and thus form the total variations. Depending on the distance between two devices on the die, the mismatch may be dominated by the random component effect or by the gradient effect.

The difference between the two types of parameters – the process parameters (the physically independent parameters) and the electrical parameters (the ones of interest to the designer) – used in mismatch modeling of MOSFET transistors is studied in [Drennan and McAndrew 2003]. As examples for process parameters, the authors give the flatband voltage, mobility, substrate dopant concentration, length and width offsets, and source/drain sheet resistance, and as examples for electrical parameters – the drain current, input voltage and trans- and output conductances. The authors also point out that the threshold voltage, however, is neither a process nor an electrical parameter. They also note that in many instances of technology, devices, geometry, and bias, the intradie parameter variations dominate the interdie variations.

The authors further note that the local variations decrease with the increase in the device size because the averaging takes place over a greater area. The global process parameter variation, however, is independent from the length/width. The mismatch is due mainly to local variations because in many cases (technology, device, geometry, and bias) the local variations dominate. Generally, all mismatch models are based on the propagation of variance relationship [Drennan and McAndrew 2003] derived from the dependence of an electrical parameter e from an independent process parameter p_i, i.e., $e = e(p_i)$, as:

$$\Delta e = \frac{\partial e}{\partial p_i}(\Delta p_i) \qquad \text{and} \qquad \sigma_e^2 = \sum(\frac{\partial e}{\partial p_i})^2 \sigma_{p_i}^2 \tag{4.1}$$

In case of MOSFETs and most BJTs, p_i are assumed to have normal distribution; for some BJTs, however, log-normal distributions are better suited for modeling purposes. The same study also presents examples for the standard deviation of the drain current of an NMOS transistor (0.25 μm CMOS technology) in function of the bias (V_{ds}), gate length, and geometry; for small values of the parameters, the standard deviation is well above 1-2%.

Kinget in [Kinget 2005] studies the bandwidth-accuracy-power tradeoff in MOSFETs and BJTs, which is largely fixed and determined by technological parameters, and the impact of the transistor mismatch on this design tradeoff at circuit level. This tradeoff applies to a wide range of applications, in particular to A-to-D converters, sensor arrays, and read-out electronics. To create mismatch models, the author focuses on the dominant factors. For example, in MOSFETs, Kinget finds out that the drain-source current or gate-source voltage mismatch is mainly caused by the threshold voltage differences and the current factor differences, each having a normal distribution with zero mean and variance which is a function of the device area. The data presented in the paper shows that the proportionality constants for the threshold voltage and for the current factor (that form their variances) decrease from about 30 $mV\mu m$ to about 5 $mV\mu m$ and from about 3% μm to 1% μm, respectively, when moving from 2.5 μm to 0.18 μm technology. In BJTs, the relative base current mismatch and the relative collector current mismatch (the standard deviation divided by the nominal) are inversely proportional to the emitter area with proportionality constants in the range from 2% μm to 5% μm and from 1% μm to 4% μm, respectively. The dominant factors for variations in BJTs are the base sheet resistance, the base-emitter current density, and the emitter size. When it comes to circuit level, in another work [Kinget 2007], Kinget points out that the offset of comparators depends on the matching of the gate-source voltage of the two transistors in the differential input pair and the accuracy of the gain of amplifiers with resistive feedback is determined by the matching of the resistor ratios.

Considering the attention which the variability at silicon level has received in the public domain only, it is clear that this variability is a design problem. Obtaining details about it for commercially available fingerprint scanners and studies that quantify it on high level, however, has been difficult. Therefore, in the next section, we summarize the work on variability in one type of image acquisition devices that are most closely related to our problem: digital cameras.

4.1.2 Variability in Digital Cameras

The established term for the variability of interest for us in digital cameras is "pattern noise," which suggests temporal variations, but since the pattern noise is a systematic distortion, this term is somewhat misleading. Pattern noise in digital cameras is used to denote "any spatial pattern that does not change significantly from frame to frame" [Holst 1996]. Generally, the pattern noise has two components: fixed-pattern noise (FPN) and photo-response non-uniformity (PRNU). The FPN, also called offset FPN, is the variation in the pixel-to-pixel values when the camera sensor array is in the dark, and, therefore, the FPN is created by the dark current differences. The FPN is generally due to variations in the detector size, doping density, and impurities; the FPN is additive and does not depend on the signal. The PRNU is the variation in the pixel responsivity when the sensor array is illuminated. The PRNU is generally caused by variations in the detector size, spectral response, and coatings' thickness; the PRNU is multiplicative and signal dependent. Both the FPN and the PRNU are present in both CCD and CMOS image sensors; the CMOS image sensors have even more sources of FPN because of the active readout circuits [El Gamal and Eltoukhy 2005].

Besides the pattern noise that is invariant across frames, the photo image sensors have temporal noise that changes from frame to frame and is independent across the pixels in the array. The temporal noise includes photodetector shot noise, pixel reset noise, readout circuit thermal and flicker noise, and quantization noise.

An early work on quantifying FPN in photomatrices is [Fry et al. 1970], where the term "FPN" was introduced as the signal obtained under uniform or zero illumination of all photodiodes. The authors studied the variability and causes for them in two methods for image acquisition and the corresponding circuits: voltage sampling and recharge sampling. They created models of the variabilities considering the typical contemporary tolerances. For example, for the voltage sampling circuit, they found that the variability of the voltage readout depends on the variability in the threshold voltages of the recharging switch and the amplifying transistors, the gate-source capacitance mismatch (due to the gate-source misalignment, gate metallization inaccuracy, etc.), the dark current mismatch of the photodiode, the gain factor variations of the amplifying transistor, and the mismatch in the output resistance of the saturated output switch transistor. The variations in the quantum efficiency at a particular wavelength and the diode junction capacitance had relatively little contribution to the FPN.

In [El Gamal et al. 1998], the authors estimated the pattern noise in two types of 0.35 μm CMOS image sensors (passive and active) as a sum of two uncorrelated components – a column and a pixel component – each modeled as an isotropic AR(1), representing the correlations among the neighboring pixels and columns. They studied the readout signal paths and found that, for example in passive pixel sensors, the pixel pattern noise is caused by the photodiode leakage, variations in the photodiode area, channel charge injection, and capacitive coupling from the overlap capacitance of the switch transistor. Since all pixels in a row share the same output amplifier, the column pattern noise is caused by the offset in the integrating amplifier, size variations in its integrating capacitor, and mismatches in the output reset transistor, giving high spatial correlations. The proposed models, however, cannot be used for camera identification but rather for measuring the quality of a batch of sensors. The major problem with using these models for our purposes is that they do not separately model FPN and PRNU (the authors use the terms "offset FPN" and "gain FPN," respectively). They also admit that characterizing the PRNU with these models needs several levels of illumination, requiring a large amount of data. In contrast, in CCD sensors the pattern noise can be modeled as a spatial white noise process because it is mainly due to variations in the photodetector area and the dark current, which are spatially uncorrelated, as all pixels share the same output amplifier.

4.1.3 Variability in Fingerprint Scanners

In this section, we summarize the work from which the existence of a pattern in fingerprint scanners can be inferred. For optical and CMOS imaging fingerprint scanners, some of the references presented in the previous section may also apply.

MITRE has developed test procedures [MITRE PIV 2006] for verifying the image quality specifications for fingerprint scanners required by the FBI's personal identity verification program (PIV) [FBI PIV 2006], which program has been developed for the purpose of identifying federal employees and contractors when accessing government facilities. Two controlled scanner characteristics provide evidence about the existence of a scanner pattern: the gray level uniformity and the Spatial Frequency Response (SFR). The other two characteristics, the geometric accuracy and the input-output relation (i.e., the linearity of the conversion), are also indicative for the inherently present imperfections in the conversion process but are difficult to be measured for our purposes.

The gray level uniformity represents the (spatial) non-homogeneity of the conversion, and as such it captures the variability of the sensor array elements (which corresponds to the pattern noise in cameras). For testing is, the input is a uniform gray level test signal (called a *target*). In this way, the test essentially quantifies the difference between the pixel values of the acquired image and the constant pixel values of the theoretical image that would be acquired if the scanner were perfect. For example, the pixel-to-pixel uniformity test requires that 99% of the pixels in fixed-sized areas do not vary from the average in more than 8 or 22 gray levels, depending on the darkness of the gray input target. Considering the range of 256 gray levels, the 22-level requirement allows 8.6% variation in the pixel values, which is considerable. Another test, the small area uniformity test, measures the difference in the average gray levels among different areas and allows a tolerance of 12 gray levels (with light gray input target).Measuring the gray level uniformity is relatively easy and potentially representative of the scanner pattern.

The SFR represents how rapid intensity changes (in space) the scanner can capture, i.e., scanner's ability to represent contrast. The SFR has a typical low-pass filter shape: slow changes in the intensity of adjacent areas are scanned with correct pixel values, while rapid changes result in adjacent pixels having close values. The tolerances for the SFR are also considerable, which is another indication for the level of imperfection of the acquisition process. Since the purpose of this test is to measure the scanner SFR with a single input target, it is difficult to derive from these requirements the allowed SFR variability across the whole sensor array. Nevertheless, one can measure the SFRs of several image blocks and then use the SFR variability (at various frequencies) across the image blocks to construct a scanner pattern. The drawback of this approach is that SFR cannot be measured directly during enrolment or authentication as it requires specifically designed input targets.

Finally, indirect evidence about the variability in fingerprint scanners is the raw image incompatibility, which is a major source of interoperability problems, i.e., the decrease in the performance of the matching algorithms when using one fingerprint scanner for enrolment and another one for verification or identification (see [Ross and Jain 2004] and [Yau et al. 2004], also the results of the fingerprint verification competitions, FVC). A study on the most important quality parameters ("operational quality") of fingerprint scanners for the accuracy of the matching algorithms is [Cappelli et al. 2008]. Methods that compensate for the distortion from the fingerprint deformation are available, but very little work (available in the public domain) has been done on compensating for the scanner-specific distortion [Harris Interoperability]. One study that addresses this problem is [Ross and Nadgir 2008] and proposes a nonlinear calibration scheme based on a thin-plate spline model that creates an average deformation relationship between the two scanners in question.

4.2 Security Applications of Variability

4.2.1 Identification and Authentication

The problem of associating a unique number with a particular device generally has been solved by storing the number in a flash memory or in a mask ROM. The major disadvantages of either method are the additional cost, the man-made (randomness of the) number, usually generated during device manufacturing, and the ability to record and track this number (by third parties). Moreover, flash memories can be reprogrammed and thus the initially stored number cannot serve as a non-alterable identifier.

Another work, [Loftstrom et al. 2000], proposes using the randomness of the silicon process for that. As already discussed, the MOSFET voltage threshold depends on the dopant atoms in the channel and is a major cause for the device mismatch. Because of this, an array of MOSFETs will have different drain currents that can be used to produce a sequence of random voltages, which can then be translated into an identifying number. The sequence of voltages is unique and repeatable because every transistor channel has different number of dopant atoms (which is a major cause for device mismatch).

A downside of this approach is that having sufficiently large dopant variations may require "process-induced, nonrandom fluctuations, such as gate length and width fluctuation" [Maeda et al. 2003]. For this, the latter work proposes another method – obtaining unique numbers by using polycrystalline silicon (poly-Si) thin-film transistors (TFT). The device-to-device variations there are created by the silicon grains present in the channel of the poly-Si TFTs; grain boundaries determine a potential barrier that traps carriers, and only those of them that go over it become current (called "thermionic emission current"). As the grain distribution is naturally random, which produces different currents, and eventually forms a virtually unique number that is non-alterable and non-duplicable. Although there is no additional cost (as a separate device) since the grain variations exist naturally in the die, this solution has hidden costs as incorporating it into fingerprint scanners will require design effort and die area; it must also be compatible with the silicon technology used for the image acquisition. And finally, it is not a solution for the millions fingerprint scanners that have been already manufactured and placed into service.

Another solution for device identification (and authentication) is proposed in [Gassend et al. 2004] and is based on measuring the circuit delays in a digital IC. As in [Maeda et al. 2003], the integrated circuits are manufactured identically (i.e., with identical logical functionality), but because of variations at silicon level, the different delay characteristics produce different responses which are then captured by an arbiter circuit, generating from them a unique binary string that is difficult to be cloned. This method allows storing a secret on a chip that is more resilient to invasive attacks than traditional techniques.

All of the methods above require additional die area and have two major disadvantages. First, they all require additional hardware design effort that is specific for the particular silicon technology and may be difficult to be incorporated in many fingerprint scanners. Their second, and more important, problem is that these methods are not applicable to the fingerprint scanners that have already been manufactured and even sold to customers because these methods cannot be used as hardware, much less software, "upgrades" of existing systems.

4.2.2 Digital Forensics

The question whether the variability in digital cameras is sufficiently large to identify a particular camera has also been answered affirmatively. The paper [Sencar and Memon 2007] gives an overview of the recent research in digital forensics in its three main areas: image source identification (including camera model identification and individual camera identification), detection of synthetic images, and detection of image forgeries. From them, the work most closely related to our study is on identifying digital cameras and flatbed scanners from the digital images acquired with them. The general approach that has been used has been on identifying unique characteristics, e.g., component imperfections, defects, or specifics in the hardware and software processing.

The imperfections in the imaging sensor provide a typical uniqueness that is relatively easy to extract. A pioneering work on identifying CCD video cameras using the FPN, caused by dark currents in the imaging sensor, is [Kurosawa et al. 1999]. The dark current noise is the difference between the pixel values when the sensor is not exposed to light. By averaging many images (the authors propose using 100 frames), the FPN, which is additive, can be isolated and used because it was found to be sufficiently unique. An interesting method, proposed in [Lukas et al. 2005] and [Lukas et al. 2006] and reporting an identification accuracy of 100% for 9 cameras, is based on a much stronger component of the pattern noise – the pixel non-uniformity noise, which is a component of the PRNU (and thus multiplicative to the signal) and is caused by the different sensitivity of the sensor elements. The image is first denoised by a wavelet-based algorithm that extracts from the image a Gaussian noise with specified variance. Then a noise residue is formed by subtracting the denoised image from the original image. The denoising algorithm used, which is actually borrowed from [Mihcak et al. 1999], is one of the weak points because it extracts a noise residual that is assumed additive to the signal, while the pattern noise (PRNU, in particular) in their model is multiplicative to the signal. Even the authors concede that "the denoising filter does a sub-optimal job in extracting the pattern noise." The pattern noise obtained in this way is then compared with the reference pattern noise, obtained in a similar way, using correlation – the typical method used for spread-spectrum watermark detection. This approach is not applicable to our problem because:

1. The assumed model of image acquisition is specific for digital cameras, and although it may be similar to the image acquisition in optical fingerprint scanners, our study showed that it significantly differs from that in solid-state fingerprint scanners and that the scanner pattern cannot be assumed to have a simple multiplicative (like PRNU) and/or additive (like FPN) relationship with the fingerprint pattern;

2. The pixel non-uniformity noise is extracted from images with easy-to-process textures (i.e., smooth images). The wavelet denoising filter "assumes that the image in the wavelet domain is a non-stationary Gaussian signal and the pattern noise is a stationary Gaussian signal," which pattern noise, in addition, is also i.i.d. However, the fingerprint pattern is a series of ridges and valleys and thus appears more as a spatial pulse train or a spatial sinusoid. Therefore, the smoothness assumption for the texture of the images also does not hold in our case because the fingerprint pattern is intrinsically very different. Finally, the statistical characteristics of the scanner pattern depend on the specific type of fingerprint scanner and may have nonnegligible spatial correlation;

3. The proposed method for estimating the reference pattern noise requires many (in the order of tens to a hundred, and at least 50) digital images, which makes it completely inapplicable for biometric authentication. We believe that this large number of images is necessary because considerable part of the signal remain in the noise residual, which has to be compensated by averaging over

many images.

4. Only two pairs of all 9 cameras tested were of the same model – the other cameras were not only of different models, but also from different manufacturers.

Finally, the algorithm is also very complex and computationally intensive.

A further study and enhancement of the described work are proposed in [Chen et al. 2008]. The sensor output model is made more accurate by incorporating gamma correction and color channel gain, but it still considers only the additive FPN and the multiplicative PRNU. An approximation is proposed that simplifies this model, although it also introduces implicit dependence between signal and noise. The same wavelet-based denoising filter is used to remove the host signal from the image. The identification is solved as a joint estimation-detection problem: the PRNU is estimated using a maximum likelihood estimator and the PRNU is detected using optimal test statistics. This allows reducing the number of necessary images, but they are still in the order of tens, which again makes it inapplicable to our problem. Furthermore, the requirement for smoothness of the images still holds, and strong assumptions about stationarity and independence are also made. A predictor of the test statistics is used to determine the unknown shaping factors and distribution of the test statistics. Pre-processing for reducing the artifacts that are systematically present in cameras of the same model but are not unique to the sensor (e.g., color interpolation, row- and column-wise operation of the sensors, and JPEG blockiness artifacts) is also proposed; this decreases the FAR and improves the robustness against compression. The results of the large-scale application of this method were published in [Goljan et al. 2009], which reports a FRR of 0.0238 at FAR of $2.4 * 10^{-5}$.

An extension of [Lukas et al. 2006] to flatbed desktop scanners is proposed in [Khanna et al. 2007]. The flatbed scanners employ a one-dimensional linear sensor array, and the image is constructed by translating the scanner head over the image. Thus, the row reference pattern noise can be extracted from a single image by averaging. The authors used the already discussed wavelet-based denoising filter and an anisotropic local polynomial estimator based on multiscale optimization, which improved the results. In addition to the correlation detector for identification, they also used SVM for classification by constructing two sets of features: (1) the statistical properties of the row pattern such as mean, median, standard deviation, skewness, and kurtosis; and (2) the statistical properties of the correlations between different rows. The SVM method gave much better results, averaging to 96% classification accuracy. Since the acquisition method of flatbed scanners is similar to that of the swipe fingerprint scanners, this approach seems to be applicable to the swipe fingerprint scanners. However, there is one major difference that can be a particular challenge: in contrast to the constant speed of the scanner head of the flatbed scanners, the speed of rolling the fingertip over the sensor cannot be made constant because it is controlled by the user.

Here is the place to mention the only work on identifying biometric scanners [Bartlow et al. 2009] that we are aware of because it uses exactly the same algorithm as the one proposed in [Lukas et al. 2006] for identifying digital cameras. Barlow *et* *al.* applied the algorithm of [Lukas et al. 2006] to 16 optical and 4 capacitive fingerprint scanners. Although they used many images from several subjects, generalizing their approach as a solution for fingerprint scanner identification is difficult because the maximum number of scanners of the same technology, manufacturer, type, and model was only 3 (optical scanners, in two of their sets). Two of the 3 capacitive scanner brands used were from the same manufacturer, but of different models, and only 2 of the 4 capacitive scanners were of the same model. Since the algorithm of [Lukas et al. 2006] works for digital cameras, its high accuracy when applied to optical scanners is not surprising. The highest accuracy reported for optical scanners using a single image for computing the noise reference pattern was 99.65%; however, most of the errors in the confusion matrix were among scanners of the same (optical) model. In the other dataset they used, to achieve accuracy of 98%, 64 training images were needed; with a single training image, the accuracy dropped to 85%. But the most problematic is their third dataset where even for optical scanners (of the same model), there were many identification errors, and the overall accuracy with a single training image dropped to 45%. Reasonable accuracy was achieved with 128 training images, but even there, it was below 90%. Clearly, these results cannot serve as proof for the ability of the algorithm of [Lukas et al. 2006] to identify individual fingerprint scanners of the same model, especially when only a single training image is available and within a large pool of scanners. A possible explanation for this is that the image acquisition process in capacitive fingerprint scanners is very different from that of optical scanners (and in digital cameras in this respect), for which reason the assumption that the algorithm of [Lukas et al. 2006] can detect and extract photo-response nonuniformity noise in fingerprint scanners, capacitive in particular, is not plausible.

Another approach, proposed in [Gou et al. 2007] and [Gou et al. 2009], characterizes the pattern noise of flatbed scanners using three groups of features: (1) the first and second moments of the log-absolute of the noise residual with several denoising filters (averaging, Gaussian, median, and Weiner adaptive filter); (2) the mean, variance, and the error due to fitting Gaussian distributions to high frequency subband wavelet coefficients; and (3) the first two moments of the prediction error applied to smooth regions. PCA is then applied to the resulting 60 features, and then SVM with RBF kernel is used for classification, yielding accuracy of 90% for 7 scanner models.

The use of other types of imperfections in imaging sensors has also been tried out with mixed success. For example, matching traces of defective pixels (hot, cold, dead pixels, cluster defects, etc.) cannot be reliably used for identification as most cameras today use techniques for detecting and compensating such defects. Moreover, some cameras do not have defective pixels and the defective pixels cannot be detected in any image. Sensor dust characteristics (e.g., dust and moisture get attracted to the sensor once the lens is removed) have also been studied for associating an image with a particular camera. Studies also have been done on identifying cameras in cell phones using binary similarity measures, image-quality measures, and higher order wavelet statistics [Celiktutan et al. 2008].

Although not directly related to identifying a particular camera, another general problem is associating a digital image with a class of sources (e.g., the camera model or brand) with common characteristics. A novel idea for such camera classification is proposed in [Kharrazi et al. 2004]. From each image, a vector of 34 numerical features designed to detect the post-processing is extracted: the average pixel value, RGB pairs correlation and energy ratio, distribution of the center of mass, and wavelet coefficient statistics, and image quality metrics. A multiclass SVM is then trained to classify the different cameras.

[Swaminathan et al. 2007] propose estimating from images the parameters and algorithms used by different camera components, such as the color filter array and the color interpolation, and using these estimates in an SVM-based classification to determine the brand and model of the camera used to acquire a particular image. [Filler et al. 2008] solve a similar camera-model classification problem but based on PRNU. From the estimated PRNU, features, designed to reflect differences in the color filter array, color interpolation, and the sensor signal transfer, are extracted and input to an SVM classifier. The proposed features are the first central moments in each color channel, the cross-correlation between the color channels, the block covariance, and the linear-pattern cross-correlation. The reported average classification accuracy is about 91%.

There has also been effort to embed into the image a digital watermark (invisible or visible) that carries information about the digital camera which acquired it. However, robust digital watermarking, i.e., one that cannot be easily detected, removed, or copied, requires computational power that is typically not available in fingerprint scanners, and, generally, comes at additional cost; this type of watermarking has been limited to "secure cameras" [Blythe and Fridrich 2004] and

[FotoNation Patent US'218] and definitely is not suited for the low-end fingerprint scanners of our study. Furthermore, such watermarking cannot be applied to already manufactured and sold fingerprint scanners.

Chapter 5

Our Technology

5.1 Framework

5.1.1 Scanner Pattern

A fingerprint scanner essentially converts the biometric information, i.e., the surface or subsurface of the skin of a fingertip, into one or several digital images. In practice, this conversion process can never be perfect. The imperfections induced by the fingerprint scanner in this process we classify into two general categories: (a) imperfections that are persistent and largely time invariant, which we call *scanner pattern*, and (b) imperfections that change rapidly over time, which we call *scanner noise*.

The scanner pattern can be a function of many and diverse factors in the scanner hardware and software, e.g., the specific sensing method, the used semiconductor technology, the chip layout, the circuit design, and the post-processing. Furthermore, pinpointing the exact factors, much less quantifying them, is difficult because such information is proprietary. Nevertheless, our general observation is that the scanner pattern is mainly caused by non-idealities and variability in the fingerprint sensor; however, the signal processing unit and even the interface unit can also contribute to it (see Figure 5.1).

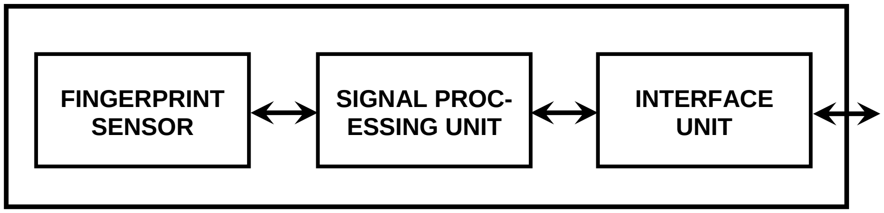


Figure 5.1: Block diagram of a fingerprint scanner

The intrinsic characteristics that cause the scanner pattern remain relatively unchanged over time (by definition). Variations in these intrinsic characteristics, however, may still exist and may be caused by environmental changes such as changes in the temperature, air pressure, air humidity, and sensor surface moisture; material aging; scratches, liquid permeability, and ESD impact on the sensor surface, changes in the illumination (for optical scanners), etc. On the other hand, the scanner noise is generally caused by non-idealities in the conversion process that vary considerably within short periods of time (by definition). Typical examples for scanner noise are the thermal noise (inherently present in any electronic circuit) and the quantization noise (introduced by the A-to-D conversion).

An example for the combined effect of such imperfections, i.e., both the scanner pattern and the scanner noise, is shown in Figure 5.2. Image A (shown on the left side) is an image acquired with no object applied to the scanner platen. A small rectangular block is enlarged and shown on the right side as image B. The three adjacent pixels 1, 2, and 3 of image B have different scales of gray: pixel 1 is darker than pixel 3 and pixel 2 is brighter than pixel 3.

Our primary objective is to estimate the scanner pattern without violating the

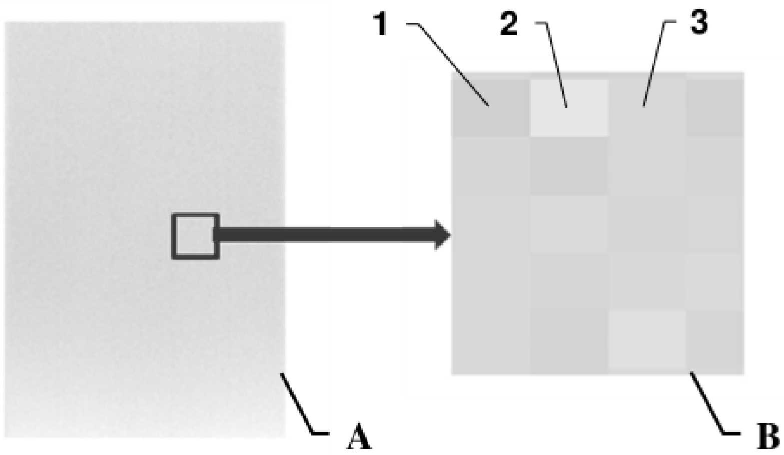


Figure 5.2: Example for scanner imperfections

integrity of the fingerprint scanner by disassembling it, performing measurements inside it, or applying any other intrusive methods, i.e., we have to estimate the scanner pattern solely from digital images acquired with the fingerprint scanner. Another objective is to demonstrate how the scanner pattern can be used to identify a particular scanner and, as such, to serve as a "fingerprint" of the fingerprint scanner. We, however, opted to use the term "scanner pattern" instead of "scanner fingerprint" as the latter may be confusing since the scanners of interest are fingerprint scanners and the term "fingerprint scanner fingerprint" is awkward.

5.1.2 Bipartite Authentication

In the sections that follow, we demonstrate the uniqueness of the scanner pattern and present the algorithms we have developed that are able to detect it and use it to verify the authenticity of a fingerprint scanner from images, acquired with it. In this section, we describe the framework of the scanner authentication.

Hereby we introduce the term *bipartite authentication* to denote the combination of two authentications: a biometric authentication and a scanner authentication. Similarly to any authentication, the bipartite authentication consists of two (sets of) operations, performed one after each other: (1) bipartite enrolment, consisting of biometric enrolment and scanner enrolment, and (2) bipartite verification, consisting of biometric verification and scanner verification.

The biometric authentication and the scanner authentication serve different purposes. The purpose of the biometric authentication is to verify the authenticity of person's fingertip, while the purpose of the scanner authentication is to verify the authenticity of the scanner with which a particular image has been acquired. The scanner authentication can thus detect attacks on the scanner, e.g., a malicious scanner replacement or a replay at the output of the scanner [Maltoni et al. 2003] of a stolen image that still contains the fingerprint pattern of the legitimate user but has been acquired with another, unauthentic fingerprint scanner. This type of attack is becoming increasingly feasible in portable devices (e.g., PDAs, cell and smart phones, and even laptops) because they can be easily stolen, giving the attacker physical access to them and thus the ability to launch so powerful an attack.

The scanner authentication consists of:

- *Scanner enrolment*: estimating and recording the reference scanner pattern of the legitimate, authentic fingerprint scanner. Similarly to the biometric enrolment, several images (e.g., 3) can be used to more reliably estimate the reference scanner pattern;

- *Scanner verification*: estimating and comparing the query scanner pattern with the reference scanner pattern and outputting a *scanner match* decision if the two scanner patterns are sufficiently similar.

An example flow diagram of the process of bipartite verification is shown in Figure 5.3 and needs little explanation. Certainly, the order of the biometric verification and the scanner verification can be reversed: first the scanner is verified and then the biometrics is verified. If the final decision is *bipartite verification match*, then (it is highly probable that): (a) the fingerprint image contains the pattern of the legitimate person and (b) the image has been acquired with the authentic fingerprint scanner.

5.1.3 Scanner Authentication Scenarios

Ideally, both the biometric authentication and the scanner authentication have to operate on one and the same (preferably single) image. Solving this problem directly, however, proved to be difficult, so as first two steps, we solved simpler problems. The three steps, described below as scenarios, the associated difficulty of solving the corresponding problem, and the security level of each scenario, depending on the image used for that (i.e., the applied object to the scanner platen), are given in Table 5.1. A predetermined object is an object known a priori. Since it is known, the difference (in general sense, not limited only to subtraction) between the image acquired with the predetermined object and the theoretical image that would be acquired if the fingerprint scanner were ideal reveals the scanner pattern (because

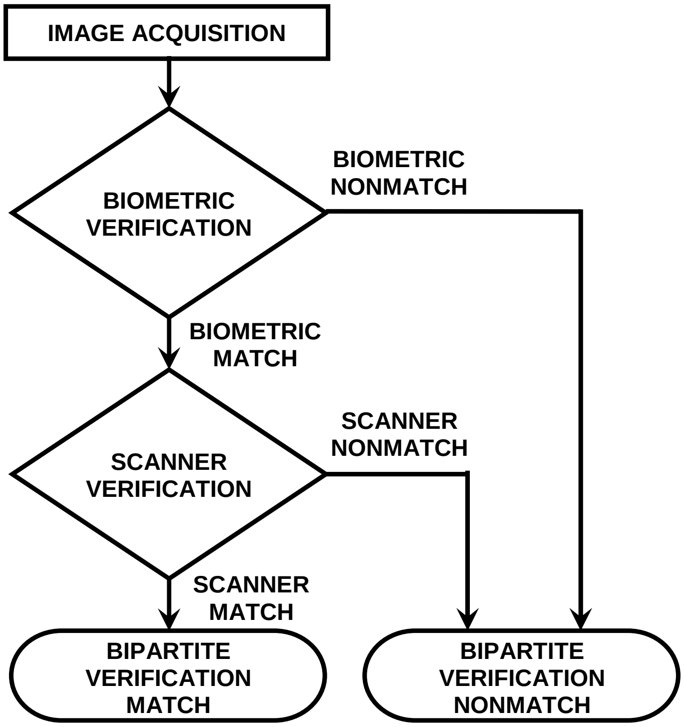


Figure 5.3: Flow diagram of the bipartite verification

the image does not contain a fingerprint pattern). The predetermined object that can serve such a purpose generally depends on the type of the scanner; for example, for capacitive scanners, we found out that air, i.e., no object applied to the scanner platen, is a perfect predetermined object because (a) most capacitive scanners can acquire images with no fingertip pressed on their platen and (b) air is homogeneous and as such it is the ideal input signal.

On the other hand, generally, a fingertip of a person is not known a priori (to the level of detail to easily subtract it from an image with its fingerprint), and an image acquired in this way is a composition of the fingerprint pattern, the scanner pattern, and the scanner noise. Finally, similarly to the biometric enrolment and sometimes to the biometric verification, it is also possible to use more than one image for the scanner enrolment and/or the scanner verification. Since the algorithms we have developed are able to operate even with a single image for the scanner enrolment and even with a single image for the scanner verification, the discussion that follows considers only the case of a single image. However, using more than one image is also possible, which may improve the authentication accuracy; these extensions of the algorithms are discussed in the corresponding sections.

Scenario	**Applied object for scanner enrolment**	**Applied object for scanner verification**	**Problem difficulty**	**Provided security**
A	Predetermined	Predetermined	Low	Weak
B	Predetermined	Fingertip	Medium	Medium
C	Fingertip	Fingertip	High	Strong

Table 5.1: Scanner authentication scenarios: problem difficulty & provided security

In Scenario **A**, both the scanner enrolment and the scanner verification use images acquired with a predetermined object. Thus, the images used for the biometric enrolment and for the scanner enrolment are different: the image for the biometric verification contains a query fingerprint, while the image for the scanner verification is acquired with a predetermined object and contains no fingerprint. Similarly, the images used for the biometric verification and for the scanner verification are also different. The time interval between the two image acquisitions in either case can be made very small (e.g., sufficient for lifting user's fingertip or about a second) within which time a replacement of the scanner is unlikely to happen without being detected by other, hardware-level methods. Detecting a replay of a stolen fingerprint image can be done in a similar way.

The security that Scenario **A** provides, however, is weak as the images acquired for the biometric enrolment (and verification) and for the scanner enrolment (and verification) inherently will be different. Although for the biometric enrolment we may assume that the user will not “cheat” his or her own system/device by replacing the scanner between the biometric enrolment and the scanner enrolment, this may not hold true for the biometric verification. However small the time difference between the biometric verification and the scanner verification is, there is always an opportunity for an attacker to inject a fingerprint image acquired with another scanner and then install the authentic scanner for scanner verification. Nevertheless, Scenario **A** can be used in applications that do not require a high level of security. Also, considering the simplicity of the implementation, this scenario looks very promising in practice as a scanner replacement within very limited time period (e.g., 1 or 2 seconds) can be detected by other methods.

In Scenario **B**, the scanner enrolment uses an image acquired with a predetermined object, and therefore the images used for the biometric enrolment and for the scanner enrolment are different. The biometric verification and the scanner verification, however, use one and the same image, and thus for the scanner verification, the scanner pattern is estimated from an image with a fingertip. This scenario provides a higher level of security than Scenario **A** because the scanner cannot be replaced between the biometric verification and the scanner verification. However, its security level is medium, yet it can still be used in applications that do not require a high level of security.

In Scenario **C**, the biometric enrolment and the scanner enrolment use one and the same image, which image is acquired with a fingertip applied to the scanner pattern. The biometric verification and the scanner verification also use one and the same image, which image is also acquired with a fingertip. This scenario provides the strongest security.

5.1.4 Applications

The method for bipartite authentication can be used to improve the biometric authentication of a user to a system by detecting attacks on the fingerprint scanner that replace a digital image containing the fingerprint pattern of a legitimate user and acquired with the authentic fingerprint scanner by a digital image that still contains the fingerprint of the legitimate user but has been acquired with an unauthentic fingerprint scanner. This type of attack will become an important security threat as the widespread use of the biometric technologies makes the biometric information essentially publicly available. In particular, since the biometric information has a low level of secrecy, an attacker may possess complete information about the fingerprint of the legitimate user, including:

- Possession of digital images of the fingerprint of the legitimate user acquired with an unauthentic fingerprint scanner, including images acquired in nearly ideal conditions and with very high resolution;

- Possession of any complete or partial information about user's fingertip obtained from a latent fingerprint, i.e., from an impression left by user's fingertip on a surface;

- Possession of fingerprint features (e.g., minutiae) extracted from user's fingerprint image;

- Ability to artificially produce digital images that are synthesized from partial or complete information about user's fingerprint.

Incorporating methods for scanner authentication (and bipartite authentication in general) is suited for systems that operate in unsupervised/uncontrolled (i.e., without the supervision of an agent or officer) environments. The methods are particularly important for portable devices, such as PDAs, cell and smart phones, wireless handheld devices, hardware tokens, and generally any mobile devices, including laptops and netbooks, because these devices can be easily stolen, which gives an attacker physical access to them and the opportunity to interfere with the information flow between the fingerprint scanner and the system. This possibility exists even in systems that have trusted computing functionality (e.g., equipped with a Trusted Platform Module, TPM, that provides complete control over the software, running in the system) since the attacker needs not modify the software in order to achieve successful authentication; only replacement of the digital image may be sufficient.

Another possible application of the scanner authentication is in hardware tokens. Many companies and organizations provide hardware tokens to their customers or employees for user authentication and for digital signing of their transactions, usually by using challenge-response security protocols over a network. Typically, the customers authenticate themselves to the hardware token using a PIN code, a password, and/or a bank card. We envision that in some hardware tokens, fingerprint authentication will be used instead as it provides convenience and saves time.

Generally, such increased security is required to gain the confidence of the users in their devices as secure universal terminals for online access to virtually all consumer services: from bank applications to mobile commerce to access to health care anywhere and at any time, for access to medical records, etc.

Local Authentication

A novel paradigm for authentication is proposed in [Ivanov et al. 2010]. It calls for splitting the authentication of a user to a network in two authentications, shown in Figure 5.4: (1) the user authenticates to a portable device, which we call *local authentication*, and then (2) the portable device authenticates to the network.

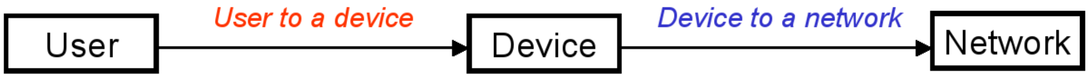


Figure 5.4: Local authentication: user-to-network authentication in two parts

The conventional methods for authentication of a user to a device are by using a PIN code or a password. We propose to instead use biometric authentication and fingerprint authentication in particular. By separating the authentication in two, the biometric information of the user need not be sent over the network to be stored on a remote server, which would be out of the control of the user. Instead, the biometric information is kept only in the device. The major challenge in this approach is that the device is portable and the local authentication takes place in an unsupervised environment. Thus, if an attacker steals the device and gains physical access to it, the attacker can launch a very powerful attack. The second part, the authentication of the device to the network can be implemented in different ways, e.g., by physical layer authentication as proposed in [Yu et al. 2008].

The portable device is user's personal property and in user's possession all of the time. In this way, the biometric information is kept only in the device, not in a computer or a server on the network, and can be locked onto the device. The locking can be implemented using special hardware (e.g., TPM) which ensures that the stored information cannot be compromised because the hardware inherently offers higher degree of security. Thus, the device essentially becomes "an extension" of the user and can be carried by the user at all times. Moreover, this approach requires little or no changes to the infrastructure, in particular, no modification of the security protocols for authentication of a device to a network. It also relaxes the expectations and assumptions about the trustworthiness of the user from the point of view of the network. And finally, the local authentication is capable of "hiding" the identity (e.g., the real name of the user) as it naturally shields the personal information from being sent over the network (or can instead use an identification number) without the need of additional network infrastructure, such as a trusted third party.

Besides the purely technical arguments, the local authentication also helps gain the confidence of the users perceptually and psychologically. Users want to use a technology they are comfortable with but do not want to understand how it exactly works. For example, a patient knows that the biometric authentication works in other authentication scenarios, e.g., when appearing in person in a doctor's office. Now the patient is using her biometrics locally to authenticate to her doctor, which "brings" the doctor right "in front of her." Therefore, in addition to the technical guarantee about preserving the secrecy of the biometric information that our approach gives, it also makes the user more readily accept, and therefore take advantage of, the medical device, enabling the doctors and medical staff to provide better healthcare.

5.2 Initial Research

In this section, we present our initial research that served two important purposes: (1) it demonstrated the uniqueness of the scanner pattern and (2) it gave us insight into the image acquisition process in capacitive scanners, which process we later studied in detail and developed signal models for it. Although the results we obtained during this initial work are not significant from a statistical point of view (because of the limited number of scanners and the limited number of images, acquired with them), the results were sufficiently good to inspire and justify the effort to develop the advanced algorithms.

5.2.1 Test Setup and Images Acquired with Air

At that stage, we used only five capacitive fingerprint scanners: Scanner 1 through Scanner 4 with UPEK area sensors and Scanner 5 with a Veridicom area sensor; details about them are provided in the appendix, Section 6.1.

As a starting point for our study on identifying unique characteristics representative for the scanners, we used the gray level uniformity test of MITRE [MITRE PIV 2006]. Constructing an appropriate target and applying it properly to the scanners, however, proved to be both difficult and expensive. Nevertheless, applying no object at all served our purpose ideally because air is homogeneous and adheres uniformly to the sensor platen. Furthermore, air is naturally detectable by capacitive fingerprint scanners as it essentially represents air gaps (i.e., valleys) of the fingertip skin. Finally, the sensors of both vendors (UPEK and Veridicom) are able to acquire images without any object applied to them.

With each of the five fingerprint scanners, we acquired 30 consecutive raw images with no object applied to the scanners (i.e., with air). We then selected one block of size 64x64 pixels (4,096 pixels in total) that has the same location on each image with respect to its top left corner. Figure 5.5 shows three of the blocks from the first image capture acquired with three different capacitive scanners. Their histograms (on the right side of Figure5.5) show that the pixel values are not constant and span a considerable range. Note that these images contain both the scanner pattern and the scanner noise since no averaging has been done.

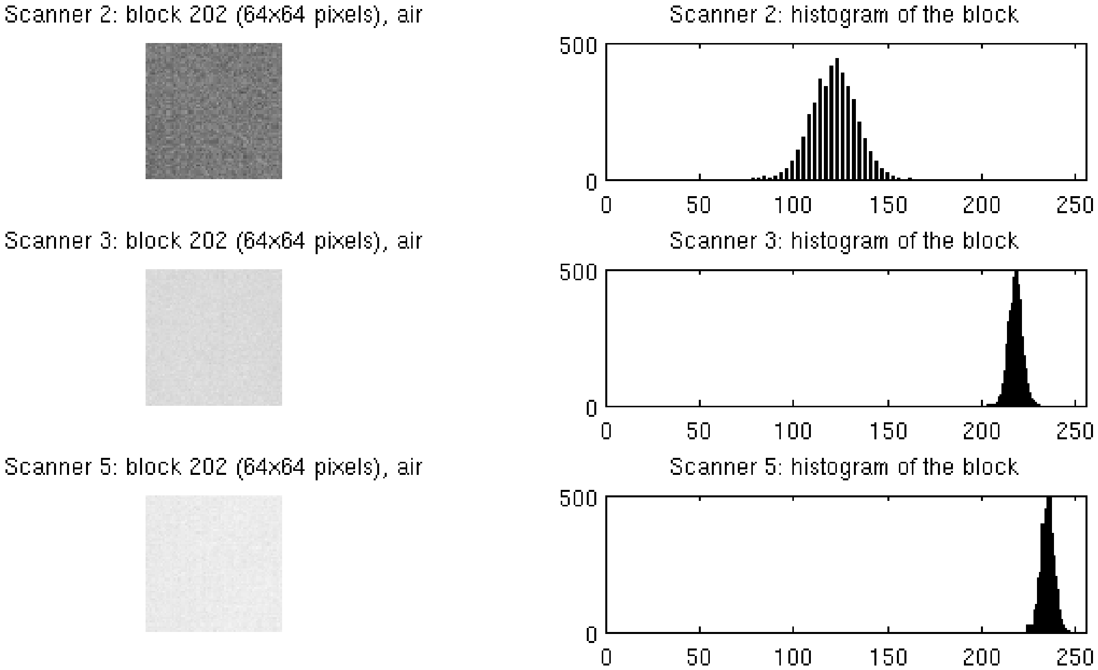


Figure 5.5: Blocks of size 64x64 pixels from images acquired with three of the scanners and their histograms (on the right side)

The 2D DFTs of two of the image blocks are shown in Figure 5.6 (the image sample means had been subtracted before computing the DFT). Observe that the DFT of the image block of Scanner 2 has power nearly uniformly distributed in frequency. The DFT of the image block of Scanner 5, in contrast, has considerable part of its spectral power concentrated in the low frequency region and along the two main frequency axes, showing correlation along the horizontal and the vertical axes of the scanner platen.

5.2.2 Scanner Pattern Estimation from Images Acquired with Air

Next we observed that the pixel values of the same block change from one image acquisition to another, proving that temporal scanner noise is present. The overall pattern, however, remains consistent. To estimate the scanner pattern from the acquired images, we assume that the scanner noise is additive:

$$g(i, j) = s(i, j) + n(i, j, t) \tag{5.1}$$

where $g(i, j)$ is the pixel value of the image at row index i and column index j, $s(i, j)$ is the scanner pattern, and $n(i, j, t)$ is the scanner noise, where t denotes time. We further assume that the scanner noise $n(i, j, t)$ is a temporal zero-mean noise. Therefore, averaging the pixel values $g(i, j)$ for several image captures gives an estimate of the scanner pattern. We averaged 3 image captures (similarly to the typical 3 images used for enrolling the biometric information), which may result in a considerable estimation error but is sufficiently accurate for illustration purposes.

The estimated in this way scanner patterns for 100 pixels (of column 60) for 3 scanners are shown in the first plot of Figure 5.7. The estimated pattern of Scanner 2

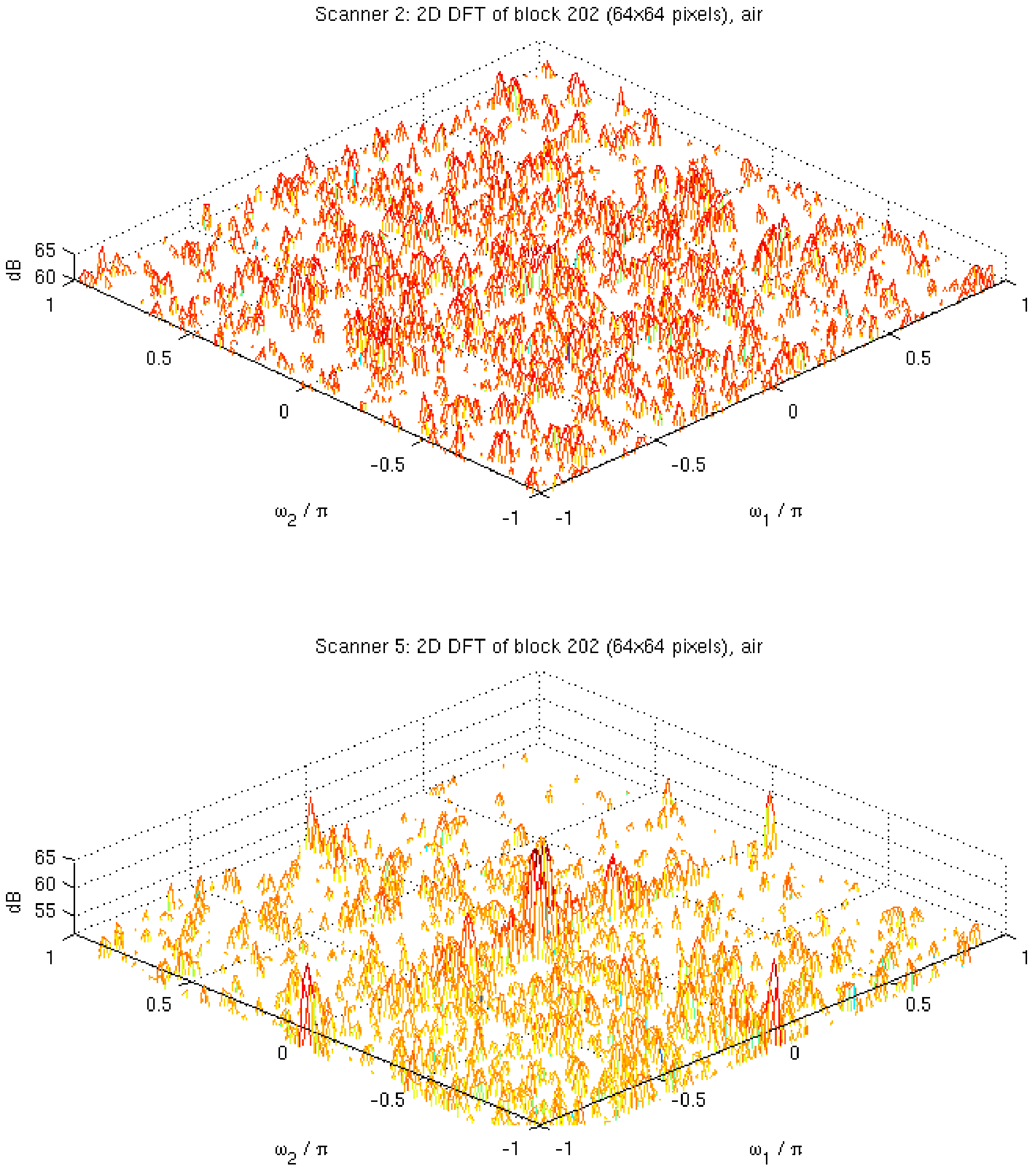
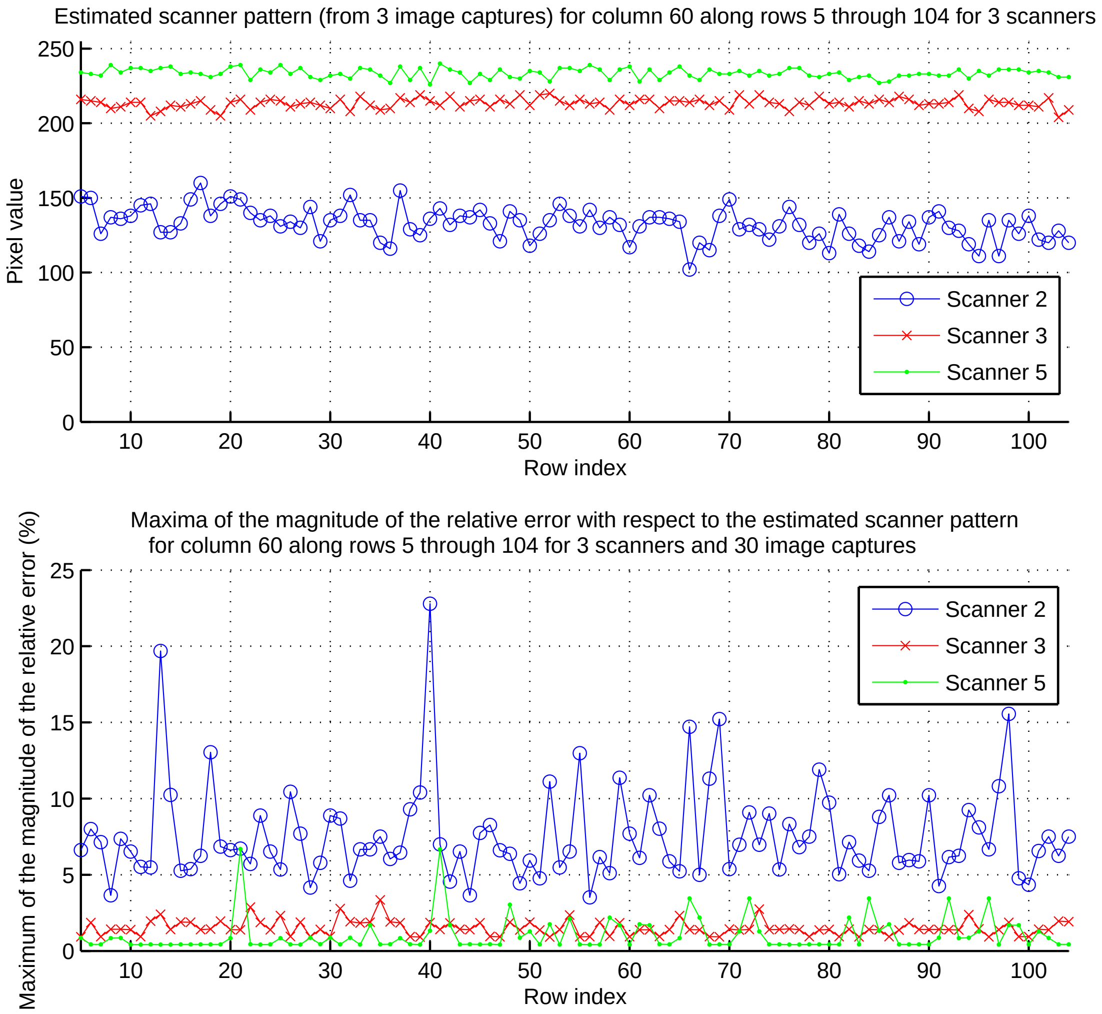


Figure 5.6: 2D DFT magnitudes of the image blocks acquired with 2 of the scanners

exhibits significant variations along rows, while, still considerable and measurable, the estimated patterns of the two other scanners show much less variability. To quantify the deviations from the estimated scanner pattern across several image captures (and thus the level of the scanner noise), we computed the relative error for 30 image captures with respect to the estimated scanner pattern. The maximum magnitude of this relative error is shown in the second plot of Figure 5.7. Similarly, the relative error of Scanner 2 pattern is high (well above 5% on average) and there are several outliers. The relative error of Scanner 3 is mostly constrained within 2% (with a few exceptions), while the relative error of Scanner 5 is generally very low (yet at several pixels, it is considerable).


Estimated scanner pattern (from 3 image captures) for column 60 along rows 5 through 104 for 3 scanners

Maxima of the magnitude of the relative error with respect to the estimated scanner pattern for column 60 along rows 5 through 104 for 3 scanners and 30 image captures

Figure 5.7: Estimated scanner pattern (from 3 image captures) and the maxima of the magnitude of the relative error for column 60 along rows 5 through 104 for 3 scanners and 30 image captures with each scanner

5.2.3 Scanner Pattern Uniqueness

In order to study the uniqueness of the scanner pattern, we take the conventional approach of matching a query to a reference, estimated during enrolment. We use the estimated scanner pattern as defined in Section 5.2.2 as a reference scanner pattern and the correlation coefficient as a measure of similarity and matching.

5.2.3.1 Correlation Matching

The correlation coefficient measures the strength of linear relationship between two vectors:

$$corr(\mathbf{x}, \mathbf{y}) = \frac{(\mathbf{x} - \bar{\mathbf{x}}) \cdot (\mathbf{y} - \bar{\mathbf{y}})}{\|\mathbf{x} - \bar{\mathbf{x}}\| \, \|\mathbf{y} - \bar{\mathbf{y}}\|}, \tag{5.2}$$

where $\mathbf{x}$ and $\mathbf{y}$ are vectors with elements the pixel values, representing, for example, the reference scanner pattern and the query scanner pattern, and $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$ are vectors with (all constant) elements equal to the means of the elements of $\mathbf{x}$ and $\mathbf{y}$, respectively. The decision is match if the correlation coefficient is greater than a predetermined threshold value.

The use of correlation for scanner pattern matching is justified because:

- Correlation, or matched filtering, is the optimal method, in sense of minimizing the probability of detection error, for detecting signals (in communication systems) in the presence of additive white Gaussian noise;

- Correlation is the conventional method for detecting digital watermarks [Cox et al. 2006];

- Correlation has already been used in a similar context, e.g., for identifying digital cameras [Lukas et al. 2006];

- Being a simple and robust method, correlation requires little computational power, which is very important for the applications;

- For the correlation to be the optimal method in our case as well, the scanner noise has to be also spatially white, e.g., at a given time instant t_0, for any column j, the noise samples $n(1, j, t_0), n(2, j, t_0), \ldots$ have to be uncorrelated; the same has also to hold true along rows. For the time being, we can consider both of these two assumptions plausible; the detailed study is provided in Section 5.4.

Despite the fact that our signals are two dimensional (i.e., images), we operate on one-dimensional signals (i.e., column and row lines) because the output signals of adjacent sensor elements in the sensor array may experience some dependence, which may lead to (local) dependence both in the scanner pattern and in the scanner noise (see Figure 5.6), either of which is undesirable for our purposes and analysis. On the other hand, it is reasonable to assume that by increasing the physical distance between two sensor elements, the dependence between their characteristics will decrease. Therefore, for the same total number of pixels, a line of pixels will exhibit less dependence among the pixels than a block of pixels. And finally, one-dimensional signals are easier both to work with and to visualize.

It is important to note that a correlation coefficient with small magnitude only implies little linear dependence, not little dependence in general.

5.2.3.2 Types of Correlation Pairs

To quantify the uniqueness of the scanner pattern both within one and the same scanner and between two different scanners, we devised four types of correlation pairs, each one for comparing a reference line of pixels with another line of pixels from the same scanner and from a different scanner. The reference line consists of 100 pixels of column 60 (this column is chosen at random), containing rows 5 through 104 (not starting from row 1 to avoid edge effect artifacts in the image). For the query lines, we define four types of correlation pairs (see Figure 5.8):

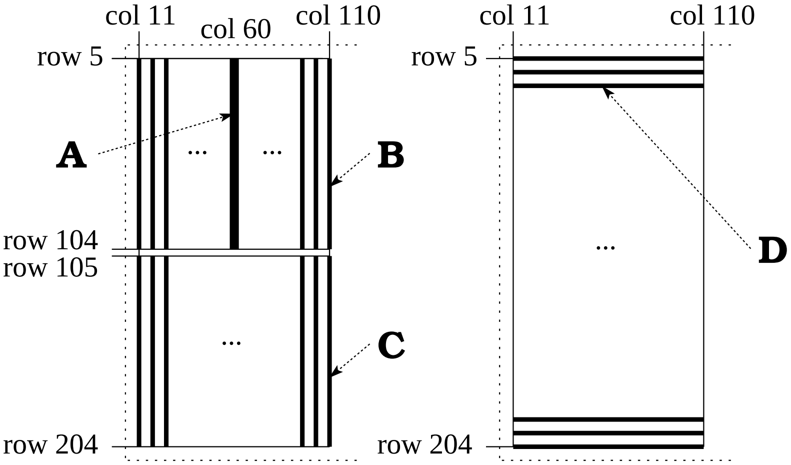


Figure 5.8: Types of correlation pairs

A The query line consists of 100 pixels of column 60, containing rows 5 through 104, from images acquired with the same scanner. Thus, the correlation is between the reference line and the same line from images captured with the same scanner, and we call this type *self correlation*.

B The query line consists of 100 pixels, again rows 5 through 104, of a column from column 11 through 110 (100 columns). The query line can be from an image acquired either with the same scanner, in which case column 60 is excluded from the results, or with a different scanner, in which case column 60 is included.

C The query line consists of 100 pixels, containing rows 105 through 204, of a column from column 11 through 110 (i.e., 100 columns in total). The query line can be from an image acquired either with the same scanner or with a different scanner.

D The query line consists of 100 pixels, columns 11 through 110, of a row from row 5 through 204 (i.e., 200 rows in total). The query line can be from an image acquired either with the same scanner or with a different scanner.

Ideally, the correlation coefficients of type A (self correlation) should be close to 1, and we look for their minimum value as it provides an estimate about the lowest level of self-similarity of the scanner pattern in the presence of scanner noise. The correlation pairs of types B, C, and D we call *cross correlations* as they quantify the dissimilarity between the reference line and other lines from the image, and thus cross correlations quantify the uniqueness of the scanner pattern. Ideally, cross correlation coefficients should be close to 0 because this implies that the two compared lines (and scanner patterns) are linearly independent. We look for the maximum cross correlation coefficient as it provides an estimate about the lowest level of uniqueness of the scanner pattern. The difference between the minimum self correlation and the maximum cross correlation gives the smallest and thus the worst case margin.

5.2.3.3 Results

In this section, we demonstrate that: (a) the scanner pattern is sufficiently random within one and the same scanner and (b) the pattern of one scanner is sufficiently different from the patterns of the other scanners, both of the same model and of different models (and types).

Scanner Pattern Uniqueness within the Same Scanner

First, we computed the correlation coefficients, using Expression 5.5, between column 60 of the estimated scanner pattern (being the reference pattern) and 100 columns from the same image for 30 image captures of Scanner 2. This corresponds to type A (self correlation) and type B (cross correlation) of the correlation pairs. The results, given in Figure 5.9, show significant differences between the self correlation and the cross correlations. The same results are shown again in Figure 5.10, where the correlation coefficients for the subsequent image captures are superimposed (i.e., Figure 5.9 viewed along the image captures). Figure 5.10 clearly shows that the variations of the correlation coefficients (for each column) along different image captures are small with respect to the difference between the self correlation and the cross correlations. The correlation coefficients, again superimposed, for type C correlation pairs are shown in Figure 5.11.

Finally, we computed all four types of correlation pairs with the reference line

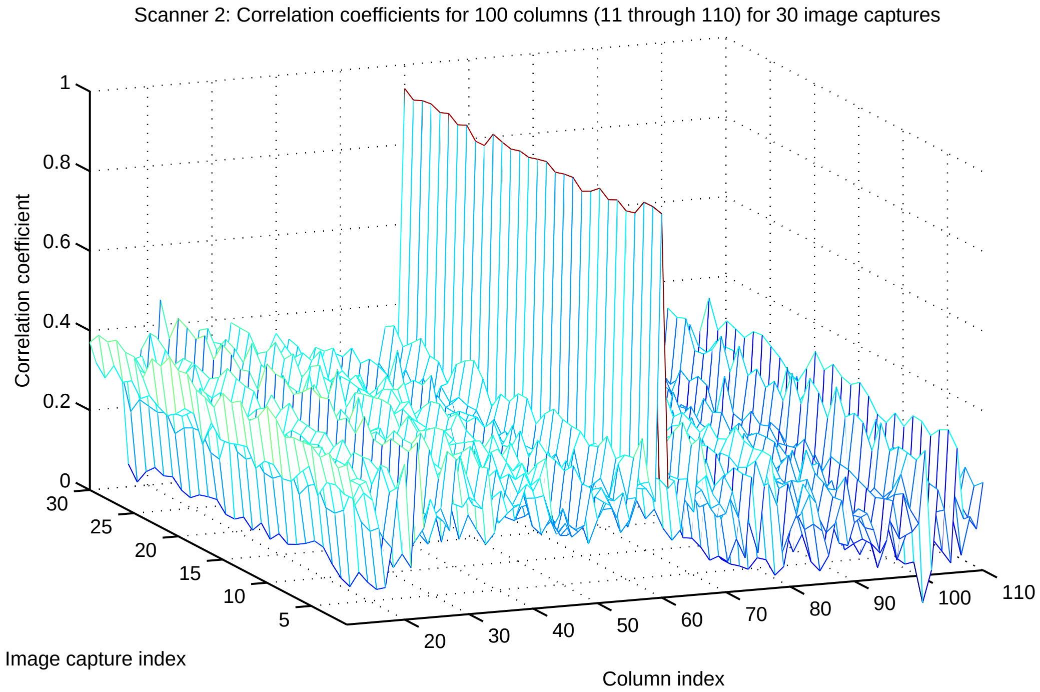


Figure 5.9: Types A and B: correlation coefficients with respect to column 60 of the estimated scanner pattern for 100 columns (11 through 110) for 30 image captures with Scanner 2

(column 60) for all five scanners. The minimum self correlation and the maximum cross correlation of the results are provided in Table 5.2 and show that all five scanners have minimum self correlations of over 0.8. The maximum cross correlations (types B, C, and D) are generally around 0.4 (with one exception: 0.463), which provides a sufficient margin for reliable discrimination between the reference line and the query line.

Scanner Pattern Uniqueness among Different Scanners

Having established that the scanner pattern is sufficiently unique within one

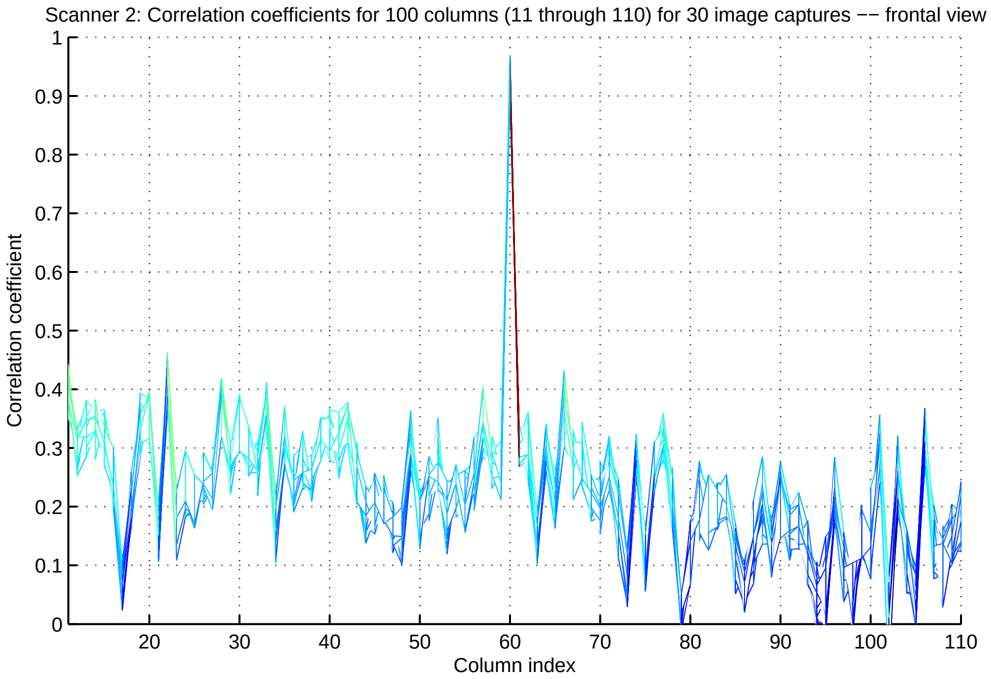


Figure 5.10: Types A and B: correlation coefficients with respect to column 60 of the estimated scanner pattern for 100 columns (11 through 110) for rows 5 through 104 for 30 image captures with Scanner 2, viewed along the image captures

and the same scanner, we then computed the correlation coefficients when the reference scanner and the query scanner are different. We again used column 60 (of each scanner) as the reference line. Since the correlation is computed between patterns of different scanners, there is no self correlation and thus all correlation coefficients represent the dissimilarity between the reference pattern and the query pattern. The maximum values of the correlation coefficients of types B, C, and D are shown in Tables 5.3, 5.4, and 5.5. We observed that, with very few exceptions, the maximum cross correlation varies in the range from 0.3 to 0.45.

Finally, we computed all correlations that we made for column 60 (chosen at

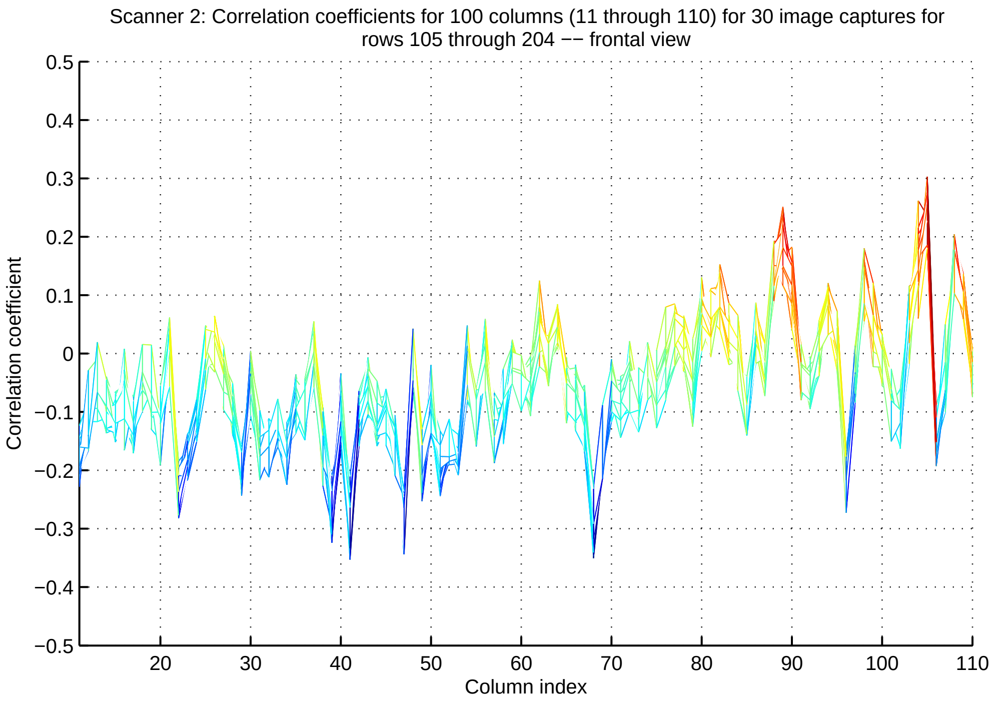


Figure 5.11: Type C: correlation coefficients with respect to column 60 of the estimated scanner pattern for 100 columns (11 through 110) for rows 105 through 204 for 30 image captures with Scanner 2, viewed along the image captures

random) also for column 17 and column 85 (also chosen at random). The results are summarized in Table 5.6. The rows "Max cross" correlation contain the maximum of the cross correlations of type B, C, and D both within the same scanner and between different scanners, giving the absolute maximum cross correlation of all cross correlations and representing the very worst case (among the studied scenarios). Based on these results, we draw four conclusions:

1. For each scanner and each of the three columns, the minimum self correlation and the maximum cross correlation are consistent;

Correlation	**A (min)**	B (max)	C (max)	D (max)
Scanner 1	**0.870**	0.368	0.394	0.400
Scanner 2	**0.901**	0.463	0.353	0.347
Scanner 3	**0.871**	0.407	0.333	0.352
Scanner 4	**0.836**	0.421	0.374	0.418
Scanner 5	**0.802**	0.355	0.299	0.367

Table 5.2: Correlation coefficients for types A, B, C, and D correlation pairs with respect to column 60 for one and the same scanner

Ref / Query	Scanner 1	Scanner 2	Scanner 3	Scanner 4	Scanner 5
Scanner 1	0.368	0.495	0.395	0.392	0.309
Scanner 2	0.390	0.463	0.360	0.439	0.342
Scanner 3	0.324	0.308	0.407	0.288	0.408
Scanner 4	0.373	0.474	0.395	0.421	0.335
Scanner 5	0.307	0.295	0.324	0.346	0.355

Table 5.3: Correlation coefficients for type B correlation pairs (maximum values) with respect to column 60 for all combinations of reference scanner and query scanner

2. Generally, with few exceptions, the larger the maximum cross correlation is, the larger the minimum self correlation is, thus the margin between them roughly remains the same;

3. There is no significant difference between the maximum cross correlations within one and the same scanner and between different scanners;

4. Among all results, the absolute lowest self correlation is 0.802 and the absolute highest cross-correlation is 0.527, giving a very worst case margin of about 0.27.

Ref / Query	Scanner 1	Scanner 2	Scanner 3	Scanner 4	Scanner 5
Scanner 1	0.394	0.303	0.331	0.332	0.432
Scanner 2	0.366	0.353	0.313	0.394	0.397
Scanner 3	0.345	0.341	0.333	0.329	0.333
Scanner 4	0.372	0.368	0.328	0.374	0.339
Scanner 5	0.299	0.352	0.313	0.335	0.299

Table 5.4: Correlation coefficients for type C correlation pairs (maximum values) with respect to column 60 for all combinations of reference scanner and query scanner

Ref / Query	Scanner 1	Scanner 2	Scanner 3	Scanner 4	Scanner 5
Scanner 1	0.400	0.364	0.339	0.450	0.309
Scanner 2	0.379	0.347	0.419	0.466	0.371
Scanner 3	0.304	0.379	0.352	0.367	0.323
Scanner 4	0.314	0.372	0.503	0.418	0.349
Scanner 5	0.401	0.368	0.385	0.404	0.367

Table 5.5: Correlation coefficients for type D correlation pairs (maximum values) with respect to column 60 for all combinations of reference scanner and query scanner

It is also important to note (as we later found out) that: (a) because of the dependence between adjacent pixels (in a column and in a row), the used line segments of 100 pixels above actually have much fewer than 100 degrees of freedom, and (b) the number 100 is roughly the minimum of pixels that should be used to compute a correlation coefficient reliably.

Correlation	Scanner 1	Scanner 2	Scanner 3	Scanner 4	Scanner 5
Column 17					
Min self	**0.871**	**0.939**	**0.889**	**0.894**	**0.873**
Max cross	0.383	0.488	0.461	0.527	0.384
Column 60					
Min self	**0.870**	**0.901**	**0.871**	**0.836**	**0.802**
Max cross	0.495	0.466	0.408	0.503	0.404
Column 85					
Min self	**0.896**	**0.934**	**0.846**	**0.881**	**0.802**
Max cross	0.471	0.458	0.386	0.450	0.482

Table 5.6: Minimum self correlation and maximum cross correlation for the 3 columns for the 5 different scanners

5.2.4 Scanner Authentication Scenarios

In this section, we present our initial research on the three scanner authentication scenarios using the test setup and the acquired images as described in Sections 5.2.1 and 5.2.3. The significance of the received results is not in the numbers, but in the concept and the insight these results gave to us and eventually became the cornerstone of the advanced algorithm development.

5.2.4.1 Scenario A

In Scenario **A**, the applied object is air, both for the scanner enrolment and for the scanner verification. Since this scenario provides weak security, here below we only prove the concept by using the simple correlation matching method and the limited data set, described in Section 5.2.3. Furthermore, we did not test the method extensively with different types of scanners and data and did not optimize the algorithm performance. Still, the perfect results (zero FAR and FRR with considerable margin) makes us confident that the proposed solution is sufficiently good, simple, and practical.

From the results presented in Section 5.2.3, obtained using the correlation coefficient between the reference scanner pattern, estimated by averaging 3 consecutive images, and the query scanner pattern, we computed example threshold values that are the middle points between the minimum self correlation and the maximum cross correlation, shown in Figure 5.12. We observed that a threshold of about 0.67 will be sufficiently good for Scanners 1 through 4 (with the UPEK sensors), while for Scanner 5 (with the Veridicom sensor) a threshold of 0.63 would be more appropriate.

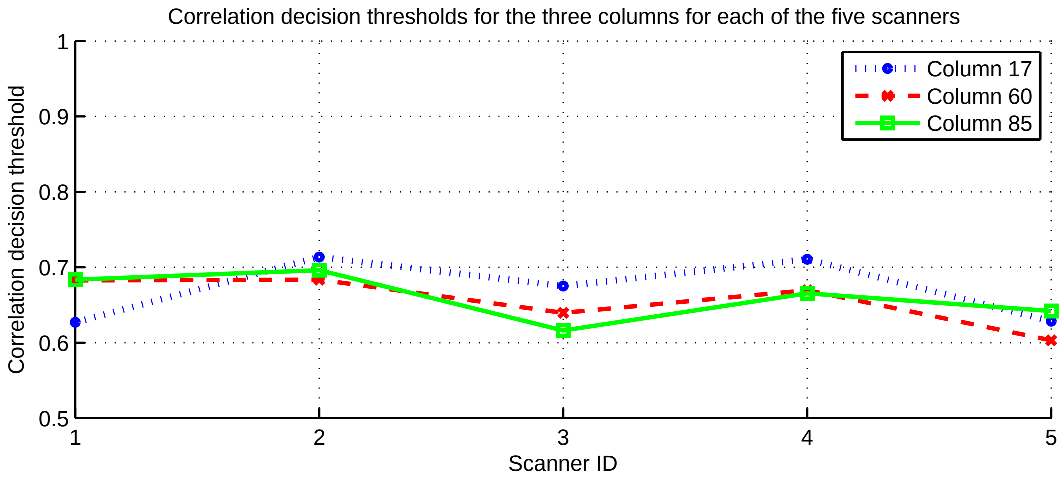


Figure 5.12: Correlation decision thresholds for columns 17, 60, and 85

5.2.4.2 Scenario B

The difference here with respect to Scenario **A** is that here the scanner verification uses the same image as the biometric verification. Thus, the pixel values are composition of the fingerprint pattern, the scanner pattern, and noise. One approach is to develop a method, possibly dependant on the particular sensing technology, that separates the scanner pattern from the fingerprint pattern. At the initial stage of our research, we developed an alternative approach that was not based on a detailed study of the image acquisition process in the scanners but on the observation that the regions of the image that (predominantly) contain only the scanner pattern can be relatively easily isolated from the rest of the pixels and matched against the scanner pattern estimated from the images acquired with a predetermined object applied to the scanner (i.e., air). The pixels of interest are not influenced by the fingertip skin or are influenced only to a relatively small degree. In capacitive sensing, these pixels are in the regions with valleys – essentially, the valleys are air gaps and as such, it is plausible to assume that the pixel values in these regions would be close to the pixel values of the image acquired with no object applied (i.e., air). The larger a valley is, the more room for air gap it provides, and thus the smaller is the influence of the nearby ridges and of the valley bottom on the pixels under this valley. Consequently, the closer these pixel values are to the pixel values of the true scanner pattern estimated from images acquired with air.

One important clarification here: when referring to a valley (and later to a valley threshold), we do not mean the whole valley (in the general meaning of the word, i.e., the area between two successive ridges), but only the parts of the valleys that experience little or no influence from the fingertip skin. Depending on the width and the depth of the whole valley, the valley according to our definition can only be a fraction of the whole valley (i.e., be as small as only a few pixels). Thus, the valley threshold (detailed later) determines the part of the valleys in which we are interested in.

To illustrate the concept, we first present the statistics of the signals. Figure 5.13 shows two image blocks, having the same location on the scanner platen, acquired with air and with a fingertip applied to the scanner, and their histograms (8-bit grayscale: black is 0 and white is 255). We note that the histogram of the image with a fingerprint consists of two regions roughly representing the ridges and the valleys, respectively.

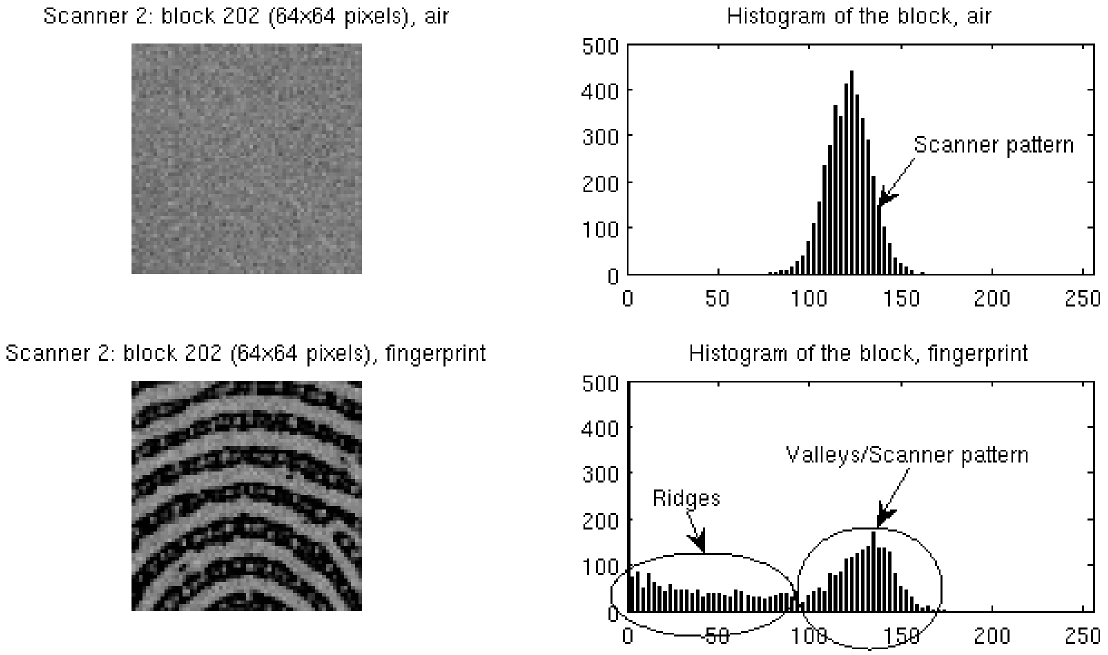


Figure 5.13: Image block and its histogram for Scanner 2 with air and fingerprint

Figure 5.14 shows three image blocks acquired with a fingertip applied to the scanner and their histograms. When analyzing the block images of Scanner 2 and Scanner 3 and their histograms, one can notice that the entire histogram of Scanner 2 image has been equalized or matched (probably by the post-processing software), while the histogram of Scanner 3 image appears to have been enhanced selectively: the pixels with values in the region with ridges have been processed, but the pixels in the valleys/scanner pattern region have been left intact.

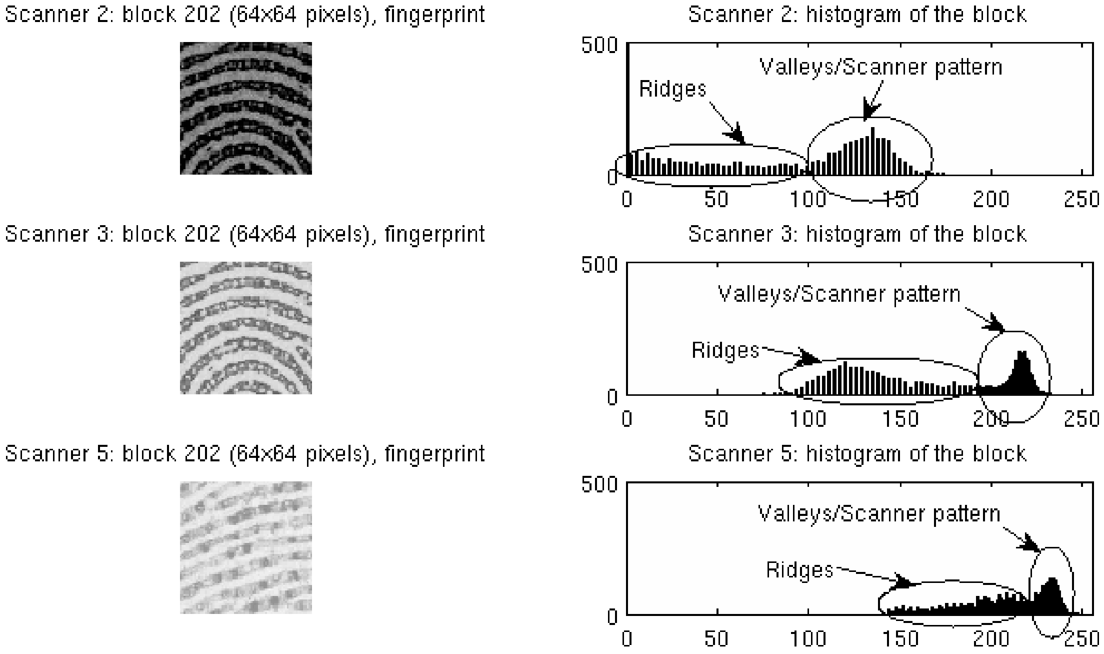


Figure 5.14: Blocks from 3 images with fingerprints and their histograms

In the regions with valleys, the larger the value of a pixel is, the higher the probability is that this pixel belongs to a valley region; the closer this pixel is to the region of the ridges, the higher the probability is that its value is influenced by the nearby ridges, and hence the more different is its value from the value of the same pixel in the image acquired with air. Therefore, as a first approximation, setting a threshold and taking only the pixels with values above the threshold will select those pixels that carry the scanner pattern acquired with air; the rest of the pixels are to be masked as not useful. The results of applying this algorithm to row 100 of Scanner 2 (UPEK) for 100 pixels (starting from column 33 to avoid edge effects) is shown in Figure 5.15. The first plot on the figure is the image acquired with air and two images acquired with a fingertip. Then, the two images with fingerprint are shifted down (to compensate for the constant offset in the different captures) and then thresholded at level 115; the pixels with level below the threshold are set to 0. This, along with the image with air, is shown on the second plot (and enlarged for better visibility). The third plot shows the relative error between the processed images and the image acquired with air (the pixels in the masked regions are not shown). Although quite noisy, the processed images resemble the image with air and have a relative error of about 10%. We then applied the same algorithm to images acquired with Scanner 5 (Veridicom), and the results were remarkably better (see Figure 5.16); a threshold at 230 contains the relative error within 3%.

The small relative error for the Veridicom scanner supports an observation we made later that the pixel values in a sufficiently large valley are close to that of the same pixel from an image acquired with air (see Expressions 5.12 and 5.33). The considerably worse results (over 3 times in terms of the relative error) for Scanner 2 (UPEK) in comparison with Scanner 5 (Veridicom) can be explained by the much higher sensitivity of the UPEK scanners.

An important consideration is the number of pixels (in the valleys) that can

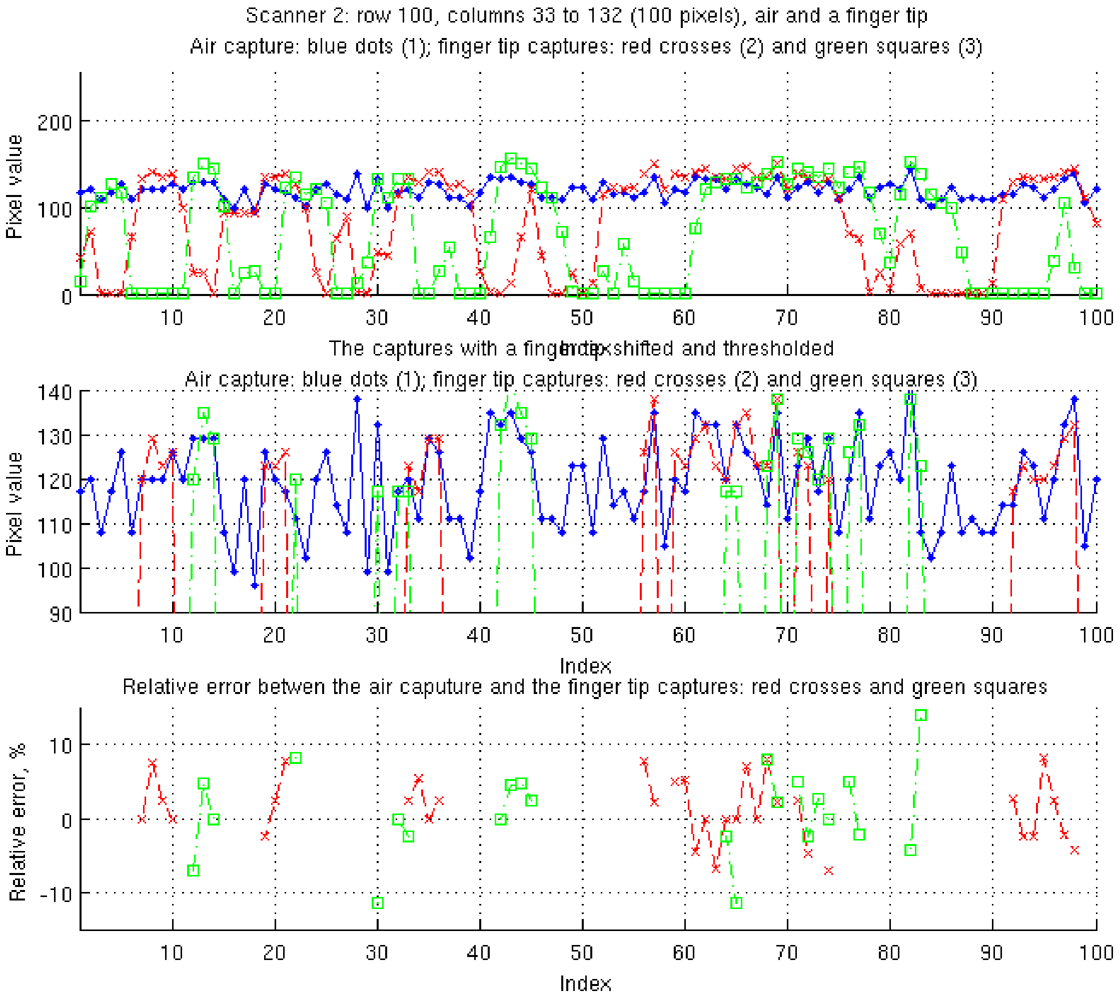


Figure 5.15: Scenario B for Scanner 2: images with air and with fingerprint and the relative errors

be used for scanner verification. This number depends on several factors: the fingerprint pattern (which is user dependent), the amount of impurities on the fingertip skin (e.g., water, grease, and dirt), the particular capacitive sensing technology, etc. Thus, image acquisitions at different time instants will have the valleys (of the same fingertip) cover different regions of the scanner platen in a relatively random way. Still, these regions are not completely random because proper positioning of the fingertip on the scanner platen will limit the possible positions of the valleys;

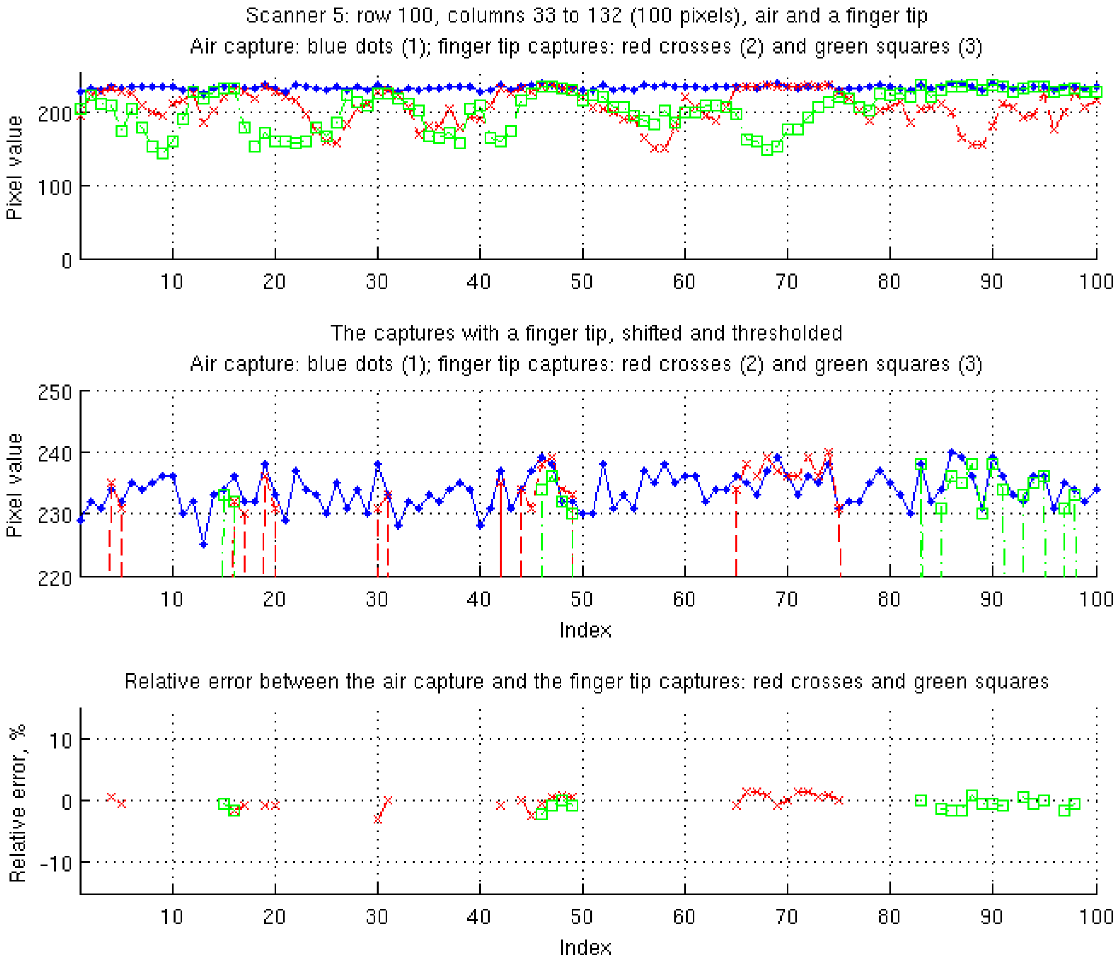


Figure 5.16: Scenario B for Scanner 5: images with air and with fingerprint and the relative errors

furthermore, the fingerprint pattern (of one fingertip) does not change significantly over time.

Based on the idea to isolate the pixels from the valleys and use them for scanner authentication, we developed and tested a set of algorithms, described as Scenario **B** in this section and Scenario **C** described in the next section. However, since the algorithms of this initial research were based on heuristics and were intended mainly to test the concept, we did not put significant effort into optimizing the algorithm parameters and we did not perform extensive testing with other scanners than those described in Section 5.2.1. Thus, with each of the five scanners (4 of UPEK and 1 of Veridicom):

- **For the scanner enrollment**, we used the 30 images acquired without applying any object to the scanner (i.e., with air) as detailed in 5.2.1;
- **For the scanner verification**, we additionally acquired 30 images for each of the 10 fingertips of a single person. Thus, we tested the algorithm on 1,500 images with fingerprints (= 5 scanners * 10 fingertips * 30 images/fingertip), some of which were acquired about half a year apart. The images with fingerprints were used in the tests in the following order: from the left little (**ll**) finger through the left thumb (**lt**) and then from the right thumb (**rt**) through the right little (**rl**) finger (see Table 5.7). Although limited to the fingertips of a single person, we believe that these images provide sufficient diversity of fingerprint patterns and valley sizes. For example, thumbs typically have much wider and deeper ridges and valleys than little fingers (of the hands of one and the same person). Also typically, index fingers have narrower and shallower ridges and valleys than thumbs, and wider than little fingers.

Both for the scanner enrolment and the scanner verification and for all images (with air and with fingerprints), we used one and the same block of pixels (referred to *region of interest* or *ROI*) from each image. Each ROI is a block of 200 rows * 100 columns (20,000 pixels in total), selected 10 columns and 10 rows away from the left and the top edges, respectively, in order to avoid edge effects. This block area

Finger	Abbreviation	Finger	Abbreviation
left little	***ll***	right thumb	***rt***
left ring	***lr***	right index	***ri***
left middle	***lm***	right middle	***rm***
left index	***li***	right ring	***rr***
left thumb	***lt***	right little	***rl***

Table 5.7: Finger abbreviations and test order

was chosen as to ensure a sufficient number of pixels (in the valleys) that overlap, giving a consistent performance of the algorithm. The block area is about 1/3 of the image area of the UPEK scanners and 1/4 of image area of the Veridicom scanner.

The essence of the algorithm, both for the scanner enrolment and for the scanner verification, is in selecting those pixels which experience little influence from the ridges of the fingertips. The algorithm is heuristic and rudimentary, and its parameters are not extensively optimizes because its objective is only to demonstrate that the concept works, not to provide a real and practical solution. The first plot of Figure 5.17 shows the histogram of an image block containing a fingerprint pattern. The region with valleys and the grayscale levels on the most right side are enlarged and shown on the second plot. The algorithm detects the maximum value of the valley region (marked as “peak value”) and subtracts from it a predetermined parameter value (the window size), yielding the valley threshold (see the second plot of Figure 5.17). Because of the existence of dead and defective sensing cells, marked as “outliers” on the same plot, which cells produce constant-value pixels that lie outside (on the right side of) the region with the valleys and mislead finding the peak, the algorithm first detects the outlying pixels and excludes them from the peak value search.

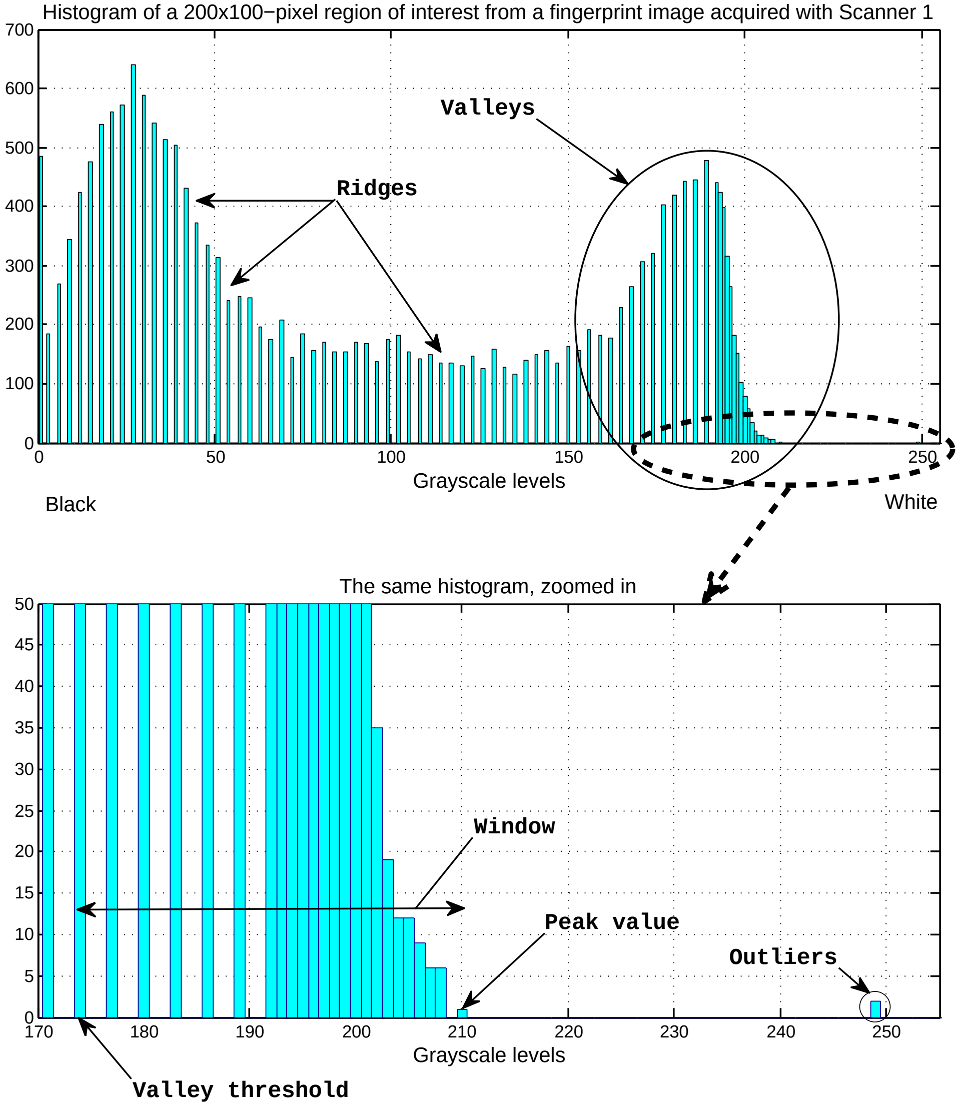


Figure 5.17: Histogram of a 200x100-pixel region of interest from a fingerprint image

Scanner Enrolment

The scanner enrolment uses three images with air, taken consecutively, and consists of two parts:

I. Computation of the threshold level for detecting the regions with valleys;

II. Computation of the decision threshold for the correlation coefficient, which threshold is then used for the scanner verification.

I. Computation of the Valley Threshold

As already discussed and also shown in Figure 5.17 (which represents a histogram of a fingerprint image, but for an image with air, the second plot is similar), the first step is isolating the clusters with outliers, which we do in 3 steps, described below in the MATLAB language:

1. Find the distances (in grayscale levels) among the pixel values in the right tail of the histogram of the ROI. The tail is defined as the last 10 non-zero bins of the histogram. The distances are actually the differences between the non-zero bins of the histogram: `roi_tail_diff = diff(roi_tail);`

2. Find the maximum value of these distances. The assumption is that this maximum distance marks the separation between the region of the valleys and the region of outliers: `[max_difference, max_idx] = max(roi_tail_diff);`

3. Compute the mean of the remaining distances that lie on the left (and thus are before this maximum):

   `mean_remaining = mean(roi_tail_diff(1:max_idx-1));`

If the maximum value, found above, is over 4 times than that of the computed mean, i.e., if `max_difference > 4*mean_remaining` then indicate that a cluster with outliers is detected and mark these bins to be excluded from the computation of the peak value of the region with valleys.

This algorithm detects only a single cluster of outliers, but our tests showed that this is sufficient. However, it is relatively easy to isolate more than one cluster by repeating the algorithm above after sequentially removing cluster after cluster. We designed the algorithm by trial-and-error and tuned its parameters (in particular, the coefficients 4 and 10) for the currently available data, and it works very well.

Once the desired clusters are isolated, the maximum of the remaining pixel values is found and is marked as the "peak value." The valley threshold then is computed by subtracting the window size (how to choose it is explained in the next section) from the peak value: `valley_threshold = peak_value - window_size`. Thus, the valley threshold is specific for each image (actually, it is even specific for each ROI).

After computing the valley threshold, the pixels to be used for the enrolment or for the verification are those whose values are above this valley threshold. In this way, the outlying pixels that were excluded in the computation of the peak value **are** included in the correlation computation and this is desirable because the dead and/or defective sensing cells are unique and representative for the particular scanner [Geradts et al. 2001]. Furthermore, since the pixel values that these nonoperational sensing cells produce are constant (or nearly constant) for different image acquisitions, they constitute reliable elements in the vector used in the correlation computations.

II. Computation of the Decision Threshold

The scanner enrolment uses three images. First, we compute the valley threshold for each image, which is specific for the image. The valleys of interest are isolated based on these thresholds. The set of indices of these pixels in the original ROIs are then intersected in pairs in order to take into account only the same sensing cells/pixels. Next, the resulting ROI matrices are read columnwise, forming three sets of paired vectors, whose three correlation coefficients (see Expression 5.5) are computed: between images 1 and 2, between images 2 and 3, and between images 3 and 1. These correlation coefficients are then used to determine the decision threshold.

If a vector of these 3 vectors has less than 200 elements, the corresponding image is discarded and a new image (with air) is re-acquired. Over the course of experiments, we observed that if the vectors have fewer that 200 elements, because of noise and distortion, the correlation coefficient becomes unreliable, giving unacceptably high error rates. This problem is particularly severe for images with fingerprints (i.e., the query image) as there the level of distortion from the fingertip skin becomes significant. We chose the number 200 as a tradeoff between the reliable computation and the availability of pixels, and it is relatively very low: only 1% of the total number of pixels in the whole ROI (20,000). In case of images with air, the number of pixels in these vectors is in the order of 17,000 (i.e., almost all pixels in the ROI), resulting in very reliable correlation coefficients. The high number of available pixels is also due to the absence of distortion from the fingertip skin.

Finally, the three correlation coefficients are averaged and this average is multiplied by a scaling coefficient to produce a decision threshold that is used for making a decision during the scanner verification step. The scaling coefficient is chosen empirically as result of the tests, and the reasoning behind the choice is explained in the next section.

Scanner Verification

The scanner verification uses as a query image an image acquired with a fingertip applied to the scanner. Next, the pixels in the valleys in this image are isolated using the same algorithm for valley threshold computation, described in the previous section, with the same parameter values (for the window size, outliers, etc.). The algorithm produces a set of indices (i.e., a mask), which mask is then intersected (as a set) with each of three masks of the images with air, determined in a similar way during the scanner enrolment, giving the appropriately-sized vectors to compute the three correlation coefficients. The three correlation coefficients are then averaged, producing a single score that is compared with the decision threshold. If the score is above the decision threshold, the decision is scanner verification match; otherwise, the decision is scanner verification nonmatch.

If the number of common elements/pixels that overlap in the intersection of two masks is smaller 200, the corresponding correlation is tagged as invalid and excluded from the average in the final score because it is unreliable, and the remaining two correlation coefficients are averaged. If all three correlation coefficients are invalid, the matching score is set to 0, giving a scanner verification nonmatch decision. In Scenario **B**, very seldom is a correlation coefficient tagged as invalid because the size of the masks of the images with air is very large, and thus the intersection mask is also very large (about one order of magnitude larger than the minimum of 200 pixels or about 10% of the whole ROI).

I. Choice and Optimization of the Parameters

The two most important parameters that control the performance of the algorithm are (i) the window size and (ii) the scaling coefficient. The model for the pixel values in images acquired with air was discussed in Section 5.2.2 (see Expression 5.1). For images with fingerprints, we can model their pixel values in a very general way as:

$$g(i,j) = d(s(i,j), f(i,j)) + n(i,j,t) \tag{5.3}$$

where $g(i,j)$ is the pixel value of the image at row index i and column index j, $s(i,j)$ is the scanner pattern and $n(i,j,t)$ represents the scanner noise at time t, again assumed to be a temporal zero-mean noise. $f(i,j)$ is the fingerprint pattern, and $d(\cdot)$ represents the "distortion" function that captures the influence of the fingertip skin on the pixel values. By selecting regions with valleys carefully, as explained earlier by thresholding, the distortion can be considerably reduced so that the pixel values $g(i,j)$ become almost equal to the scanner pattern $s(i,j)$ (ignoring the noise). Finally, since the ROI is "sampled" by selecting only pixels in the valleys, the dependence between the subsequent pixel values $g(i, j)$ that come in as elements of the vectors in computing the correlation coefficient also decreases because the pixels in these vectors not necessarily pertain to adjacent sensing cells.

(i) Window Size

The size of the window that determines the valley threshold is the most critical and sensitive parameter and was subject to considerable optimization and tests. Its optimal value is a tradeoff between two factors:

- The **distortion level** introduced by the fingerprint pattern via the distortion function $d(\cdot)$: the smaller the window size is, the smaller is the distortion from the fingerprint, and thus the closer the pixel values $g(i, j)$ are to the scanner pattern $s(i, j)$;

- The **number of pixels** for computing the correlation coefficient: the larger the window size is, the larger is the number of pixels in the vector for computing the correlation coefficient, and thus the more statistically reliable the coefficient is.

If the window size is too small, more pixels in the selection (after thresholding) have little distortion, but they also become very few in number and thus the cross correlation becomes unreliable, giving rise to excessive FAR. On the other hand, if the window size is too large, the distortion level becomes so big that the self correlation becomes too low (because there is less scanner pattern contained in the pixels that are being correlated), the gap between the self correlation and the cross correlation distributions closes in, giving rise to both higher FRR and higher FAR. Thus, the window size, which depends on the type of the scanner, and the minimum number of pixels for correlation computation (currently set to 200 for all five scanners) are interdependent. The window size parameter is particularly sensitive and its current values have been selected after careful tests and observations, and the values we used are:

- For Scanners 1 and 2 (the FIPS-201 certified UPEK scanners): 55 grayscale levels;
- For Scanners 3 and 4 (the regular UPEK scanners): 30 grayscale levels;
- For Scanner 5 (the Veridicom scanner): 18 grayscale levels.

The minimum number of pixels for computing the correlation is also important but is not as critical as the window size.

(ii) Scaling Coefficient

Generally speaking, the scaling coefficient controls the tradeoff between the FRR and the FAR. In the scanner enrolment, by averaging the 3 correlation coefficients for the 3 images acquired with air, we compute an estimate for the mean/center of the distribution of the self correlation. We observed that the cross correlation coefficients are centered around 0. Thus, if the two distributions (self and cross) were similar in type (and variance), the optimal point would be the middle point between 0 and the estimated mean of the self-correlation (during enrolment), giving a scaling coefficient of 0.5. However, the scanner verification is performed by computing the correlation coefficient between the enrolled images acquired with air and a query image with a fingerprint pattern in it, in which case, due to the distortion from the fingerprint pattern, the self correlation is much smaller than the self correlation computed during enrolment when no distortion from fingerprint pattern is present.

Furthermore, the self correlation distribution in the scanner verification is much closer to 0 than the self correlation distribution in the scanner enrolment. We also observed that the self correlation distribution is much wider than the cross correlation distribution (see Figure 5.18). Hence it is clear that the middle point between 0 and the estimated center of the self correlation distribution would not be optimal as it would result in a high FRR. Without further quantifying the optimal decision threshold accurately since our goal was to only illustrate the concept, the tests we performed led us to choosing the following values for the scaling coefficient for Scenario B: 0.20 for Scanners 1 through 4 (i.e., all four UPEK scanners) and 0.25 for Scanner 5 (the Veridicom scanner).

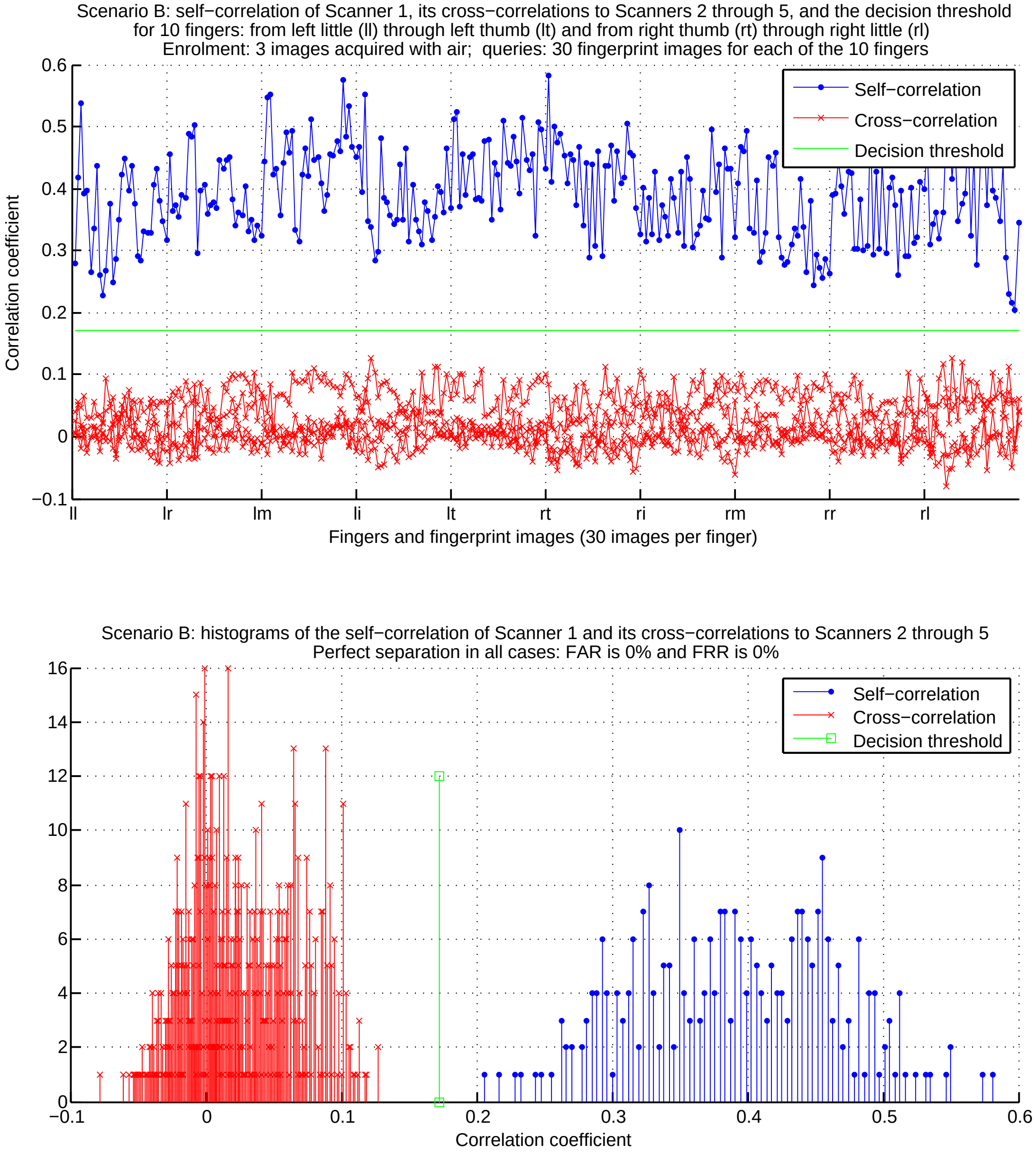
II. Results

The query images, all containing fingerprint patterns, that we used are 1,500: 30 images per finger for all 10 fingers acquired with all 5 scanners. We computed the error rates individually for each 300 images per scanner (30 images * 10 fingers); thus, an error rate of 0.3% in the table is equivalent to one decision error, 0.7% – to two decision errors, etc. The first plot of Figure 5.18 shows the correlation coefficients for Scanner 1. The x-axis represents the 10 fingers (from left little to right little, see Table 5.7), each containing 30 images. To demonstrate the algorithm robustness for different enrolled images, we used four sets of enrolled images, each containing three images with file indices, respectively, being (1,2, and 3), (4, 5, and 6), (10, 11, and 12), and (28, 29, and 30).

The most challenging proved to be the query images with little fingers (left little ***ll*** and right little ***rl***) because their valleys are both narrow and shallow. This leads to (a) fewer number of pixels for computing the correlation because the valleys are narrow and (b) higher level of distortion from the fingerprint pattern because the valleys are shallow, thus yielding unreliable and small self correlation coefficients (see Figure 5.18) that potentially can result into a high FRR. A reasonable decision threshold is 0.17 for Scanners 1 through 4 (UPEK) and 0.23 for Scanner 5 (Veridicom). All results with these thresholds are given in Table 6.1 in the appendix. In summary, for scanners 1, 2, 3, and 5, the results are perfect: both the FAR and the FRR in all cases are zero. The only non-zero false error rates are for Scanner 4 (a non-FIPS compliant UPEK scanner): the FRR is in the range of 1 to 2%, which is due to errors with the right little finger, and the FAR when matched against Scanner 5 (Veridicom) is around 1% (maximum of 1.3%), which is due to errors with the two little fingers.

5.2.4.3 Scenario C

In Scenario **C**, images with fingerprints are used for the scanner enrolment (as well as for the scanner verification). The Scenario **C** solution employs the same algorithms as Scenario **B**, described in the previous section. The values of the window



Scenario B: self−correlation of Scanner 1, its cross−correlations to Scanners 2 through 5, and the decision threshold for 10 fingers: from left little (ll) through left thumb (lt) and from right thumb (rt) through right little (rl)
Enrolment: 3 images acquired with air; queries: 30 fingerprint images for each of the 10 fingers

Scenario B: histograms of the self−correlation of Scanner 1 and its cross−correlations to Scanners 2 through 5
Perfect separation in all cases: FAR is 0% and FRR is 0%

Figure 5.18: Scenario B correlations and their histograms for Scanner 1

size for each scanner type are the same as in Scenario, but the minimum number of pixels for computing the correlation and the scaling coefficient are different.

In this scenario, we observed that the number of pixels used for computing the correlation can be much smaller than that in Scenario B, especially for the cross correlations, falling from about 2,000 in Scenario **B** down to about 200 to 400, which is only 1% to 2% of all pixels in the ROI (20,000). This made difficult meeting the requirement for a minimum of 200 overlapping pixels that constitute the vectors for computing the correlation. Generally, this would not be a problem if a larger part of the image (not the current one fourth to one third of it) is used as an ROI, but in order to be consistent with Scenario **B**, we preserved the ROI size and decreased the minimum number of pixels for the enrolment to 170, which (a) is not much less than 200 and (b) is reached only on two occasions. Since the three images with fingerprints are from the same fingertip and thus are close enough to each other to ensure reliable computation of the correlation coefficient, this decrease in the minimum number of pixels is not a problem.

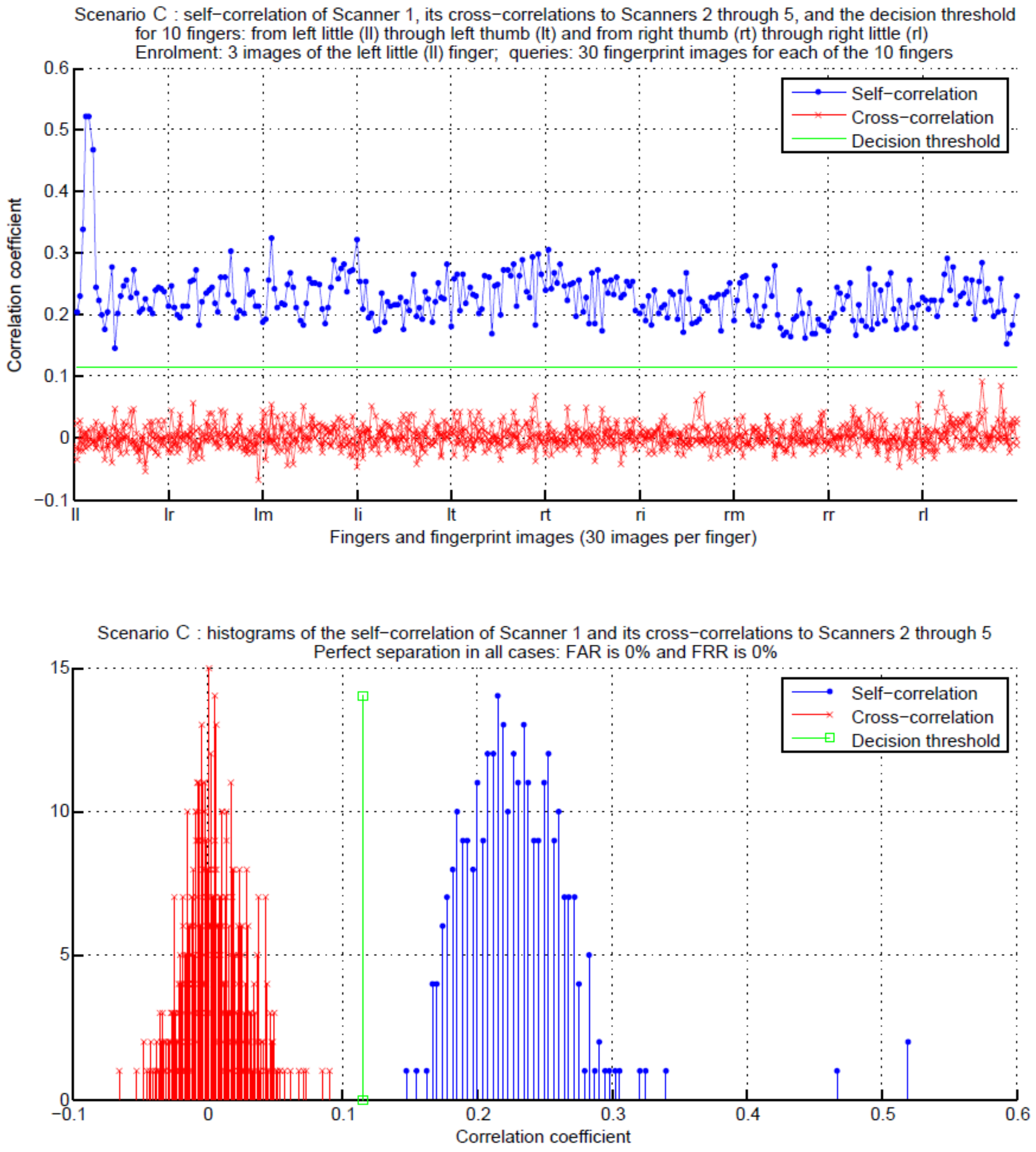
The scaling coefficient for determining the decision threshold for the UPEK scanners (1 through 4) is doubled: from 0.2 in Scenario **B** to 0.4 in Scenario **C**. The need for this can be explained with the higher sensitivity of the UPEK scanners, and since the enrolment uses images with fingerprints, the correlation coefficients computed during enrolment are smaller than in Scenario **B** because of the distortion introduced by the fingerprint pattern. To compensate for the weaker correlation, we increase the scaling coefficient for the decision threshold. The scaling coefficient for Scanner 5 (Veridicom), however, is not changed.

We tested the Scenario **C** algorithm much more extensively, using enrolment images with 5 different fingers: left little, left middle, left thumb, right index, and right ring finger, which, we believe, provides sufficient diversity as it covers all 5 types of fingers and both the left and the right hand. For enrolment images, we again used four different sets of files (for each of the 5 fingers listed above) with indices (1,2, and 3), (4, 5, and 6), (10, 11, and 12), and (28, 29, and 30). The query images are the images with **all** other fingers, not only images of the same finger used for enrolment.

The results for Scanner 1 using images with a left little finger for enrolment are shown in Figure 5.19. Comparing the plots with those of Figure 5.18 (Scenario **B**), we observed that the average self correlation is smaller and thus the margin between the self correlation and the cross correlation is smaller. Still, both the FRR and FAR are zero. The results for Scanner 1 for all 5 fingers are shown in Table 5.8, from which it is visible that most of the FRR and FAR are zero. The maximum FRR is 0.7% (meaning 2 errors in 300 tests), and the maximum FAR is 2%. These decision errors occurred because of the images with the little fingers.

The detailed results for the other four scanners we include in the appendix, Section 6.3, and here we only summarize them:

- **For Scanner 2**, the FRR increases to 2.3%, while the FAR is zero (see Table 6.2 in the appendix). If this algorithm would be used in practice in the contemplated security application, we believe that the FAR should be minimized (and ideally be 0 as in this case), while the FRR of a couple of percents



Scenario C : self−correlation of Scanner 1, its cross−correlations to Scanners 2 through 5, and the decision threshold for 10 fingers: from left little (ll) through left thumb (lt) and from right thumb (rt) through right little (rl) Enrolment: 3 images of the left little (ll) finger; queries: 30 fingerprint images for each of the 10 fingers

Scenario C : histograms of the self−correlation of Scanner 1 and its cross−correlations to Scanners 2 through 5 Perfect separation in all cases: FAR is 0% and FRR is 0%

Figure 5.19: Scenario C correlations and their histograms for Scanner 1

would be tolerable;

- **For Scanner 3** (the non-FIPS compliant UPEK scanner), the results, shown in Table 6.3 in the appendix, are noticeably worse. The FAR jumps to 6%

Query scanner ⟶	**Scnr 1**	**Scnr 2**	**Scnr 3**	**Scnr 4**	**Scnr 5**
Indices of the files used for enrolment ⇓	***FRR, %***	FAR, %	FAR, %	FAR, %	FAR, %
Left little finger					
1, 2, and 3	***0***	0	0	0	0
4, 5, and 6	***0***	0	0	0	0
10, 11, and 12	***0***	0.7	0	0	0
28, 29, and 30	***0***	0	0	0	0
Left middle finger					
1, 2, and 3	***0***	0	0	0	0
4, 5, and 6	***0***	0	0	0	0
10, 11, and 12	***0***	0	0	0	0
28, 29, and 30	***0***	0	0	0	0
Left thumb finger					
1, 2, and 3	***0.3***	0	0	0	0
4, 5, and 6	***0***	0	0	0	0
10, 11, and 12	***0.7***	0	0	0	0
28, 29, and 30	***0***	0	0	0	0
Right index finger					
1, 2, and 3	***0***	0.3	0	0	0
4, 5, and 6	***0***	0	0	0	0
10, 11, and 12	***0***	0	0	0	0
28, 29, and 30	***0***	2.0	0	0	0
Right ring finger					
1, 2, and 3	***0***	1.3	0	0	0
4, 5, and 6	***0***	0	0	0	0
10, 11, and 12	***0***	0	0	0	0
28, 29, and 30	***0***	0	0	0	0

Table 5.8: Scenario C results: FAR and FRR for Scanner 1 (parameters 55/0.40)

on one occasion and this is with a scanner of a different type (the FIPS certified Scanner 1). The FAR with the scanner of the same type, Scanner 4, however, is still small – about 2%. Also, in most cases the FAR is again zero. Unfortunately, the FRR is here is again high and peaks at 6.3%;

- **For Scanner 4**, the FRR, shown in Table 6.4 in the appendix, is also comparable to that of Scanner 3, but the FAR is much better, with a maximum of 1.7%;

- **For Scanner 5**, the results, shown in Table 6.5 in the appendix, are also very good: maximum 1.3% for the FAR and maximum 1.7% for the FRR.

It is important to note that the high FAR and FRR in the results above are for the case when the scanner enrolment uses images with one finger, while the scanner verification uses (one) image with another finger; nevertheless, they are still very good. When one and the same finger is used both for the scanner enrolment **and** for the scanner verification, the results are nearly perfect.

5.2.4.4 Scanner Authentication Scenarios: Discussion

The first and the most important conclusion from the excellent results in Scenario **A** and the very good results in Scenarios **B** and **C** is the choice of the correlation (coefficient) for scanner pattern matching. Correlation is a robust method for estimating the similarity between signals in the presence of high level of noise and distortion with unknown characteristics, but in order to be reliable, correlation requires vectors with large sizes. Our application is very well suited for this because it operates on images and thus data samples with large sizes are available.

Nevertheless, since the characteristics of the noise and especially of the distortion caused by the fingertip skin were difficult to be estimated and quantified at this early stage of our research, the algorithms of Scenarios **B** and **C** occasionally produce high levels of decision errors. As the major cause for this we identified the unreliable estimate for the mean of the distribution of the self correlation, particularly in Scenario **C** where the enrolled images contain fingerprints. Since the distribution of the cross correlation is usually and typically centered around 0, improving the accuracy of the estimate for the mean of the self correlation is sufficient to significantly improve the detection performance.

A major challenge for the algorithms of Scenarios **B** and **C** were the little fingers, and generally, fingers which have shallow and narrow valleys. From a practical point of view, however, this would not be a big problem because the index finger, the middle finger, and occasionally the thumb are typically used in fingerprint authentications. When such fingers are used for the scanner enrolment, the results are nearly perfect.

Although not fully optimized and tested on a large number of scanners and fingers, these early algorithms for scanner authentication in Scenarios **A**, **B**, and **C** are important because of several reasons:

1. The approach is applicable to any type of capacitive scanners and does not require knowledge of the specific acquisition technology being used. We also believe that this method can be extended to other types of scanners, using, for

example, RF imaging. The only requirement is the sensing method to provide pixels that are little influenced by the fingertip skin or not influenced at all;

2. The algorithms have very low complexity. Computing the correlation coefficient with the chosen ROI takes a couple of thousand integer multiplications-and-additions (in the worst case), the whole processing is very simple and does not require any kind of filtering or transforms. The rest of the operations are comparisons, integer searches, and indexing. This fulfills the objective for a low-complexity implementation;

3. The algorithms are inherently random because the attacker has no knowledge about the parts of the enrolled images that will be used for scanner pattern matching (the attacker has control only on the query image);

4. Arguably, however, the most important benefit for us from developing the algorithms for the three scenarios was the qualitative understanding of the process and the proof of the concept, both of which were indispensable for the development of our advanced algorithms.

Besides the possible obvious performance improvements of the algorithm, such as testing it on a large number of scanners and fingers to better optimize the parameters, one can use a more accurate detection of the regions with valleys by incorporating already developed algorithms for that instead of the simple threshold-based algorithm that does not take into account the fingerprint pattern (and the shapes and continuity of the valleys). However, we opted not to perfect these early algorithms because this approach has inherent limitations that, we believe, will be difficult to overcome. The most important is its general security weakness: the simple way of isolating/extracting the scanner pattern from the composition of the scanner pattern and the fingerprint pattern by using only a small part of the valleys creates a vulnerability as these pixels can be identified relatively easily by an attacker and replaced without distorting the fingerprint pattern sufficiently so that the biometric verification can result in a biometric non-match decision; thus, the attack may pass undetected by the bipartite verification (see Figure 5.3).

5.2.5 Simple Algorithm Using Wavelets

The very good performance of the algorithms of Scenarios **A**, **B**, and **C**, described in the previous sections, and the resulting conclusions were very important for the development of another algorithm, disclosed in [Ivanov and Baras US'907] and [Ivanov and Baras 2011], using more sophisticated signal processing. The algorithm is still simple, yet extremely accurate, and is able to distinguish one fingerprint scanner from another scanner of exactly the same manufacturer, type, and model using only a single image, acquired with each scanner. The algorithm extracts scanner patterns from the two images using wavelets, selects parts of these patterns, and computes the correlation coefficient as a similarity score between them. Each of the two images contain a fingerprint pattern in it and thus the algorithm falls in the framework of Scenario **C**. We tested the algorithm on 2,160 images acquired with 24 capacitive fingerprint scanners of exactly the same model (see the appendix, Section 6.1), and based on the histograms, we computed a decision threshold and estimated the equal error rate (EER).

5.2.5.1 Algorithm

The algorithm does not assume any specific model for the scanner pattern and the fingerprint pattern in the composite signal (i.e., the image pixels). Our only two assumption are: (1) the scanner pattern is mainly caused by non-idealities and variability in the sensing matrix and the subsequent signal processing within the fingerprint scanner, and (2) the scanner pattern is revealed in the pixels that experience little or no influence from the fingerprint pattern and therefore these pixels need to be only located and used. The algorithm consists of 3 steps and uses 2D wavelets for scanner pattern extraction and correlation coefficient for matching. Thus, it is similar to the algorithm of Scenario **C** described in the previous section but uses less heuristics and more powerful signal processing.

The conceptual diagram of the sequence of signal processing modules is shown in Figure 5.20. The signal **g**, the image, is processed to produce the signal d, the scanner verification decision.

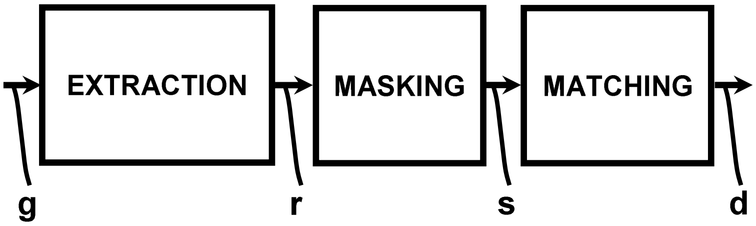


Figure 5.20: Conceptual diagram of the simple algorithm using wavelets

Let $g_e(i, j)$ and $g_q(i, j)$ be the pixel values at row i and column j of the two acquired images, where the subscript e (from *enrolled*) refers to one of the two images and the subscript q (from *query*) to the other image. This referencing is conditional because all processing is the same for each image.

1. *Wavelet extraction.* Each image is decomposed using 2D wavelets and then reconstructed by setting the LL-subband coefficients to 0, yielding the signals $r_e(i,j)$ and $r_q(i,j)$. The biorthogonal wavelets with decomposition order 5 and reconstruction order 1 gave the best results, but other wavelets, e.g., Daubechies or symlets, both of order 2 (4-tap filter length), also work well.

2. *Masking.* We observed that selecting only some of the pixels based on the magnitude of their values from $r_e(i,j)$ and $r_q(i,j)$ is necessary. Therefore, the signal $s_e(i,j)$ (and similarly $s_q(i,j)$) is constructed using:

$$s_e(i,j) = \begin{cases} r_e(i,j) & \text{if } |r_e(i,j)| \leq \theta \\ \\ NU & \text{otherwise} \end{cases} \tag{5.4}$$

   where NU denotes a mark that the corresponding pixel will not be used for further processing. We achieved the best results with $\theta = 4$, but 3 or 5 is also possible.

3. *Correlation matching.* Similarly to Scenarios **A**, **B**, and **C**, here we also use the correlation coefficient as a matching score:

$$corr(\mathbf{t_e}, \mathbf{t_q}) = \frac{(\mathbf{t_e} - \bar{\mathbf{t_e}}) \cdot (\mathbf{t_q} - \bar{\mathbf{t_q}})}{\|\mathbf{t_e} - \bar{\mathbf{t_e}}\| \, \|\mathbf{t_q} - \bar{\mathbf{t_q}}\|}, \tag{5.5}$$

   where $\mathbf{t_e}$ and $\mathbf{t_q}$ are vectors derived from $s_e(i,j)$ and $s_q(i,j)$, respectively, by taking only the common useful pixels (i.e., the pixels marked with NU are discarded) and then ordering the common useful pixels in vector form. $\bar{\mathbf{t_e}}$ and $\bar{\mathbf{t_q}}$ are the means of the elements of vectors $\mathbf{t_e}$ and $\mathbf{t_q}$, respectively. The decision d is 1, i.e., match, if $corr(\mathbf{t_e}, \mathbf{t_q})$ is greater than a predetermined decision threshold (discussed next); otherwise it is 0, i.e., nonmatch.

5.2.5.2 Results

We acquired raw images with 24 UPEK area scanners: u101 through u122, u151, and u152 (see Section 6.1 in the appendix).

With each scanner, we acquired 30 images for three fingers: an index, a thumb, and a little finger of one person, with each set having $(24 \cdot 30) = 720$ images per finger, giving a total of $(3 \cdot 720) = 2,160$ images for all 3 fingers. Each image has $360 \cdot 256$ pixels, with pixel values ranging from 0 to 255 (8 bits per pixel).

Images of one and the same finger

We first applied the algorithm to images with fingerprints of one and the same finger, for which we chose the index finger because typically it is used for biometric authentication. For $g_e(i,j)$ and $g_q(i,j)$, we chose each of the 720 images in the set, yielding to $720 \cdot (720 + 1)/2 = 259,560$ comparisons. The normalized (integrating to 1) histograms of the self correlation coefficients (when the two images were acquired with the same scanner) and cross correlation coefficients (when the two images were acquired with two different scanners) are shown in Figure 5.21.

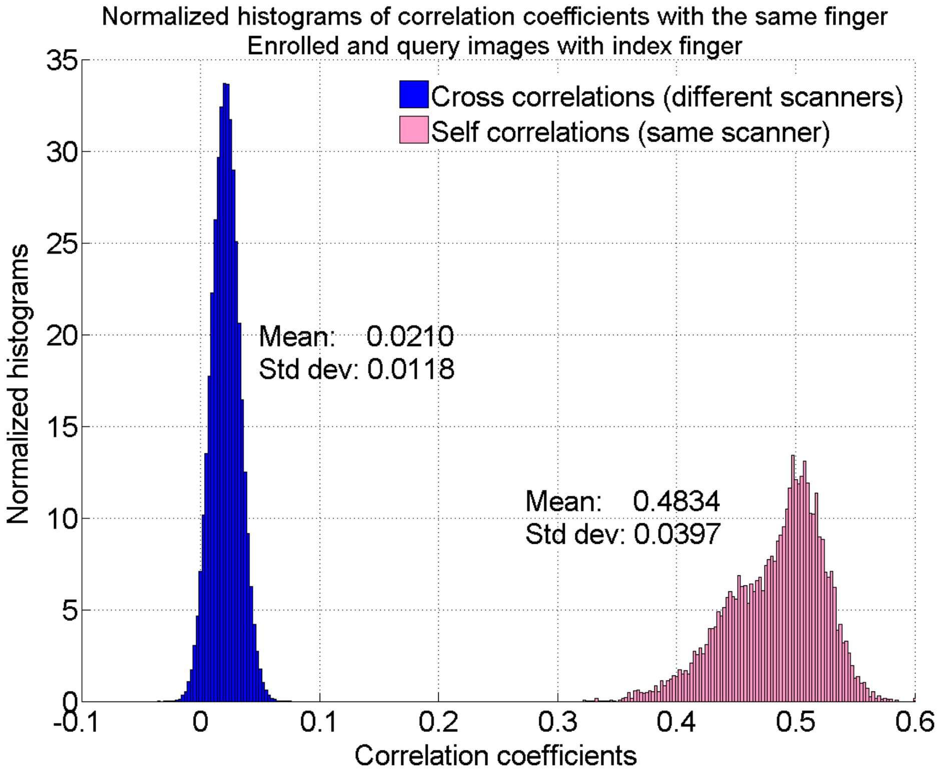


Figure 5.21: Normalized histograms of the correlation coefficients when both $g_e(i, j)$ and $g_q(i, j)$ are with index finger

Images of two different fingers

To demonstrate that the proposed algorithm does not depend on the finger, next we applied it to a set of pairs of images where one of them contains one finger and the other image contains a different finger. It is known that thumbs typically have much wider ridges and valleys than little fingers (of the hands of one and the same person). Also typically, index fingers have narrower ridges and valleys than thumbs, and wider than little fingers. The histograms of the correlation coefficients where $g_e(i, j)$ are images with the index finger and $g_q(i, j)$ are images with the thumb are shown in Figure 5.22, and the results where $g_e(i, j)$ are images with the thumb and $g_q(i,j)$ are with the little finger are shown in Figure 5.23. As in the case above when both images are of the same finger, each set for a finger contains 720 images, thus yielding $(720 \cdot 720) = 518,400$ comparisons in each case (because here the two fingers are different). Since all processing is symmetric, the choice which finger is in $g_e(i,j)$ and which in $g_q(i,j)$ is immaterial.

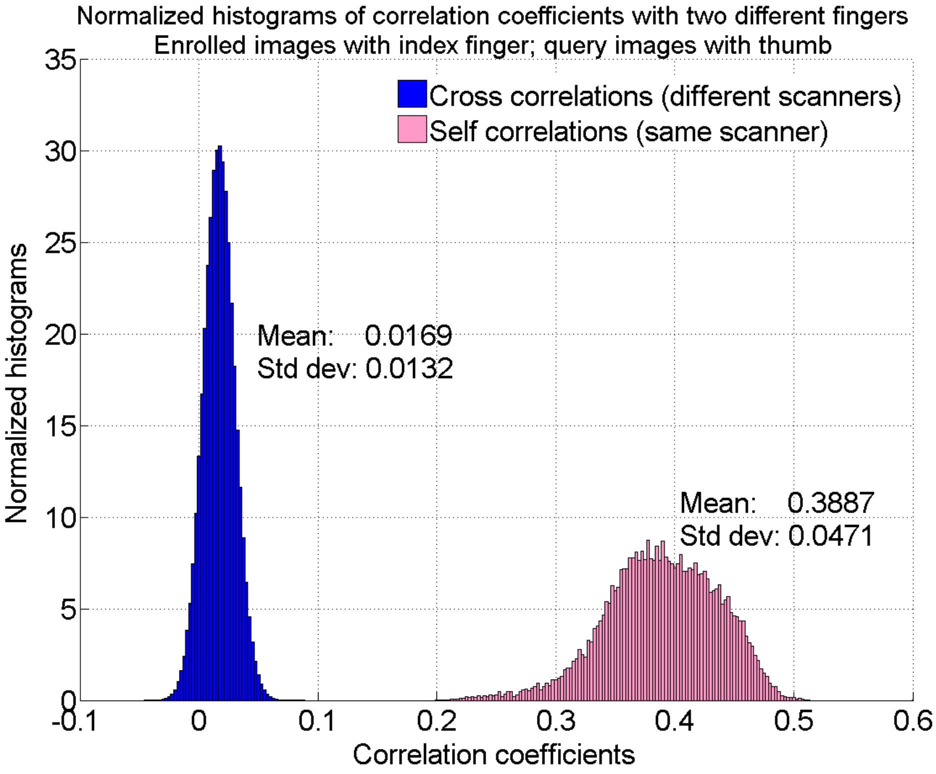


Figure 5.22: Normalized histograms of the correlation coefficients when $g_e(i,j)$ is with index finger and $g_q(i,j)$ is with thumb

5.2.5.3 Decision threshold and error rate

No decision errors with a threshold chosen anywhere roughly between 0.1 and 0.2 were registered for any of the 1,296,360 comparisons, which is also visible from

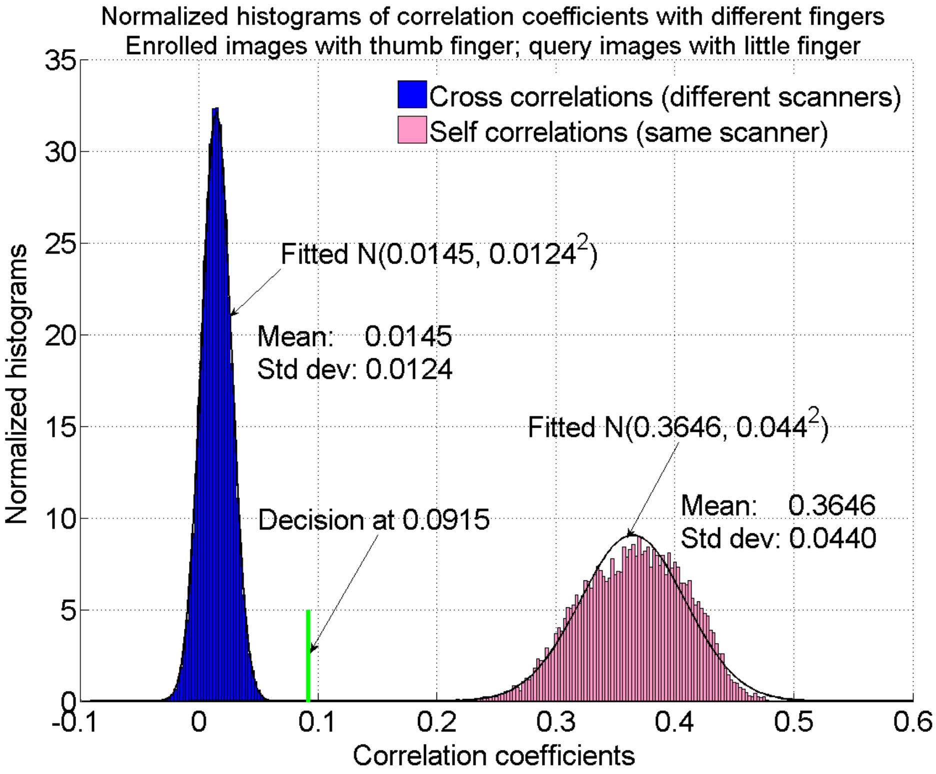


Figure 5.23: Normalized histograms of the correlation coefficients when $g_e(i,j)$ is with thumb and $g_q(i,j)$ is with little finger

the clear separation of the histograms in the three figures and the large distance between the centers of the two distributions (of the self and cross correlations). We computed an estimate for the equal error rate (EER), when FAR = FRR, for the third case (which is the worst one of the three) by fitting Gaussian PDFs (see Figure 5.23). The histogram with the cross correlation coefficients fits extremely well with $N(0.0145, 0.0124^2)$, and $N(0.3646, 0.044^2)$ well approximates the histogram with the self correlation coefficients. Based on the fitted PDFs and the computed decision threshold of 0.0915, the EER is $2.8 \cdot 10^{-10}$.

Although simple and extremely accurate, the algorithm using 2D wavelets described above has several disadvantages. First, it requires a wavelet analysis of two-dimensional signals (the images) and wavelet reconstruction, which may be too computationally demanding in portable devices, especially if required to be computed within a very short time period; the algorithm also uses all pixels in the image, which also adds a considerable computational burden. Next, although operating on images containing fingerprints, we observed that this algorithm relies heavily on the areas of the image that are not covered by the fingerprint pattern, e.g., close to the image edges. The problem is that the pixels in these areas can be relatively easily detected and replaced by an attacker with pixels taken from another image that has been acquired with the authentic scanner. Therefore, the wavelet algorithm may fail to detect the doctored in this way image, leading to a security breach. And finally, the robustness of the algorithm under different environment conditions, e.g., wide range of temperature variations, moisture, and fingertip pressure on the scanner, is unclear. To overcome or completely avoid these problems, we developed the advanced algorithm, presented in Section 5.5.

5.3 Image Acquisition and Signal Models

The actual function describing the relationship among the scanner pattern, the scanner noise, and the fingerprint pattern (when present) can be very complex. This function depends on the particular fingerprint sensing technology and on the particular fingerprint scanner design and implementation, which are usually proprietary. Furthermore, even if the exact function is known or determined, using it for estimating the scanner pattern may prove mathematically intractable or require computationally intensive and extensive signal processing. However, this function can be simplified into a composition of additive/subtractive terms, multiplicative/dividing terms, and combinations of them by taking into account only the major contributing factors and by using approximations. This simple, approximate model of the actual function we call *signal model*. We performed our analysis of the actual scanner implementations (in particular at semiconductor level) only from publicly available information and made certain assumptions about the dominant factors which determine the scanner pattern. For this reason, it is difficult to provide a rigorous proof for the validity of the models; instead, we can only judge about the accuracy of the signal models indirectly: by comparing the processed images with the expected resulting signals according to the models.

The first, and critical, part of the analysis is studying the sensing process, i.e., the conversion from a ridge/valley into electrical signal. We studied the capacitive sensing technologies of two companies: Veridicom, which was one of the first capacitive-sensing technology and was later acquired by Fujitsu, and of UPEK, whose scanners are the most widely used capacitive scanners.

5.3.1 UPEK Scanners

Our assumption is that the UPEK scanners most widely sold and also present in our lab are designed as active capacitive feedback circuits with enhanced fingerprint detection, as disclosed in [UPEK Patent US'381]. This assumption is supported by the success of the algorithms we developed and the microscopic photographs we took of the scanners (discussed later).

A major problem for all capacitive sensors developed by other companies before UPEK had been their poor sensitivity in distinguishing between a ridge and a valley. The technique UPEK developed significantly increased this sensitivity by combining two effects: a plate effect and a fringing effect. Hereby we present the principle of operation, a summary of the formulas as given in [UPEK Patent US'381], and our analysis and models of pixel values in an image as a function of the scanner pattern and fingerprint pattern.

Figure 5.24 shows two adjacent sensing cells and Figure 5.25 shows the equivalent electrical circuit of one sensing cell. The input voltage change ΔV_{in} in Figure 5.24 is applied at terminal 99 which is in direct electrical contact with the body of the person (e.g., their fingertip) and acts as an input capacitor by providing a variable charge transfer and determines the plate effect. C_{plate} is the capacitance being sensed between plate 24 (and 23) and the finger 18, and it varies depending on the part of the skin present at that cell. In case of a ridge, C_{plate} is large (cell

2a); in case of a valley, C_{plate} is small (cell 2b). Thus, it is convenient to represent their relationship as: $C_{plate\ ridge} = C_{plate\ valley} + \Delta C_{plate\ ridge}$.

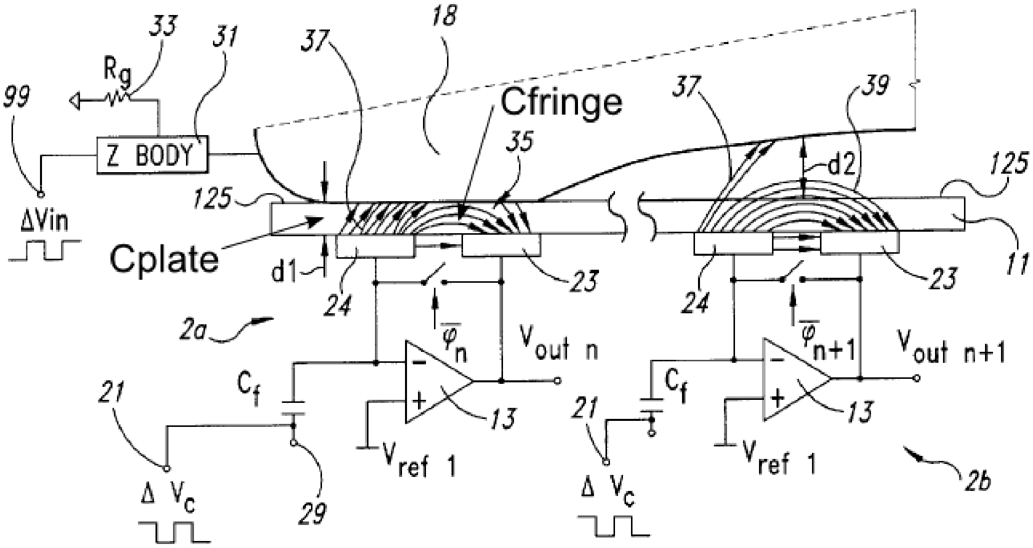


Figure 5.24: UPEK scanner diagram (source: [UPEK Patent US'381])

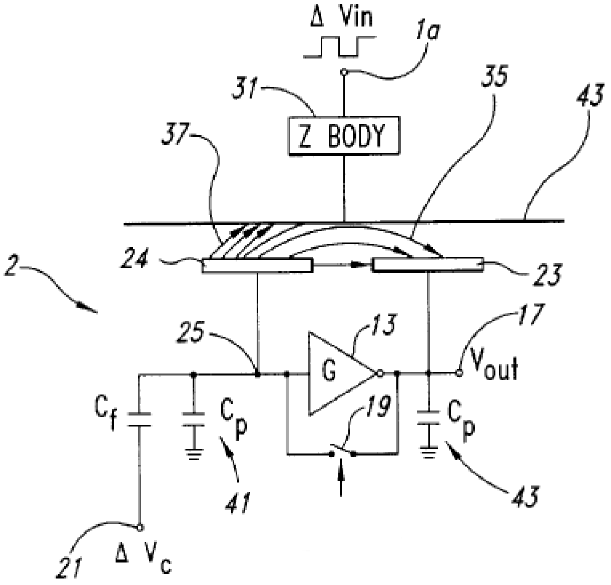


Figure 5.25: UPEK scanner equivalent electrical circuit (source: [UPEK Patent US'381])

The fringing effect is exhibited by a change in the capacitance between plates 23 and 24. A ridge reduces the fringing capacitance C_{fringe} because a conductor adjacent to the plates short circuits some of the field lines between the two plates. If a valley is present, however, C_{fringe} is greater because a valley essentially is an air gap. Thus, $C_{fringe\ valley} = C_{fringe\ ridge} + \Delta C_{fringe\ valley}$. In the ideal case when the skin in the valley is far enough from the plates 23 and 24, $C_{fringe\ valley}$ is close to the measured capacitance when no object is applied to the scanner platen, i.e., with air. The effect of C_{fringe} is amplified by the differential amplifier 13 (which usually is implemented as a high-gain inverter, see Figure 5.25) and behaves as $(1+G)C_{fringe}$ at node 25 due to the Miller effect.

Therefore, the presence of a ridge creates two opposite effects in C_{plate} and C_{fringe}: it increases C_{plate} and decreases C_{fringe}. A valley does exactly the opposite: it decreases C_{plate} and increases C_{fringe}. Hence, $C_{plate\ ridge} > C_{plate\ valley}$ and $C_{fringe\ ridge} < C_{fringe\ valley}$. This change in opposite directions of the two types of capacitances determines the enhanced sensitivity of the UPEK scanners and their ability to well distinguish between a ridge and a valley. The parasitic capacitance is modeled as C_p (see Figure 5.25). Using information from [UPEK Patent US'381], we derived an expression for the change in the output voltage at terminal 17 (in the same figure) in function of the parameters of interest as:

$$\Delta V_{out} = -\frac{\Delta V_{in} C_{plate} + \Delta V_c C_f}{\frac{C_f}{G} + \frac{C_p}{G} + \frac{C_{plate}}{G} + (\frac{1+G}{G})C_{fringe}} \tag{5.6}$$

Devising a signal model directly from Expression 5.6 and then developing signal processing algorithms that separate the scanner pattern from the fingerprint pattern proved to be very difficult. Furthermore, since we did not have information about the typical absolute or relative values (even on the order of magnitude), we had to speculate/hypothesize, to make various approximations, assuming dominance of certain terms over the other terms, and to study the accuracy of each one of the models and its suitability for our purposes. The model that worked best and the underlying assumptions in it are described next.

Taking into account that voltage ΔV_c, applied at terminal 21 (in Figures 5.24 and 5.25), is normally the inverse of ΔV_{in}, i.e., ΔV_c = -ΔV_{in}, we rearrange Expression 5.6 as follows:

$$\Delta V_{out} = -\frac{\Delta V_{in} G (C_{plate} - C_f)}{C_f + C_p + C_{plate} + (1+G)C_{fringe}} =$$

$$= -\frac{\Delta V_{in} G (1 - \frac{C_f}{C_{plate}})}{\frac{C_f}{C_{plate}} + \frac{C_p}{C_{plate}} + 1 + (1+G)\frac{C_{fringe}}{C_{plate}}} \qquad (5.7)$$

Next, we make the following assumptions for the terms in Expression 5.7:

- C_{plate} represents the fingerprint pattern and therefore it cannot be a constant. However, it appears that C_{plate} is always much larger (at least one order of magnitude) than C_f. Therefore, the term $\frac{C_f}{C_{plate}}$, both in the nominator and in the denominator, can be neglected as it is much smaller than 1, the additive constant present both in the nominator and in the denominator;

- A similar reasoning as of C_f also applies to C_p. Hence $\frac{C_p}{C_{plate}}$ in the denominator can be neglected as it is much smaller than 1;

- G is the gain of the differential amplifier 13 in Figures 5.24 and 5.25, which gain is typically very large (several orders of magnitude larger than 1), and therefore $(1 + G) \approx G$. Furthermore, it is well known that making the gain of a differential amplifier (and also of an opamp) precise and uniform across all amplifiers, even in the same batch/die, is practically impossible. Finally, each sensing cell has its own, dedicated amplifier. All this reasoning leads to the credible assumption that the gain G is a sufficiently dominant and unique factor for each sensing cell and let us believe that the scanner pattern is mainly determined by the gain G. However, it is also possible that the scanner pattern is simply (approximately) proportional to the gain G, not exactly equal to it.

Applying the above assumptions to Expression 5.7, we obtain the approximate expression:

$$\Delta V_{out} \approx -\Delta V_{in} \frac{G}{1 + G(\frac{C_{fringe}}{C_{plate}})} \tag{5.8}$$

As already explained above, the fingerprint pattern is represented by C_{plate} and C_{fringe}, the value of either of which depends on the valley or ridge positioned above this particular sensing cell. ΔV_{in} (and ΔV_c) is the excitation voltage, which, although possibly noisy, is the same for all sensing cells. The excitation voltage applied to each sensing cell may also depend on the physical location of the cell with respect to the common voltage source, i.e., the chip layout. However, such dependence most likely will result in a common change for a series of cells (e.g., in a row or in a column), yielding to a gradient effect, but not influencing each sensor cell in a unique way. Therefore, we can assume that ΔV_{in} (along with the minus sign) is a common scaling constant that eventually is factored out in the conversion from the (analog) voltage ΔV_{in} into a (digital) pixel value. Finally, since this is an active electronic circuit, it is reasonable to assume that ΔV_{out} contains also an additive (thermal) noise and quantization noise (from the A-to-D conversion). Thus, we transform Expression 5.8 into the following signal model, henceforth referred to as Signal Model A:

$$g(i,j) = \frac{s(i,j)}{1 + s(i,j)f(i,j)} + n(i,j,t) \tag{5.9}$$

where $g(i,j)$ are the pixel values of the image at row index i and column index j, $f(i,j)$ is the fingerprint pattern, $s(i,j)$ is the scanner pattern, and $n(i,j,t)$ is the scanner noise, which also depends on the time t because the scanner noise is time varying (by definition). All operations in Expression 5.9, i.e., the addition, the multiplication, and the division, are element by element (i.e., pixel by pixel) because our study led to the conclusion that the point spread function (PSF) of these fingerprint scanners, viewed as a two-dimensional linear space-invariant system, can be well approximated by a Dirac delta function. The range of $g(i,j)$ is from 0 to 255 grayscale levels (8 bits/pixel), although some scanner implementations may produce narrower range of values. An implicit assumption in Expression 5.9 is that the pixel values $g(i,j)$ as saved in a computer file are not further enhanced (or compressed) by image processing algorithms in order to facilitate the biometric authentication, or are enhanced (or compressed) but the scanner pattern information in them is not substantially altered or destroyed.

To corroborate our claims, in Figure 5.26 below we show the schematics of one possible implementation (taken from [UPEK Patent US'381]) where the voltage ΔV_{in} is applied via electrode 92 and provides voltage to many sensing cells 2. Figure 5.27 shows a top view of the scanner platen with two conducting cells: element 30, providing ΔV_{in}, and element 31, providing ground (the exact type of electrical signal applied depends on the timing of the sensing operation and the location of the cell).

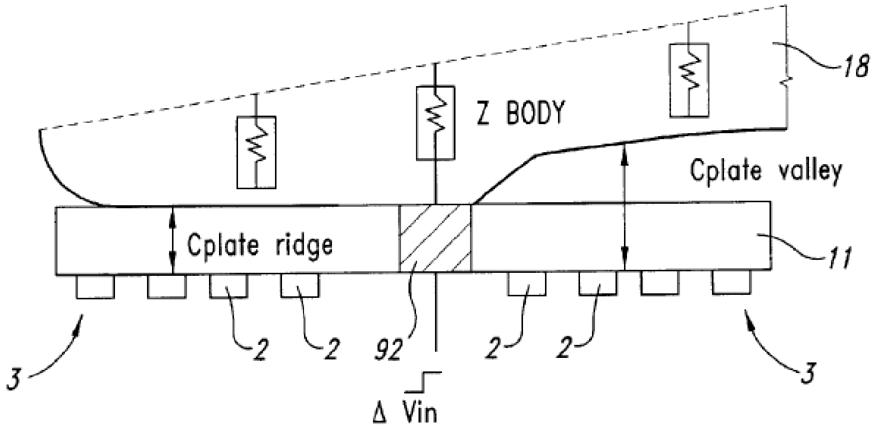


Figure 5.26: UPEK scanner conductor (source: [UPEK Patent US'381])

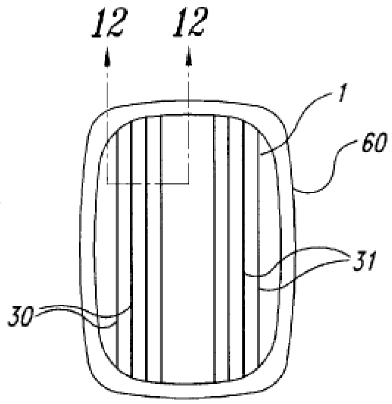


Figure 5.27: UPEK scanner platen (source: [UPEK Patent US'381])

Finally, Figure 5.28 shows two microscopic photographs of the surface of the platen of a UPEK scanner, which photos we took specifically for our study. The squares 1 are the sensing cells, and the stripe 2 is the conductor that provides ΔV_{in}. On the right side of the same figure, one sensing cell is enlarged, and 4 and 5 point to the two capacitive plates 23 and 24 of Figures 5.24 and 5.25.

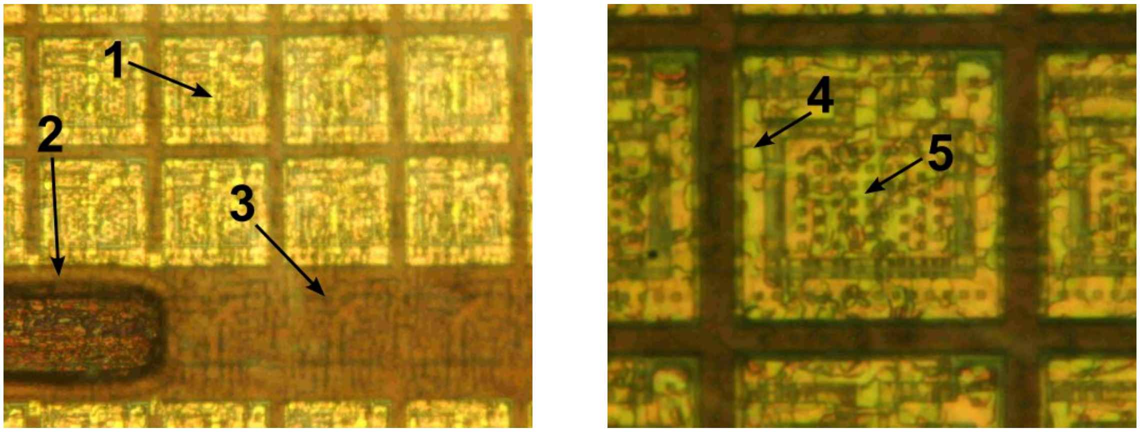


Figure 5.28: Microscopic photos of UPEK scanner surface (left) and cell (right)

Details about the UPEK scanners used are provided in Section 6.1 in the appendix.

5.3.2 Veridicom Scanners

In this section we discuss and summarize [Veridicom Patent US'620], which we believe describes the design of their scanners. We have used only one Veridicom scanner (referred in the text as "v1") because at the time we started this research, the scanner product line has already been discontinued, and therefore purchasing more scanners of the type proved impossible. Details about this Veridicom scanner are provided in Section 6.1 in the appendix.

Nevertheless, we did study the Veridicom scanners because: (a) their principle of operation is much simpler than that of UPEK, which gave us important initial insight and ideas, and (b) the Veridicom scanners still have considerable market share of already deployed capacitive scanners and therefore offer a large base for application of our technology.

The Veridicom fingerprint sensing technology is based on determining the value of the capacitance between a single metal plate 120 (see Figure 5.29, left), being one of the electrodes, and the finger 160, serving as the other electrode. This capacitance is inversely proportional to the distance between the plate and the finger surface ($C = \epsilon A/d$), and thus it varies with the finger topography: the capacitance C_{ridge} between the plate 120 and the ridge 180 is greater than C_{valley}, which is the capacitance between the plate 120 and the valley 190, or $C_{ridge} > C_{valley}$. The depth of a valley is typically about 100 μm. The metal plates have constant areas (A) and are covered by an insulating material (SiO_2, glass, or plastic) of several microns.

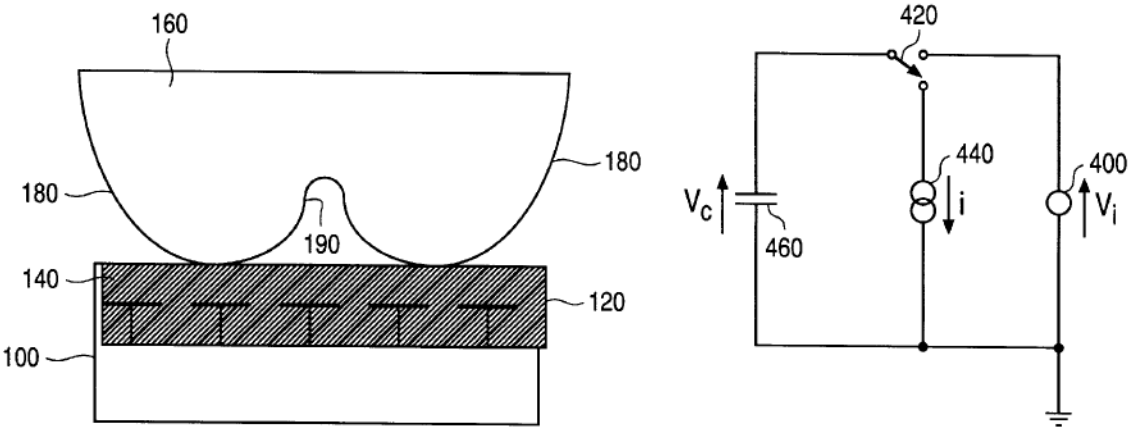


Figure 5.29: Veridicom scanner and principle of operation (source: [Veridicom Patent US'620])

If C_f is the capacitance between the metal plate 120 and the finger surface

160, then C_f is the serial combination of the capacitance C_{fi} between the plate 120 and the top of the insulator 140 (with insulator thickness d_i and dielectric constant ϵ_i) and the capacitance C_{fa} between the top of the insulator 140 and the finger 160 (with distance d_a and dielectric constant ϵ_a), and therefore $C_f = C_{fi} \parallel C_{fa} = C_{fi}C_{fa}/(C_{fi}+C_{fa})$. For a ridge that touches the insulator 140, $d_a = 0$, and therefore $C_{f\ ridge} = C_{fi}$. For a valley, $d_a >> d_i$ and thus $C_{f\ valley} \approx C_{fa}$.

Each metal plate 120 has also a parasitic capacitance C_m to the grounded substrate. Therefore, the equivalent capacitance to ground is $C_{eq} = C_m + C_f$, which means that (variations in) C_f will be largely masked by the parasitic capacitance C_m. This explains the lower sensitivity of the Veridicom scanners with respect to the sensitivity of the UPEK scanners.

The principle of operation is shown on the right side of Figure 5.29. The capacitance C_{eq} (460) is measured by precharging it to a predetermined voltage V_i (400), then turning the switch 420, removing from the capacitance a fixed charge $\Delta q = I_c \Delta t$ by drawing a fixed current I_c over a fixed time interval Δt, and finally measuring the final voltage V_f across it. As noted further in the patent, “mapping the voltage variation across the array of capacitors without converting to distances provides a relative measurement of the capacitance variation and is sufficient to create an image.” If $\Delta V = (V_i - V_f)$ is the measured voltage difference, then $C_{eq}\Delta V = (C_m + C_f)\Delta V = I_c \Delta t$ and the measured voltage (difference) is:

$$\Delta V = \frac{I_c \Delta t}{C_m + C_f} \tag{5.10}$$

Figure 5.30 shows a detailed circuit diagram of a single sensor cell (within the square with the dashed line), the current source on the left-hand side of it, and the read-out circuitry on the right. The current source contains three stages. The first one (transistor 606 and resistor 602) is common for the entire chip, and it sets the reference current I_{REF}. The second stage (transistors 608 and 610) is replicated in each column of the sensor array and mirrors the current I_{REF} so that I_0 is common for all sensor cells in one column. Therefore, variations in I_0 will affect in a similar way all sensor cells in a column. The third stage (transistors 607 and 514, inside the square) is replicated in each sensor cell, and the current I_c mirrors the current I_0 from the second stage (a ratio of 2:1 is used between the stages). Therefore, variations in I_c will be local for each sensor cell and thus can determine the scanner pattern. Furthermore, the scanner pattern in each column of sensor cells is expected to exhibit correlation because all of the cells' reference currents I_c mirror the single current source I_0 of that particular column.

Within each sensor cell (the square with the dashed line), the equivalent capacitance C_{eq} of the metal plate 500 has two components: capacitance C_f (508) to the finger and parasitic capacitance C_m (502). During the measurement of C_{eq}, the CMOS switch (between signals 524 and 532 and enabled by them) connects the metal plate 500 to the charge input node. The switch transistor 514 is enabled by the precharge signal 520 and thus precharges C_{eq} to VDD. Then the transistor 514 is turned off and the current I_c discharges C_{eq}. The voltage on the metal plate 500, through the gate of transistor 512 and the switch transistor 516, enabled by the row signal 530, becomes the output voltage V_{OUT} (526). Therefore, all capacitances of one row of metal plates are measured in parallel.

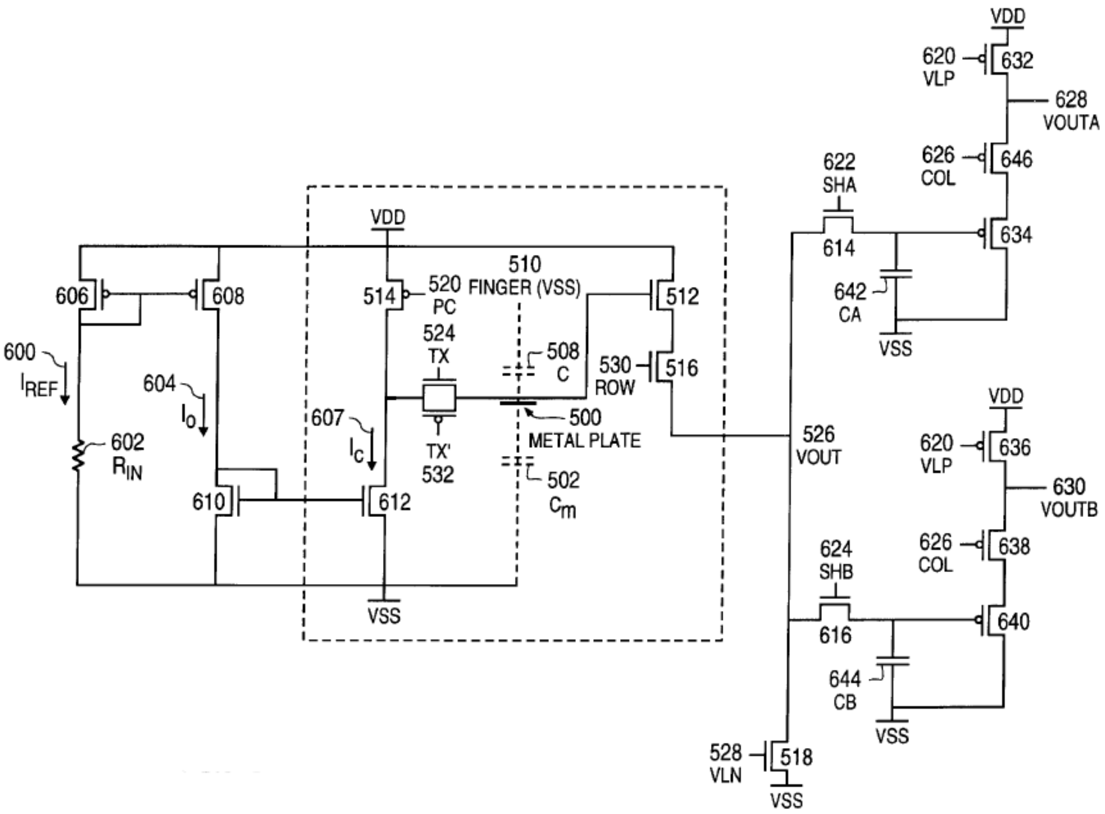


Figure 5.30: Veridicom scanner sensor element schematics (source: [Veridicom Patent US'620])

The read-out part (on the right-hand side of Figure 5.30) is common for a column of sense cells in the array and contains two sample-and-hold circuits, each one consisting of a sample-and-hold switch, a capacitor, and a source follower. The first sample (of voltage) is stored immediately after precharging C_{eq}, and the second sample is stored after a fixed interval of time Δt. The sample values are scanned differentially by enabling the column selection switches.

This construction of common-row and common-column components and signals suggests strong row-wise and column-wise correlations in the scanner pattern, which was indeed observed along the two main frequency axes in the 2D DFT of an image acquired with air the Veridicom scanner (see Figure 5.6 in Section 5.2.2).

To model the relationship between the fingerprint and the scanner patterns, we speculate the following:

- Since I_c is the value of the discharging current and is specific for each sensor cell, it can determine the scanner pattern. I_c may also experience random variations in time, i.e., to be noisy. The other term in the nominator of Expression 5.10 is Δt, the time interval between the openings of the first and the second sample-and-hold output gates. Δt is predetermined and fixed, but the time for which the two transistor gates switch may be specific for each sensor cell. Similarly, Δt may also experience random variations in time, i.e., to be noisy. In sum, the combined term $I_c \Delta t$ can form the scanner pattern.

- The capacitance between the metal plate and the finger surface is $C_f = C_{fi} \parallel C_{fa}$, where C_{fa} is the capacitance between the top of the insulator and the skin surface, and C_{fi} is the capacitance between the metal plate and the surface of the insulator. C_{fi} is also sensor-cell specific and potentially may form the scanner pattern, although its variations are probably small in comparison with the variations of C_{fa} as the thickness of the insulator is well controlled. Furthermore, separating C_{fi} from C_{fa} is difficult because they are in series. For these reasons, we assume C_{fi} to be constant across the sensor array, and then C_f represents only the fingerprint pattern.

- C_m is the parasitic capacitance to the grounded substrate, and therefore its value may be sensor-cell specific. Furthermore, variations of C_m across the area of the scanner platen are very well possible to exist and to potentially form gradient effects, which effects, however, are unlikely to contribute substantially to the formation of unique differences among adjacent sensor cells that can constitute a scanner pattern. Finally, from a physical construction perspective, the relative variations of the parasitic capacitance C_m among adjacent sensor cells are likely to be much smaller that the absolute value of C_m (which is typically large). Thus, the sensor-cell specific variations of C_m can potentially be neglected and C_m be assumed to be constant (barring possible gradient effects) for all sensor cells.

- Since this is an active electronic circuit and also ΔV is quantized, there will be also an additive noise.

Next, we rearrange Expression 5.10 as:

$$\Delta V = \left(\frac{1}{C_m} \right) \frac{I_c \Delta t}{1 + \frac{C_f}{C_m}} \tag{5.11}$$

It is not necessary to pinpoint which term in the nominator $I_c \Delta t$ forms the scanner pattern as the term that is constant will be factored out during the A-to-D conversion, similarly to the term $\frac{1}{C_m}$. And finally, assuming that the pixel values of the acquired image are proportional to ΔV, we arrive at Signal Model B:

$$g(i, j) = \frac{s(i, j)}{1 + f(i, j)} + n(i, j, t) \tag{5.12}$$

where, similarly to Signal Model A, $g(i, j)$ are the pixel values of the image at row index i and column index j, $f(i, j)$ is the fingerprint pattern, $s(i, j)$ is the scanner pattern, and $n(i, j, t)$ is the scanner noise, which also depends on the time t because the scanner noise is time varying (by definition). All operations in Expression 5.12 are again element by element (i.e., pixel by pixel) because the point spread function of the Veridicom scanner can also be approximated (with some caveats) by a Dirac delta function. The range of $g(i, j)$ is again from 0 to 255 grayscale levels (8 bits/pixel). And finally, the implicit assumption in Expression 5.12 again is that the pixel values $g(i, j)$, as saved in a computer file, are not further enhanced (or compressed) by image processing algorithms in order to facilitate the biometric authentication, or are enhanced (or compressed) but the scanner pattern information in them is not substantially altered or destroyed.

A figure in the same patent [Veridicom Patent US'620] shows that when the distance d_a between the metal plate and the skin surface, e.g., the depth of a valley, is over 4 μm, C_f goes below 10 fF, thus it vanishes in Expression 5.11 and therefore the term representing the fingerprint pattern in practice disappears. In addition, from the same figure, it is also clear that for $d_a \geq 2$ μm, C_{eq} $(= C_m + C_f)$ effectively flattens out to about 200 $fF \approx C_m$, i.e., the effect of the finger on it can be neglected, and C_{eq} becomes as small as when no finger is present/applied to the scanner platen. Therefore, the pixel value at a sufficiently large/deep valley will be approximately equal to the pixel value of an image that is acquired with no object applied to the scanner platen, i.e., with air:

$$\Delta V \approx \frac{I_c \Delta t}{C_m} \tag{5.13}$$

5.4 Signal Characteristics

Constructing a model for the composite signal of the two imperfections of the fingerprint scanners, the scanner pattern and the scanner noise, as introduced in Section 5.4.3, and the fingerprint pattern as well as characterizing these three signals is critically important for the design of the advanced algorithms and optimizing their parameters. The image shown in Figure 5.2 is an image acquired without any object applied to the scanner, i.e., the fingerprint pattern is not present. The variations in the pixel values across the image (enlarged as image B on the right in Figure 5.2) are due both to the scanner pattern and to the scanner noise, and separating the two when only a single image is available, even in this simple case, is not trivial. We will show that this separation, in general, can be avoided for the purposes of authenticating the scanner. Nevertheless, understanding the characteristics of these two signals individually, as well as those of the fingerprint pattern, and then quantifying them (for the purposes of the algorithm design), as accurately as feasible, is important and is therefore discussed in this section.

We start by revisiting the general observation we made in Section 5.2.4.2. Figure 5.31 shows blocks of pixels from two images acquired with one and the same fingerprint scanner (Scanner u105, one of the FIPS-compliant capacitive area scanners of UPEK). The first image was acquired with air and the second one – with a fingertip. Since there is no fingerprint pattern in the image with air (in the upper plots), the pixel nonuniformity can be due only to the combined effect of the scanner pattern and the scanner noise. On the lower plot, the histogram can be split into two general regions: a region with ridges and a region with valleys. The region with valleys contains the scanner pattern and scanner noise, and is roughly similar to the histogram in the upper plot.

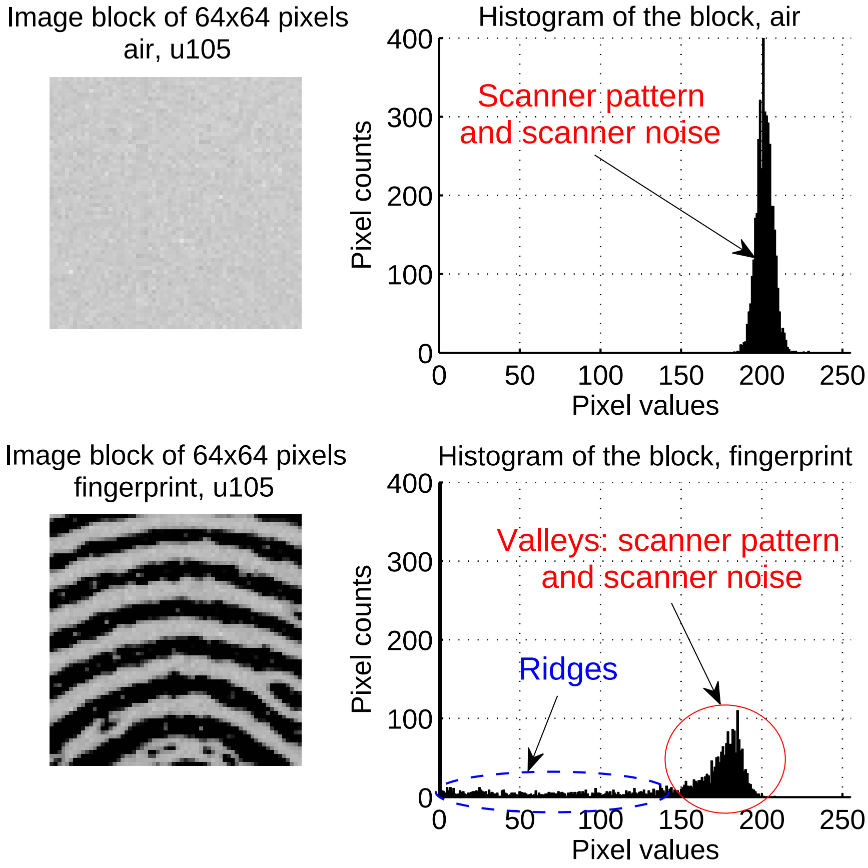


Figure 5.31: Blocks and their histograms from images with air and a fingerprint

Another example for the case when the image contains only the scanner pattern and the scanner noise is shown in Figure 5.32 (the scanner used is Scanner u116). At first look, it seems that the histogram is bimodal: next to the dominating Mode 1, there is another, smaller Mode 2. This suggests that the underlying process might be a mixture of two different distributions or, if the distribution is only one, then its parameters may be changing across the scanner area. Therefore, using a simple

Gaussian model for it will be clearly inaccurate.

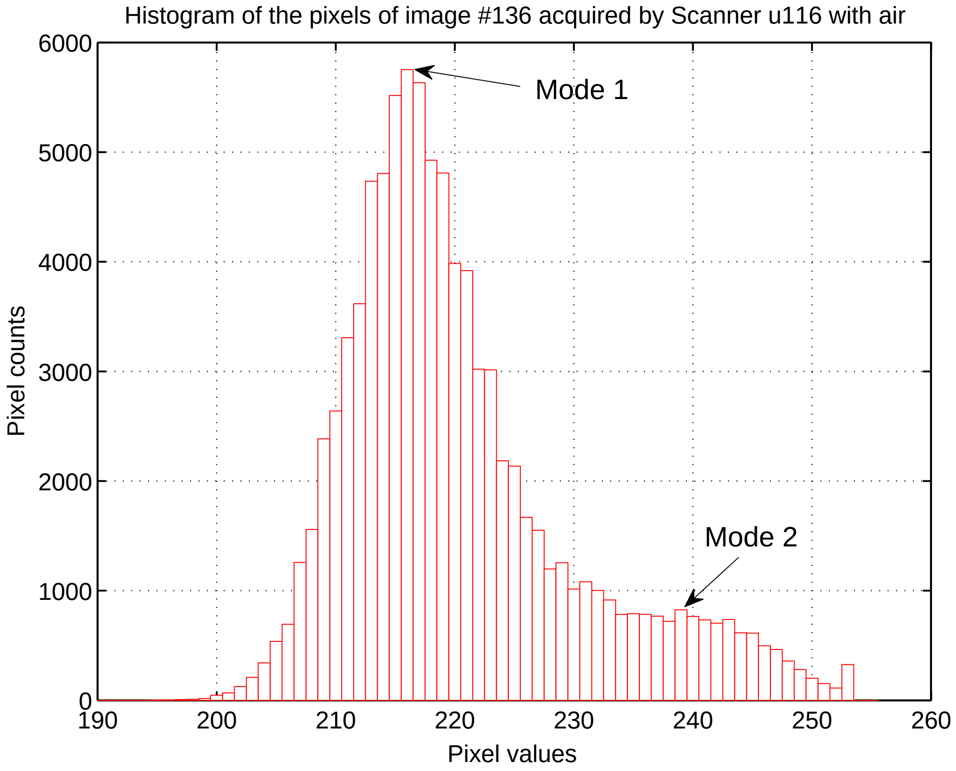


Figure 5.32: Histogram of an image acquired by a UPEK area scanner with air

Nevertheless, our study showed that with some caveats, the scanner pattern and the scanner noise can still be assumed to have Gaussian distributions. Once we established this, the problem of the separation of the scanner pattern and the fingerprint pattern (and the optimality of the proposed solution) became greatly simplified. For verification of the normality assumption, we used the conventional method of applying hypothesis tests for the underlying distribution:

$$H_0 : \text{ Gaussian with unknown parameters} \qquad H_1 : \text{ otherwise.} \tag{5.14}$$

Since H_0 is a composite hypothesis, the parameters of the Gaussian distribution have to be estimated from the data, which makes many hypothesis tests not suited for our problem.

Because of the specifics of the scanner pattern and the scanner noise, our study concluded that the hypothesis tests have to be both carefully selected and carefully applied. For this reason, here below we provide a very brief summary of several hypothesis tests, of their power (the probability to correctly reject the null hypothesis), and some of their limitations. Generally, the hypothesis tests (for normality) can be classified in three main groups/types:

1. χ^2 Hypothesis Test

The Pearson χ^2 goodness-of-fit test is presumably the oldest and the best known hypothesis test [D'Agostino and Stephens 1986]. Its test statistic is the sum of the squared differences between the observed frequencies and the expected (theoretical) frequencies (asserted by the null hypothesis), normalized by the expected frequencies. Under the null hypothesis, the test statistic asymptotically has a χ^2 distribution.

The χ^2 test, however, is (much) weaker than the other tests we discuss below. Furthermore, since it inherently presumes that the sample is drawn from a discrete distribution, applying it to continuous distributions makes the test even less accurate as the data is artificially grouped into bins. This becomes a significant problem when the number of bins is too small as a lot of information is lost due to the grouping. In addition, there is a necessary minimum (i.e., 5) for the expected frequency per bin.

Nevertheless, the χ^2 test is very well studied, can be applied to any null-hypothesis distribution, and can be easily adapted (by decreasing the degrees of freedom of the asymptotic sampling distribution) to test a composite hypothesis (as our case is). Furthermore, it also provides an additional advantage for testing grouped (categorical) data as the centers of the bins can predetermined. This is particularly important for the scanner noise because the quantization is too coarse for it. We use this by setting the centers of the bins to integers, being the grayscale levels (from 0 to 255) of the image pixels, that span the signal range; we call this test *centered* χ^2. Finally, another advantage of the χ^2 test (for us) is its insensitivity to outliers as it inherently groups them in the outer bins.

As a computer program, we used the MATLAB implementation of the χ^2 test (`chi2gof.m`), which, along with the other MATLAB hypothesis tests discussed below, is available in the Statistics Toolbox of MATLAB.

2. EDF-based Hypothesis Tests

The empirical distribution function (EDF) is arguably the main "tool" we used not only in the hypothesis tests, but also for studying other aspects of the signal characteristics. A powerful group of hypothesis tests based on the EDF uses the distance between the EDF and the CDF of the null hypothesis. Depending on how the test statistic is computed from this distance, the tests in this group can be further classified into two main subgroups: (a) the test statistic is a function of the extremities in this difference, such as the Kolmogorov-Smirnov test, and (b) the test statistic is a function of the average (integral), also potentially weighted, of this distance, such as the Cramer-von Mises and the Anderson-Darling tests.

From subgroup (a), the Kolmogorov-Smirnov test, unfortunately, cannot be used for a composite hypothesis. [Lilliefors 1967] extended the Kolmogorov-Smirnov test for normality when the mean and the variance are unknown by using Monte Carlo simulations. In our tests, we used the MATLAB implementation of the Lilliefors test (`lillietest.m`).

For subgroup (b), our study on the topic concluded that the most powerful test in this subgroup and for which we could find an implementation is the Anderson-Darling test ([Anderson and Darling 1952] and [Anderson and Darling 1954]). It is also a good omnibus test and is comparable in power to the Jarque-Bera test (discussed below). The Anderson-Darling test also does not have problems with outliers because it is EDF based.

This test is an extension of the Cramer-von Mises test that further incorporates a weighting function that places more weight on the tails of the distribution, thus making it more sensitive in the tails. It can also be used for a composite hypothesis. We found a free implementation of the test (`AnDartest.m`) from [Trujillo-Ortiz et al. 2007], which we compared and verified with the methodology for the test as described in [D'Agostino and Stephens 1986] before using it.

The main problem of the EDF-based tests is with discrete distributions, especially with signals that are quantized very coarsely. Nevertheless, the Anderson-Darling test is one we used very extensively by applying it where the quantization problem is not severe.

3. Moment-based Hypothesis Tests

The test statistic in this group is a function of one or several statistical moments. The advantage of these tests over the EDF-based tests is that the former do not suffer from the coarse quantization problem; however, as a downside, their accuracy is very low in the presence of outliers. For this reason, we had to remove the outliers before applying them to our signals.

For this group of tests, our study concluded that the most powerful one is the Jarque-Bera test [Jarque and Bera 1987], which is specifically designed for alternatives in the Pearson system of distributions. It is also claimed to be more powerful than the Shapiro-Wilk and the Shapiro-Francia tests (again from the same group).

The Jarque-Bera test statistic is a function of the sum of the squared sample skewness and the squared sample (excess) kurtosis. Under Gaussian null hypothesis, it asymptotically has a χ^2 distribution with two degrees of freedom. A problem of this test is that for small sample sizes, the χ^2 approximation is not sufficiently accurate and the test often incorrectly rejects the null hypothesis. We used the MATLAB implementation of the test (`jbtest.m`), which for small sample sizes (below 2,000, which is our case) uses tables computed with Monte Carlo simulations.

Because of the specifics of the signals of interest for us, to study their characteristics, we used those hypothesis tests which are more likely to give accurate results for the particular signal in question. Furthermore, whenever possible, we also tried to use more than one test – at least one test from each group, as a cross-check.

5.4.1 Scanner Noise

With *scanner noise* we denote the combined effect of all time-varying factors that result in short-term variations, i.e., from within several seconds to much faster, in the pixel values of consecutively acquired images under exactly the same acquisition conditions (e.g., when the fingertip applied to the scanner is not changed in position, the force with which the fingertip is pressed to the scanner platen is kept constant, and the skin moisture is unchanged) and under exactly the same environmental conditions (e.g., without changes in the temperature, air humidity, or air pressure). Examples for factors contributing to the scanner noise are the thermal, shot, flicker, and so on noises that are present in any electronic circuit, and the quantization noise, which is the distortion introduced by the conversion of an analog signal into a digital signal. Other contributing noise sources may also exist, but identifying and quantifying them without details about the specific scanner implementation, which information is proprietary, is very difficult, and we have, therefore, made no effort to do so. A plausible assumption is that the combined effect of all such factors is similar to the combined effect of many noise sources, which is modeled as a (temporal additive) noise in, for example, communication systems. Therefore, of importance for our study are the statistical characteristics only of this aggregation of all short-term temporal noise sources.

Certainly, reproducing the exact same conditions in two consecutive image acquisitions when the object applied to the scanner platen is a fingertip is impossible and remains an abstraction we use only to define the scanner noise. However, it is still possible to ensure that the acquisition and environmental conditions in consecutive image acquisitions are sufficiently similar with another object, a predetermined one, whose application onto the scanner platen can naturally satisfy the similarity requirement. Because of the specificity of the capacitive sensing, the predetermined object that works best for both Signal Model A and Signal Model B (see Expressions 5.9 and 5.12), is air, i.e., no object applied to the scanner platen. It is possible, however, to use other predetermined objects, e.g., water. In case the predetermined object is air, $f(i, j) = 0$, the expressions for both signal models greatly simplify, and the pixel values at row index i and column index j for either signal model become:

$$g^{(po)}(i, j) = s(i, j) + n(i, j, t) \tag{5.15}$$

To study the characteristics of the scanner noise, we acquired 300 images with air with each of the 22 UPEK area scanners (Scanners u101 through u122, without u104 and u109, as well as u151 and u152; see the appendix, Section 6.1 for details), in three groups, each group containing 100 images. The second group was acquired about 1.5 years after the first group, and the third group – within a couple of days after the second group. The images in each group were acquired consecutively, immediately one after each other (with few exceptions in the first group where some images were acquired over a period of a couple of days). The time interval between two consecutive image acquisitions was at least 2 seconds: the time needed for the sensor matrix to acquire the electrical signal and sample it out, then for the image transfer over the USB interface, and finally to be saved as a bitmap file in the computer. However, no attempt was made to keep the time interval between two consecutive image acquisitions constant; thus, only the minimum time period between the consecutive image acquisitions is 2 seconds. All images per scanner in a group were acquired within several minutes, and it is therefore reasonable to assume that the scanner pattern remained constant within each group of 100 images. The scanner pattern of each scanner, however, did change from one group to another, mainly because of temperature, humidity, and air pressure differences on the different days of acquisition. Besides these factors, another major factor, however, may have created even larger variations in the values of certain pixels or even areas of pixels in the acquired images – the impurities on the scanner platen. We put a considerable effort into carefully cleaning the scanner platen when acquiring images with air; however, small particles (including dust) remained, particularly in the areas close to the edges of the platen where cleaning was difficult. The presence of such impurities can make the estimation of the scanner pattern and characterization of the scanner noise considerably inaccurate. However hard we tried to limit the impact of the impurities, they are still present and even noticeable in some of the images.

We also acquired 300 images with air with the Veridicom/Fujitsu scanner (v1, see the appendix, Section 6.1). However, most of these 300 images were acquired within a five-day period (one group per day). Also, the minimum time interval between two consecutive image acquisitions for this scanner was considerably larger – about 10 seconds – due to limitations of the acquisition software and the hardware interface to the computer.

Since it is not possible to determine exactly the time t in Expression 5.15, we instead denote with an index the corresponding pixel value and scanner noise. Thus, if k is the k-th image with air acquired with a particular scanner in this sequence of 300 images, then the pixel values at row index i and column index j in Expression 5.15 are:

$$g_k^{(po)}(i,j) = s(i,j) + n_k(i,j) \tag{5.16}$$

Our initial observation and working assumption was that the scanner noise has a Gaussian distribution. However, the later, detailed study showed that the validity of this assumption is difficult to be verified accurately, and for some of the pixels it may not hold true. Furthermore, the scanner noise, as a time-varying factor that changes the pixel values of the images acquired with air (see Expression 5.15), is presumably mainly a temporal process and thus its temporal characteristics need to be quantified. However, the advanced algorithms estimate the scanner pattern from a single image and therefore they need to take into account primarily the spatial characteristics of the scanner noise, i.e., the characteristics across a single image at a given time instant. Because of the specificity of the acquisition process on hardware level (see Section 5.3), the temporal and the spatial characteristics of the scanner noise are not necessarily the same. Therefore, we studied them separately. Before proceeding, however, we first discuss the impact of the signal quantization on the hypothesis tests because the problem the quantization creates is significantly more pronounced for the scanner noise for than it is the scanner pattern (and it is, therefore, easier to study for the scanner noise).

5.4.2 Quantization and Signal Amplitude Distribution

Unfortunately, the only signal available to our study was the amplitude-quantized (and space-sampled) version of the analog, nonquantized signal acquired by the sensor elements. As already explained, the scale resolution of the fingerprint scanners, which for capacitive scanners is at most 8 bits/pixel, is designed to facilitate the fingerprint recognition but is not sufficiently fine for high-accuracy estimation of the scanner pattern and especially for characterizing the scanner noise. With 8 bits/pixel, the range of the pixel values in an image is from 0 to 255; the step size is 1. The typical scanner noise variance, estimated from such images, is around 2 for Signal Model A and around 1 for Signal Model B. With such small values of the standard deviation (from 1 to 1.4), one can expect that so coarse quantization significantly alters the nonquantized scanner noise. Although the variance of the scanner pattern (as estimated from the image pixels) is much larger than that of the scanner noise (typically about 10 to 15 times larger), this coarse quantization may also affect the much stronger scanner pattern and its accurate estimation.

When quantizing so coarsely a nonquantized signal with a Gaussian distribution, the distribution of the resultant quantized signal, even very approximately, is not necessarily Gaussian. The conventionally assumed probability distribution of the error, introduced by this conversion, between the nonquantized signal and its quantized version is uniform; this error is also modeled as a noise added to the signal being quantized and is called quantization "noise." This model, however, is sufficiently accurate when (a) the signal being quantized is sufficiently random and

(b) the quantization step is sufficiently small (with respect to the signal dynamic range). The quantization of a periodic (or pseudo-periodic) signal results in a periodic distortion (called spurious tones); the quantization with a large step size results in significant changes in the signal amplitude distribution, which is discussed next.

Performing hypothesis tests on coarsely quantized signals may produce inaccurate results, but we observed that by running carefully selected hypothesis tests and taking into account the specifics of their operation, we can draw conclusions about the (possible) characteristics of the nonquantized signal that becomes the scanner noise, as we define and can estimate from the images. For example, it is reasonable to expect that hypothesis tests based on EDF metrics that do not inherently partition the EDF into bins will often reject the null hypothesis for the quantized signal even when the nonquantized signal is Gaussian. Since studying theoretically the effects of quantization on the signal probability distribution was beyond the scope of this thesis and quantifying them precisely was not necessary for developing the advanced algorithms, we studied the process only by simulations. These simulations, one of which we illustrate next, gave us insight on which hypothesis tests are more likely to be less affected by the quantization and how to use them in order to study the scanner noise by mitigating the quantization artifacts.

We generated toy signals with Gaussian distribution with the (built-in) MATLAB function `randn()`, which generates a sequence of normally distributed numbers using a pseudorandom generator. The toy nonquantized signal is one such sequence of 100 numbers with Gaussian distribution with zero mean and variance equal to 2 (which is representative for Signal Model A); in MATLAB code this is:

`randn(100,1)*sqrt(2)`. For quantization, we assumed truncation (the MATLAB function `floor()`) as this is the most typical quantization used in analog-to-digital converters (we don't have information about the converter used in the UPEK or Veridicom scanners). The time sequences of one such simulation are shown in Figure 5.33.

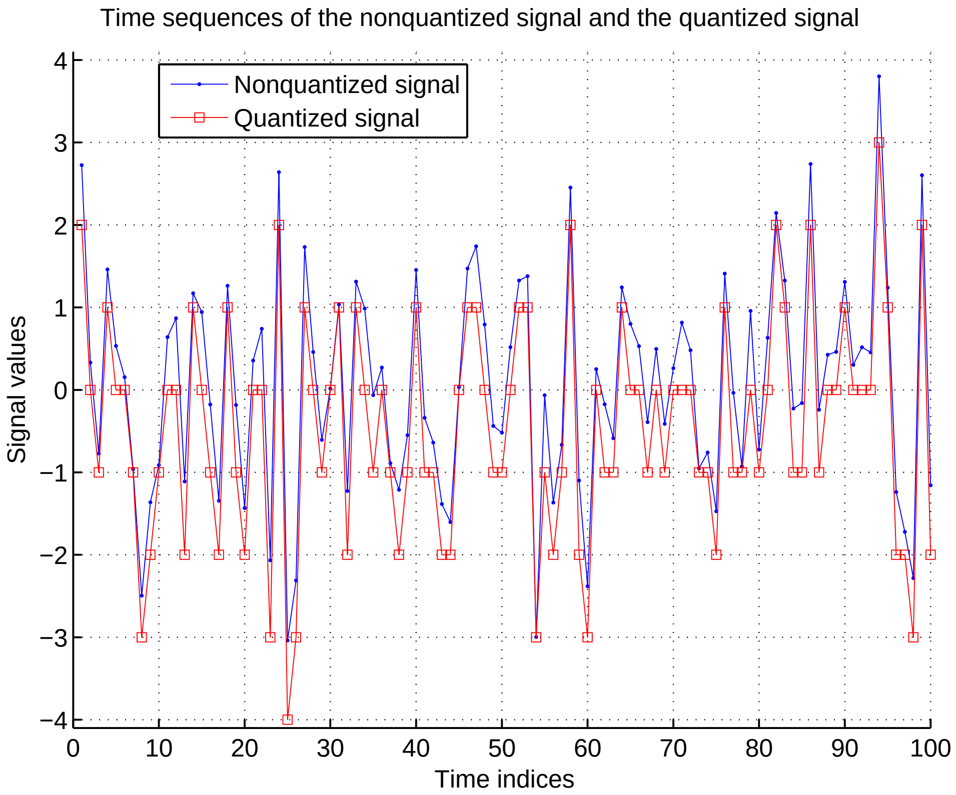


Figure 5.33: Quantization of a toy signal: time sequences

Figure 5.34 shows the EDF of the nonquantized signal, together with the fitted to it Gaussian CDF (with mean 0.0632 and variance 1.7475, the MLE estimates from the nonquantized signal) and the EDF of the quantized signal, together with the fitted to it Gaussian CDF (with mean -0.46 and variance 1.8267, the MLE estimates from the quantized signal); the true Gaussian CDF with mean 0 and variance 2 is also shown.

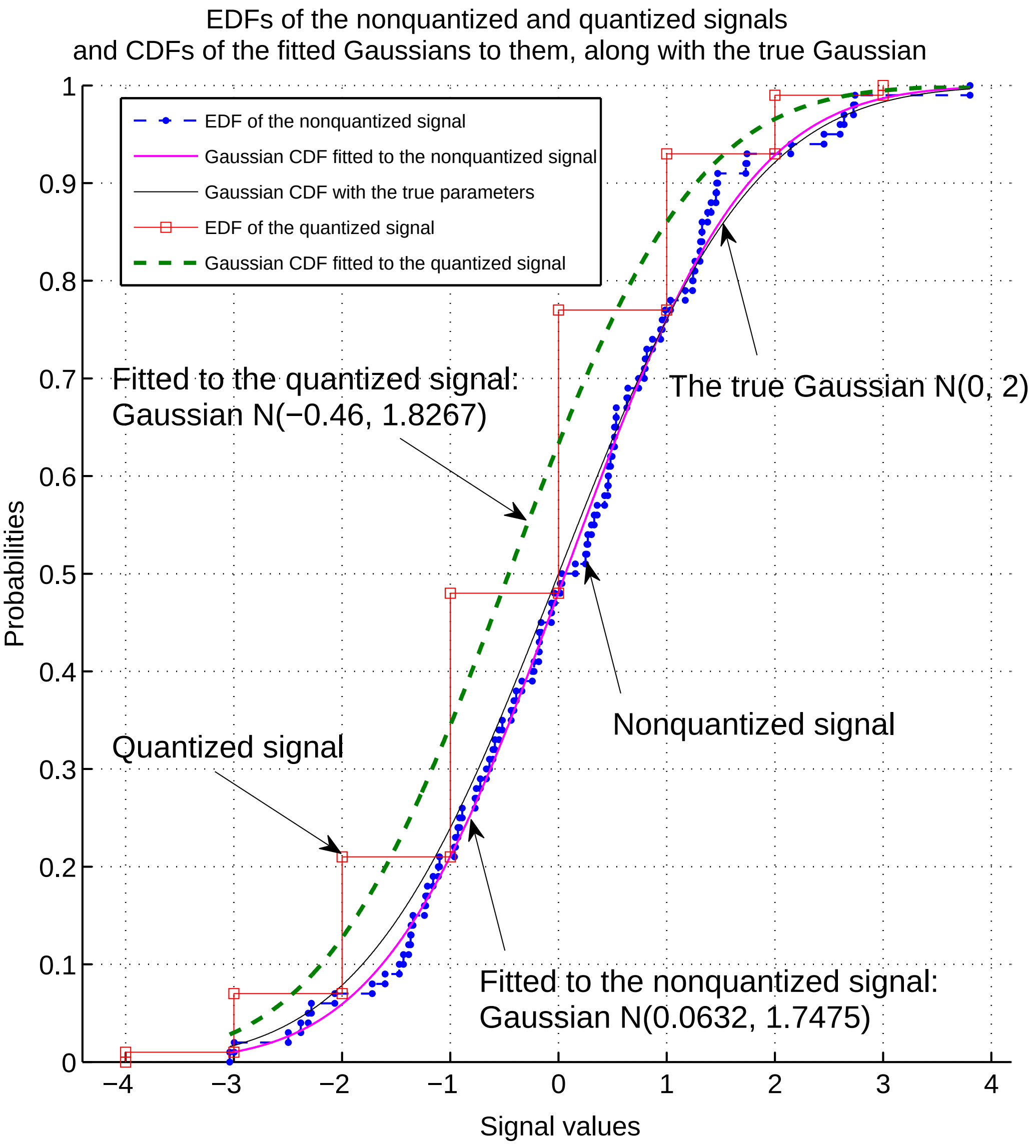


Figure 5.34: Quantization of a toy signal: EDFs and fitted CDFs

It is clear that the EDF of the nonquantized signal is extremely close to the true Gaussian CDF. In fact, even the two-sided Kolmogorov-Smirnov test (with mean and variance of the true Gaussian CDF) fails to reject the null hypothesis at

0.05 significance level; actually, the p-value of the test is quite large (0.8393) and thus there is extremely little evidence to reject the null hypothesis. In contrast, the same test for the EDF of the quantized signal gives a p-value of 0.01188, thus rejecting the null hypothesis at 0.05 significance level and narrowly failing to reject the null hypothesis at 0.01 significance level. (The latter test was run for a null hypothesis with mean -0.5 as to compensate for the offset in the mean introduced by the truncation, which is $\frac{1}{2}$ of the step size). Running the Kolmogorov-Smirnov test on other toy signals produced p-values well below 0.01, but we decided to provide this example as an evidence that the Kolmogorov-Smirnov test may sometimes fail to reject the null hypothesis at a significance level that is not sufficiently small. Nevertheless, our conclusion is that it also reasonable to believe that any EDF-based hypothesis test from both the Kolmogorov-Smirnov and the Cramer-von Mises group will very likely reject the null hypothesis for the quantized signal (the results of other tests are shown later).

Figure 5.35 shows the same distributions, but on a normal probability plot, where is easier to judge by eye as the EDFs are compared with straight lines. This figure gave us the idea that a hypothesis test that groups the signal values in bins centered at the integer numbers may not suffer from the quantization problem; a test that can be easily adapted in this way is the χ^2 test.

We also generated another set of toy signals with a smaller variance, equal to 1, which is intended to represent Signal Model B (although the scanner noise there has peculiar behavior). Here the quantization problem is much more severe as the quantized signal effectively occupies only 5 levels: -2, -1, 0, 1, and 2.

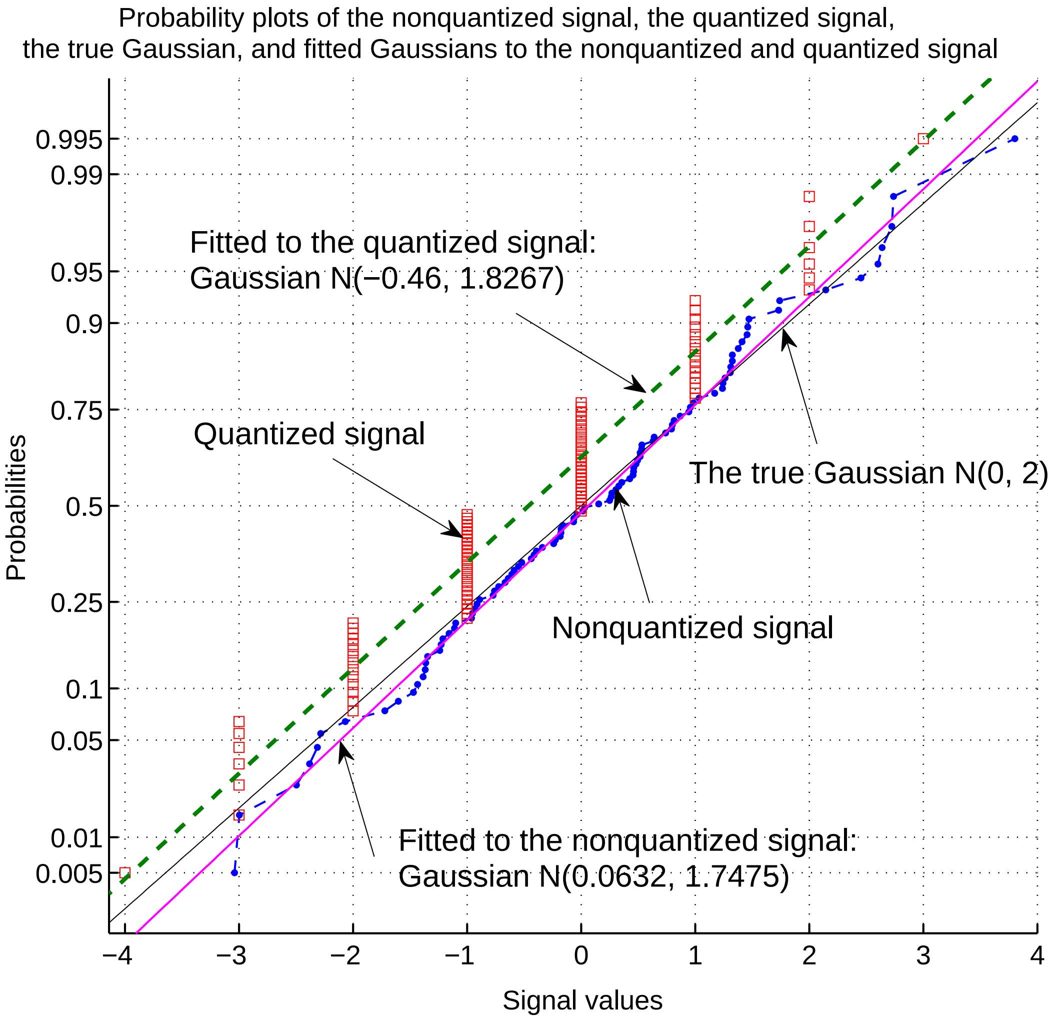


Figure 5.35: Quantization of a toy signal: normal probability plots

To study the applicability of the hypothesis tests for testing the quantized signal, we ran several of the tests on the already discussed nonquantized signal with variance 2 and on its quantized version, and on another such pair of signals from the second set, with variance 1. The p-values of the tests are shown in Table 5.9. Each test estimates the parameters of the Gaussian distribution from the signal. All tests fail to reject the null hypothesis for the nonquantized signal even at 0.5 significance level (when the p-value is beyond the largest tabulated value, the MATLAB implementations of the Jarque-Bera and Lilliefors tests return p-values of 0.5).

Hypothesis test	**Variance=1 Nonquantized**	**Variance=1 Quantized**	**Variance=2 Nonquantized**	**Variance=2 Quantized**
χ^2	0.8477	4.935e-20	0.6201	1.024e-08
χ^2 **centered**	N/A	**0.6005**	N/A	**0.9563**
Jarque-Bera	> 0.5	$> \mathbf{0.5}$	> 0.5	$> \mathbf{0.5}$
Lilliefors	> 0.5	< 0.001	> 0.5	< 0.001
Anderson-Darling	0.7732	1.414e-10	0.8313	9.584e-06

Table 5.9: Quantization of toy signals: p-values of several hypothesis tests

As expected, the EDF-based Lilliefors and Anderson-Darling tests reject the null hypothesis for both quantized signals even at 0.001 significance level (the MATLAB implementation of the Lilliefors test returns a p-value of 0.001 when the p-value is smaller than the smallest tabulated value). The regular χ^2 test also rejects the null hypothesis for both quantized signals with overwhelming evidence; however, the centered χ^2 test fails to reject the null hypothesis with p-values well above 0.5 for both quantized signals. Furthermore, the moment-based Jarque-Bera test also fails to reject the null hypothesis for both quantized signals with p-values above 0.5, thus making it also a good candidate for hypothesis testing of the actual signals, the scanner noise and scanner pattern. Therefore, for signal characterization, we focused on using the centered χ^2 and the Jarque-Bera tests, while running the other tests for comparison.

Note: the documentation of the function `chi2gof.m` describes that "[the] bins in either tail with an expected count less than 5 are pooled with neighboring bins until the count in each extreme bin is at least 5." Thus, it is possible that there is no bin at each and every integer number in the tails if testing a quantized signal when calling the function with an option that sets the centers of the bins (case "χ^2 centered" in Table 5.9 above).

A possible question that may arise is why we focus on hypothesis tests only for normality. The reason is this: the study of the fingerprint scanner acquisition hardware along with the conventionally known characteristics of variability and noise in electronic devices strongly suggested that our signals of interest in the analog domain have Gaussian distributions, making the Gaussian distribution a strong candidate for the null hypothesis. The coarse quantization, however, may distort these signals upon conversion so that the signals that we measure and can estimate from the digital images may appear as produced by non-Gaussian sources. In order to avoid possible artifacts of the hypothesis tests when used with so coarse quantization, we examined the operation of the tests and selected those of them that are likely to produce the most truthful results. If the Gaussian assumption, barring the quantization problem, holds true for a significantly large portion of the signals, we can use the conventional theory of signal processing to design simple, optimal, and robust algorithms. Assuming and modeling non-Gaussian signals only because of the coarse quantization problem is not only much less tractable and more difficult, but it may have also led to unnecessarily complex and potentially non-robust algorithms.

5.4.2.1 Scanner Noise: Temporal Characteristics

Before quantifying the temporal characteristics of the scanner noise, we present one example specifically chosen to illustrate the noise process in time. As already explained, it was very difficult to ensure image acquisitions at precise, regular time intervals, so we put no effort to achieve this. Therefore, the image index in a group cannot be considered as a time stamp of the acquisition event, but only as a sequence index. The pixel values $g_k^{(po)}(143, 108)$ (see Expression 5.16) of the pixel at row 143 and column 108 of Scanner u106 for the sequence of images are shown in Figure 5.36. The three groups of 100 images, each acquired with air, are:

- **Group 1** – images 1 through 100: acquired consecutively within about 5 minutes; thus, the time between each two consecutive images is about 3 seconds;
- **Group 2** – images 101 through 200: acquired consecutively within about 30 minutes. This group of images was acquired about 1 year and 8 months after those in Group 1;
- **Group 3** – images 201 through 300: acquired consecutively within about 4 minutes. This group of images was acquired 3 days after those in Group 2.

Since the scanner noise is zero mean (by assumption) and that the scanner pattern does not change within short periods of time (by definition), the pixel value variations within each group represent the scanner noise. Therefore, by subtracting the sample mean (of each group) from the pixel values within that group, we could have presented (in Figure 5.36) only the scanner noise. We, however, decided to

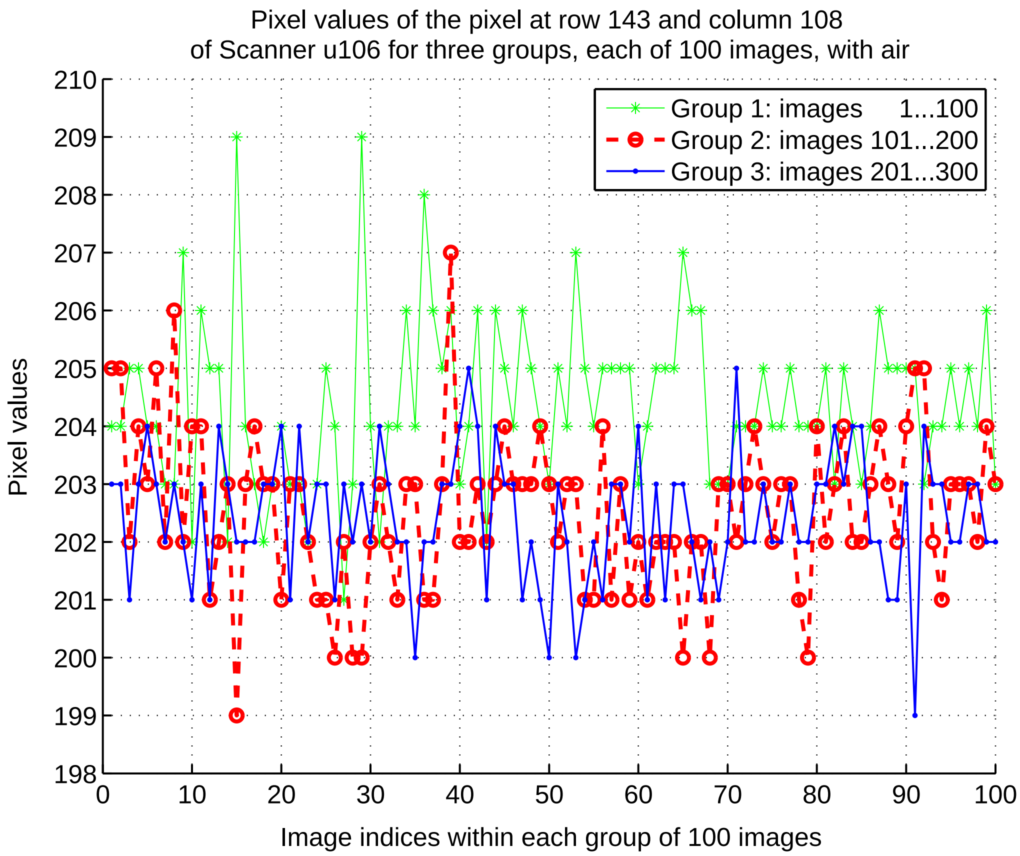


Figure 5.36: Temporal values of pixel (143,108) of Scanner u106 for the 3 groups

present the pixel values without subtracting the sample mean and then study the signal characteristics by analyzing the pixel values as they are in the images because in this way we do not introduce artifacts that may possibly impact the hypothesis tests. The temporal characteristics, computed separately for each group of images, of are shown in Table 5.10.

The first observation in Table 5.10 is that the sample mean of Group 1 is very different from the sample means of Group 2 and Group 3 – the difference is about 2 grayscale levels and thus about 14 times larger than the standard error of the sample mean; this standard error is *sqrt(sample variance/sample size)* $\approx \sqrt{2/100} \approx$

Signal characteristics	**Group 1 images 1. . . 100**	**Group 2 images 101. . . 200**	**Group 3 images 201. . . 300**
Sample mean	204.39	202.54	202.41
Sample variance	2.018	1.907	1.254
Hypothesis tests	**p-values**	**p-values**	**p-values**
χ^2 centered	0.4702	0.4859	0.4062
Jarque-Bera	0.0095	0.2907	> 0.5
Lilliefors	< 0.001	< 0.001	< 0.001
Anderson-Darling	9.691e-07	2.710e-06	1.529e-08

Table 5.10: Temporal characteristics of the values of pixel (143,108) of Scanner u106

0.141 (the sample variance is also shown in Table 5.10). This large difference can be attributed to one reason: a variation of the scanner pattern (at this particular pixel), which variation was probably caused by substantially different environmental conditions in the different days (about 1 year and 8 months apart from each other). This observation also led to the conclusion that care must be taken when combining quantities from different groups and images. On the other hand, the difference between the sample means of Group 2 and Group 3 is 0.13 and thus is within the standard error of the sample mean. This fact appears to come logically as Group 3 was acquired only 3 days after Group 2, and it is reasonable to believe that the environmental conditions did not change significantly in the meantime.

The second observation from Table 5.10 is that the sample variances (and thus the power of the scanner noise) of Group 1 and Group 2 are considerably different from the sample variance of Group 3, despite the fact that the sample means of Group 2 and Group 3 are very close. One possible explanation is that the temperature, air humidity, or air pressure difference between these two acquisition days was sufficiently large to change the scanner noise, yet sufficiently small to not have much impact on the scanner pattern. For example, a temperature drop of 1 degree Celcius (between Group 2 and Group 3) might have been sufficient to decrease the scanner noise variance (and power) from 1.907 to 1.254, yet insufficient to change significantly the scanner pattern, decreasing it only slightly from 202.54 to 202.41. This is consistent with our observation (discussed later) that a temperature drop decreases the scanner pattern; it is also well known that a temperature drop typically decreases the noise in electronic circuits. Another explanation for this, however, may be that this difference in the sample means was due solely to the standard error, in which case a possible temperature drop very little affected the scanner pattern.

The EDF-based Lilliefors and Anderson-Darling tests (see Table 5.10) both overwhelmingly reject the null hypothesis for this signal, which essentially is the (quantized) scanner noise. This is expected and the reason for it was already explained; therefore, we will not use anymore these two tests for scanner noise characterization. The p-values of the centered χ^2 test for all three groups are very large (over 0.4) and fail to reject the null hypothesis, implying that the (quantized) scanner noise for this particular pixel most likely has a Gaussian distribution; the p-values are also relatively close to each other. The moment-based Jarque-Bera test also fails to reject the null hypothesis for Group 2 and Group 3, corroborating the Gaussian assumption for the scanner noise. The Jarque-Bera test for Group 1, however, narrowly rejects the null hypothesis even at 0.01 significance level.

The EDFs, the CDFs of the fitted Gaussians, and the normal probability plots of the pixel values of the three groups of images are shown in Figure 5.37. One can notice the remarkable difference between the EDFs (the upper plot) of Group 1 on one side and of Group 2 and Group 3 on the other. We also made an experiment by removing the two largest values of the signal of Group 1, shown in the lower plot, "Trimmed values" (this problem is discussed later). The fitted Gaussian to the trimmed in this way signal shows a better fit to the EDF. Furthermore, running the Jarque-Bera test on the trimmed signal results in failure to reject the null hypothesis with a p-value reported to be larger than 0.5, concurring with the Jarque-Bera test results for Group 2 and Group 3 and therefore suggesting that the signal, except for the trimmed values, may be truly Gaussian and that these two pixel values of 209 (out of 300 pixels in all three groups) are outliers. Indeed, the odds of occurrence of an observation with value as extreme as this one (209) under a Gaussian assumption are 1 in $1/Q((209 - 204.39)/\sqrt{2.018}) \approx 1704$ observations (the sample mean and variance are taken from Table 5.10, Group 1). In contrast, the second extreme pixel value of 209 occurred (within the same Group 1!) mere 14 observations after the first extreme (see Figure 5.36); furthermore, similar extreme values are not present in Group 2 and Group 3. All this corroborates the hypothesis that the cause for these extreme values may be a very different, bursty noise component that is present in the aggregate scanner noise.

Certainly, the presence of such a bursty noise in the scanner noise is not something that can be overlooked and thus be assumed that the scanner noise in time has a Gaussian distribution as it may lead to instability of the algorithms for scanner

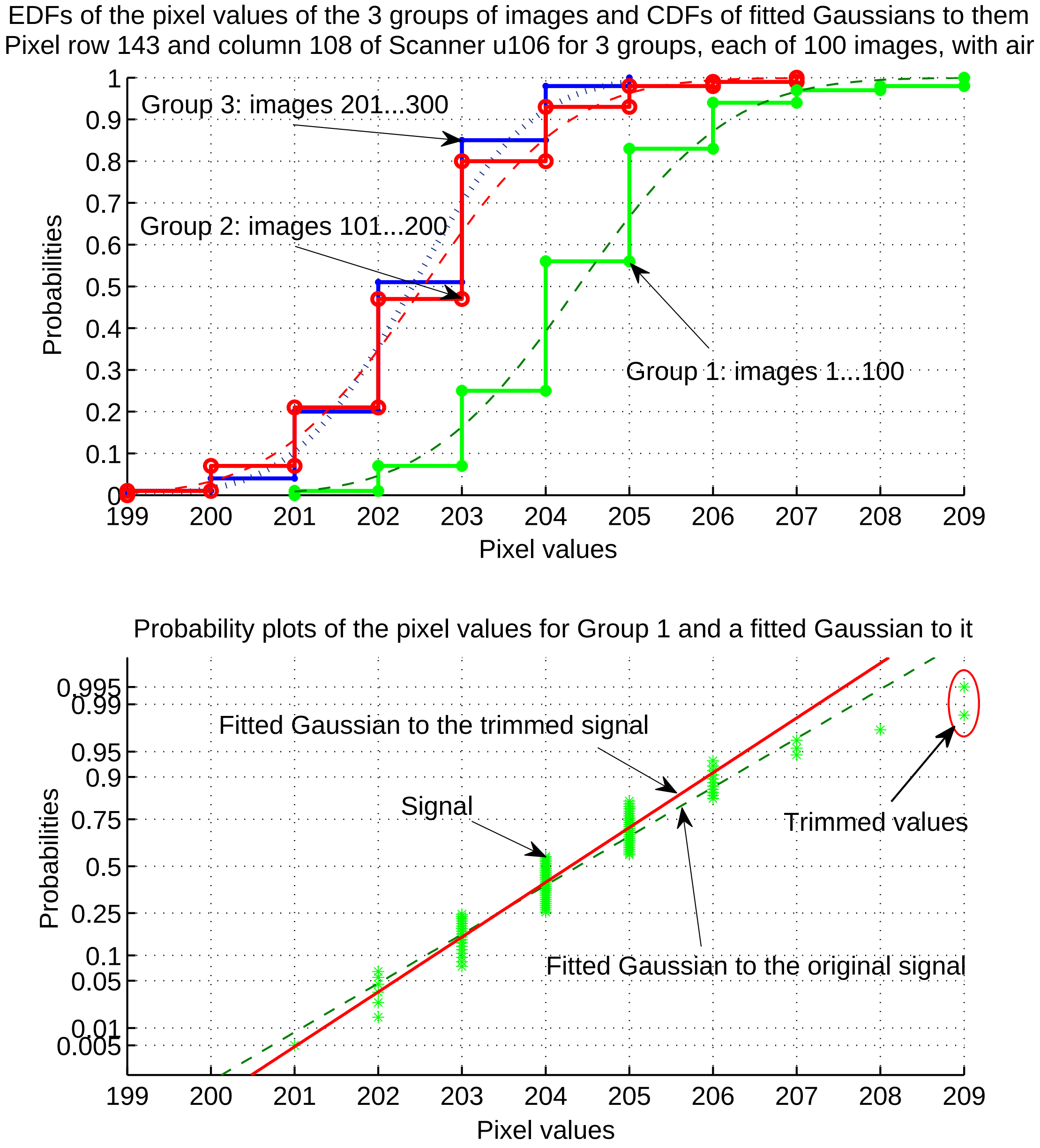


Figure 5.37: EDFs and probability plots of the temporal values of pixel (143,108)

pattern estimation (because the scanner pattern and scanner noise essentially cannot be separated if a single image is used, as it will become clear later). Nevertheless, the advanced algorithms are sufficiently robust to accommodate this bursty noise.

Since we wanted to quantify as accurately as possible the temporal characteristics of the scanner noise, excluding this bursty component, and thus verify the validity of the general Gaussian assumption, we used a rudimentary method to detect and exclude such outliers from the hypothesis tests because, as already shown, outliers can severely affect the hypothesis tests, in particular the Jarque-Bera test which is moment based. For this detection, before applying the hypothesis test, we exclude ("trim") the pixels with values farther than $\pm 3\bar{s}$ away from the sample mean, where $\bar{s}$ is the sample standard deviation. We believe that this trimming leads to negligible impact on the test accuracy for a truly Gaussian noise as only about 0.27% of such noise observations will be excluded from the test.

In addition, besides the outliers, there exist also pixels that are defective (called "dead pixels") and do not operate properly (probably due to hardware malfunction), producing a constant value (typically 253), even when the applied object is a fingertip at different positions. Obviously, such pixels have to also be excluded from the hypothesis tests. It is important to note, however, that these dead pixels have peculiar behavior as once in a while their value may change; this erratic behavior is difficult to be detected automatically and consequently excluded from the hypothesis tests with the routines we wrote and thus may have produced inaccuracy in the hypothesis tests.

The EDFs of the p-values of the centered χ^2 and the Jarque-Bera tests with no trimming and with the 3σ trimming are shown in Figure 5.38 (the trimming of dead pixels is always active). Each p-value in the figure is the result of a hypothesis test over the sample of 100 pixel values $g_k^{(po)}(i, j)$ (see Expression 5.16) of the images in one of the three groups, each of 100 images. Each point on the EDF curve represents the percentage of pixels whose p-values are smaller than the p-value of this point. For example, at p-value of 0.05, the EDF curve of the centered χ^2 test with no trimming has value 0.483, which means that 48.3% of the pixels have the null hypothesis rejected by the centered χ^2 test (when no trimming is done) at 0.05 significance level, i.e., the p-value on the abscissa, because their p-values are smaller than 0.05. The EDF curves shown in the figure are those which have the largest EDF value (at a given p-value) among the three groups of images (i.e., the highest of the three curves), thus providing the worst-case percentage of pixels for which the null hypothesis gets rejected at the corresponding level of significance. The total number of pixels of an image (for the UPEK area scanners used) is $(360 * 256) = 92,160$.

Figure 5.38 shows only the p-values from 0 to 0.1 as only these levels of significance are of potential interest for us and particularly the significance levels of 0.01 and 0.05 (the dash-dotted grid lines). For either one of these two significance levels, both tests with no trimming reject the null hypothesis for a substantial percentage of the pixels, with the Jarque-Bera test rejecting well above 50% of the pixels. This result is not surprising because the moment-based Jarque-Bera test is sensitive to outliers. Trimming the outliers at 3σ significantly reduces the percentage of rejected pixels to about 10% to 12% at 0.01 significance level and to about 25% at 0.05 significance level. It is also remarkable that with this trimming, the EDF curves of the two hypothesis tests, albeit of very different types, are very close to each other. In contrast with the case when no trimming is done, the two EDF curves are far apart. To study the effect of the trimming level, we increased it from 3σ to 4σ, in

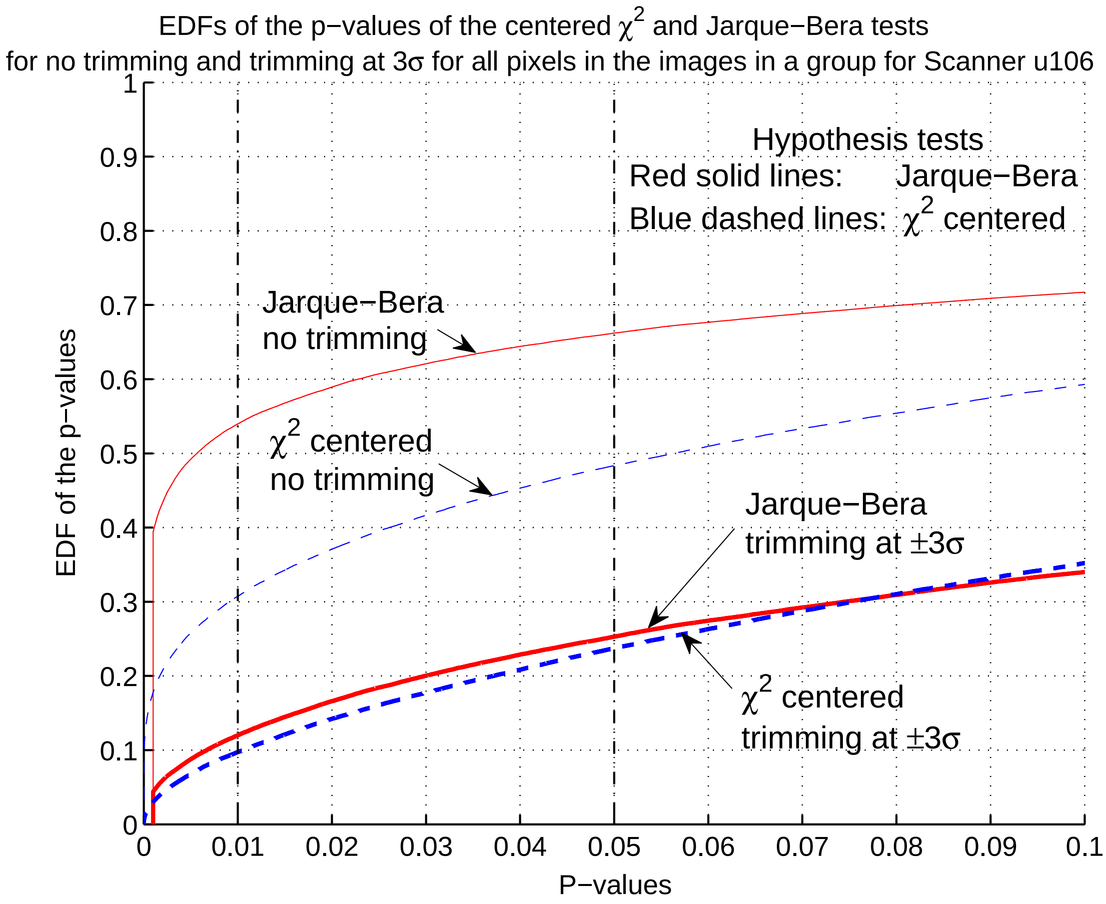


Figure 5.38: EDFs of the p-values of the centered χ^2 and Jarque-Bera tests for the temporal scanner noise, Scanner u106

which case the EDF curves were essentially the same as for the case of no trimming, indirectly indicating that most outliers are from 3σ to 4σ away (roughly) from the mean.

Since the Jarque-Bera test appears to be more conservative than the centered χ^2 test (see Figure 5.38), to generalize these observations for all scanners, we ran the Jarque-Bera test on all 3 groups of images for all scanners. We recorded the percentage of pixels for which the null hypothesis is rejected at 0.01 and 0.05 levels of significance and then took the maximum percentage (as the worst case) among the percentages for the three groups, at each significance level. The results for the UPEK scanners, provided in the appendix, Table 6.7, show that the scanner noise distribution in time cannot be assumed to be Gaussian at 0.01 significance level for about 7.5% to 13% of the pixels and for about 18% to 28% of the pixels at 0.05 significance level, the more conservative significance level.

For the Veridicom scanner, however, the results are considerably different. Here the scanner noise (in time) varies so little relatively to the quantization step that for most pixels, their values occupy only 2 or 3 levels. With so high a level of "quantization noise," it is expected that the Gaussian hypothesis for the nonquantized scanner noise (in time) cannot be verified truthfully as the number of levels are insufficient for accurate hypothesis testing. For example, the χ^2 composite test requires at least 4 bins as then it has the lowest degree of freedom, which is 1 (the degree of the asymptotic distribution is *the number of bins–1–the number of parameters to be estimated from the data*); thus, the χ^2 test essentially cannot be used here. The problem is compounded by the presence of outliers with very high amplitudes, which we also observed here. Nevertheless, we ran the Jarque-Bera test on Scanner v1 and computed that for about 26% of the pixels at 0.01 significance level and for about 51% of the pixels at 0.05 significance level, the scanner noise cannot be assumed to have a Gaussian distribution, which are roughly twice the percentages as for the UPEK scanners. The details are also provided in the appendix, Table 6.7.

The presence of outliers in the scanner noise of the UPEK scanners is considerable, especially close to the edges and in the bottom about 100 rows of the images.

In Figure 5.39, the upper plot shows the pixel values of the pixel at row 355 and column 1 of Scanner u106 for the 100 images in Group 1; the lower plot shows the EDF of the pixel values. The extreme value of 240 is clearly an outlier. To illustrate how much different it is from the rest of the signal, we computed several statistics. First, the sample mean is 216.59, very close to the median 216, which implies that the effect of the outlier (in this group of 100 pixels) on the sample mean is limited. The sample variance, however, is significant – 6.85. Computing the sample mean and the sample variance without the outlier gives 216.35 and 1.2717, respectively, and even though the sample mean in the two cases is similar, the sample variance with the outlier is over 5 times larger than without it. Running the centered χ^2 and the Jarque-Bera tests on the signal with the outlier, without surprise, rejects the null hypothesis (the p-value of the centered χ^2 test is 3.31e-18). When the outlier is removed, the p-value of the centered χ^2 test increases to 0.0211 and that of the Jarque-Bera – to 0.0201, thus very close to that of the centered χ^2 test; both tests reject the null hypothesis at 0.05 significance level, but fail to reject it at 0.01 significance level.

The existence of outliers in the scanner noise is important and has to be considered in the algorithm design, but our advanced algorithms are robust against them. Furthermore, we neither need nor use information about the likelihood of appearance of outliers in different areas of the image (and the scanner). Nevertheless, to illustrate how different the magnitude of the outliers can be in different parts of the image, we developed a simple metric for comparison. For the values of a pixel in the group of 100 images, the ratio between the maximum absolute deviation (in

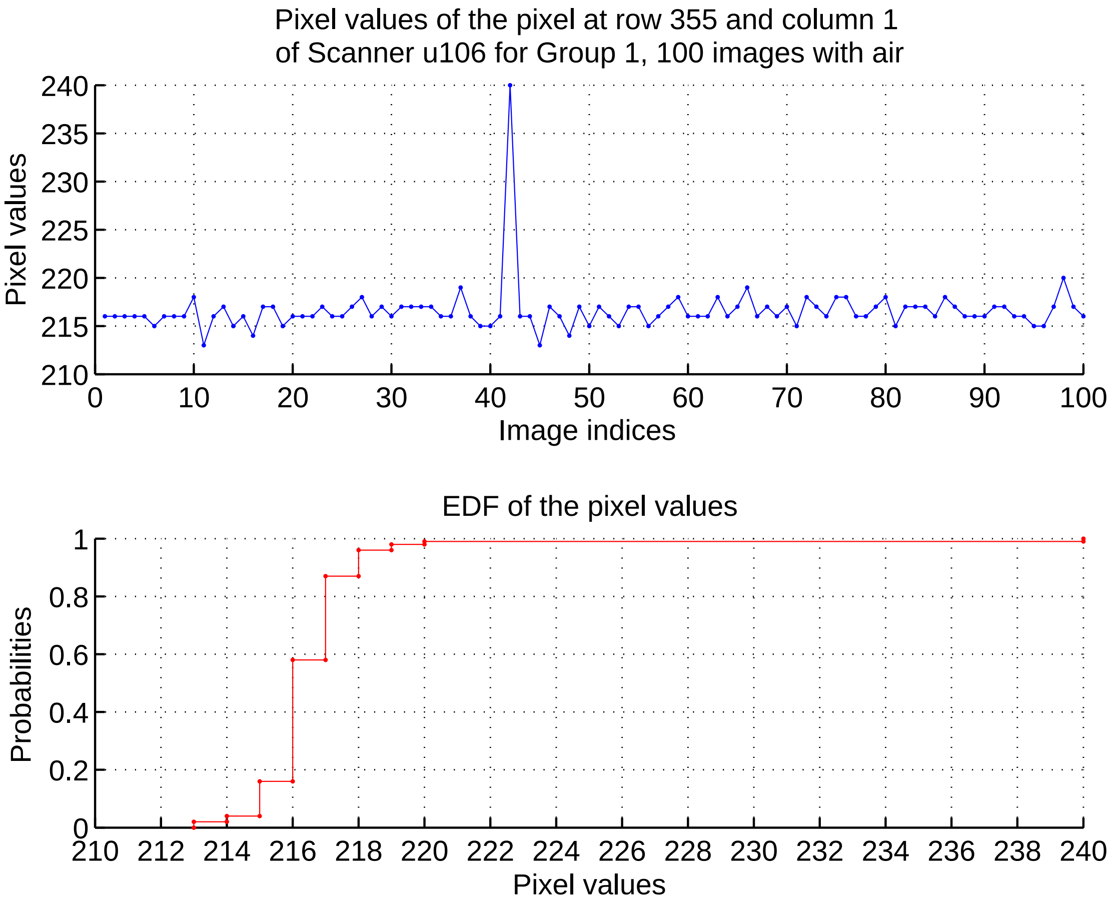


Figure 5.39: Temporal values and their EDF of the pixel (355,1), located on the platen edge of Scanner u106, for 100 images

either direction, positive or negative) from the median and the median provides a relatively accurate basis for comparison. For example, this ratio for the signal, shown in Figure 5.39, with the outlier present is 0.1111, while after removing it, the ratio drops to 0.0185, which is 6 times smaller. These ratios for all pixels of three columns of the same Scanner u106 are shown in Figure 5.40. The three columns are:

- Column 1 (at the image edge): the ratios are predominantly well above 0.02. The mean of the ratios is 0.0302;
- Column 10 (close to the image edge): the ratios are noticeably smaller than those of Column 1. Their mean decreases 0.0251;
- Column 100 (about in the middle of the image): with the exception of the last 100 rows, almost all ratios are 0.015 and smaller. The mean of the ratios further decreases to 0.0151, which is half of the mean of Column 1 ratios.

Estimation of the Variance of the Scanner Noise in Time

In addition to its amplitude distribution, the other most important characteristic of the scanner noise is its variance. Since the scanner pattern does not change in short periods of time (by definition), the sample variance of the values of a pixel across all images, acquired with air, in a group will provide the sample variance of the scanner noise (see Expression 5.16). From the preceding discussion, it is clear that the outliers play a significant role in the sample variance and may increase it considerably if not removed before computing it. On the other hand, the outliers may have different probability of occurrence in the different areas of the scanners (and images); therefore, it is very difficult, if even possible, to remove these outliers from the composite signal while extracting the scanner pattern from it. For this reason, completely neglecting the impact that the outliers have on the signal will be incorrect. Since the scanner noise variance, although indirectly, is an input parameter to the advanced algorithms, we computed an estimate of it. We

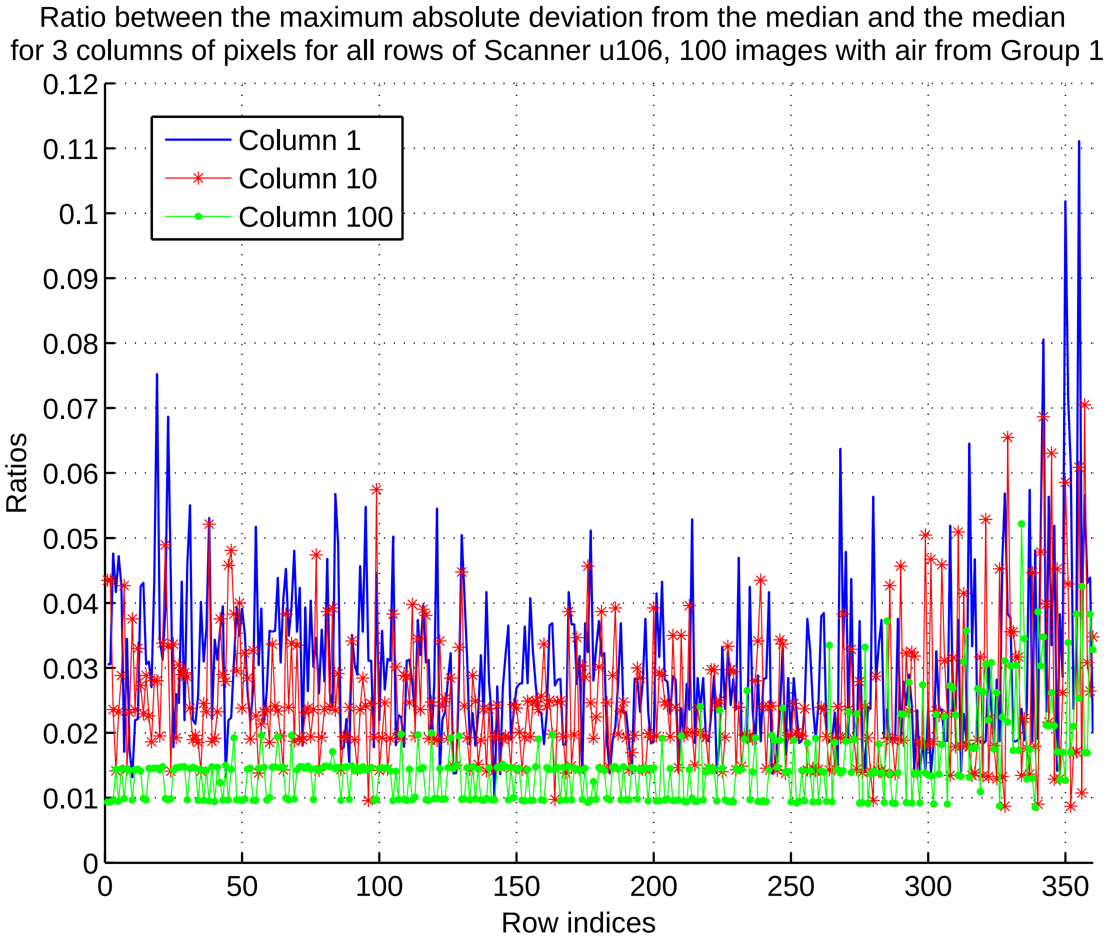


Figure 5.40: Ratios between the maximum deviation and the median for 3 columns of Scanner u106

have to emphasize, however, that the advanced algorithms do not use the scanner noise variance separately and, therefore, it is not required that it is estimated very accurately. Furthermore, as discussed further, the scanner noise variance depends also on the particular scanner; we also observed that the scanner noise variance also depends on the temperature and other environmental factors.

In the process of looking for a sufficiently accurate estimate, we observed that only a small percentage of the pixels (of a given scanner) produce noise observations with large-valued outliers. Figure 5.41 shows the EDFs (for one scanner of UPEK and one scanner of Veridicom) of the sample variances of all pixels of a scanner, computed within one group of 100 images (for the other two groups, the EDFs are essentially the same). For Scanner u106 (UPEK), although some of the pixels have variance well above 10, the variance of 71% of the pixels is below 2 and of 92% of the pixels – below 3, which appears to be the "knee" point of the curve, suggesting that the outliers are pronounced only in the remaining about 8% of the pixels. Thus, not surprisingly, when averaging the sample variances of all pixels for this particular group, the average sample variance is only 1.8724. The EDFs for three other scanners of UPEK (u113, u122, and u151) are practically the same (not shown in the figure) and differ slightly only in the numbers: the percentage of pixels with sample variances below 2 is in the range from 72% to 75%, and below 3 – in the range from 91% to 93%. The EDF for Scanner v1 (Veridicom) is similar, with about 81% of the pixels having variance below 1 and about 91% – variance below 2.5; the average sample variance across all pixels of the scanner is 0.8903.

Considering all discussed above, as a sufficiently good estimate that takes into account the stated limitations, we decided that the conventional sample variance will be accurate enough. Therefore, we first computed the sample variance for each pixel of a given scanner separately for each of the 3 groups. Then we averaged the variances for all pixels per scanner and per group; these three variances per scanner and per group were next averaged to compute the average variance per scanner. Finally, all average variances per scanner were averaged across all scanners and this is the average scanner noise variance in time that we assume for each signal model.

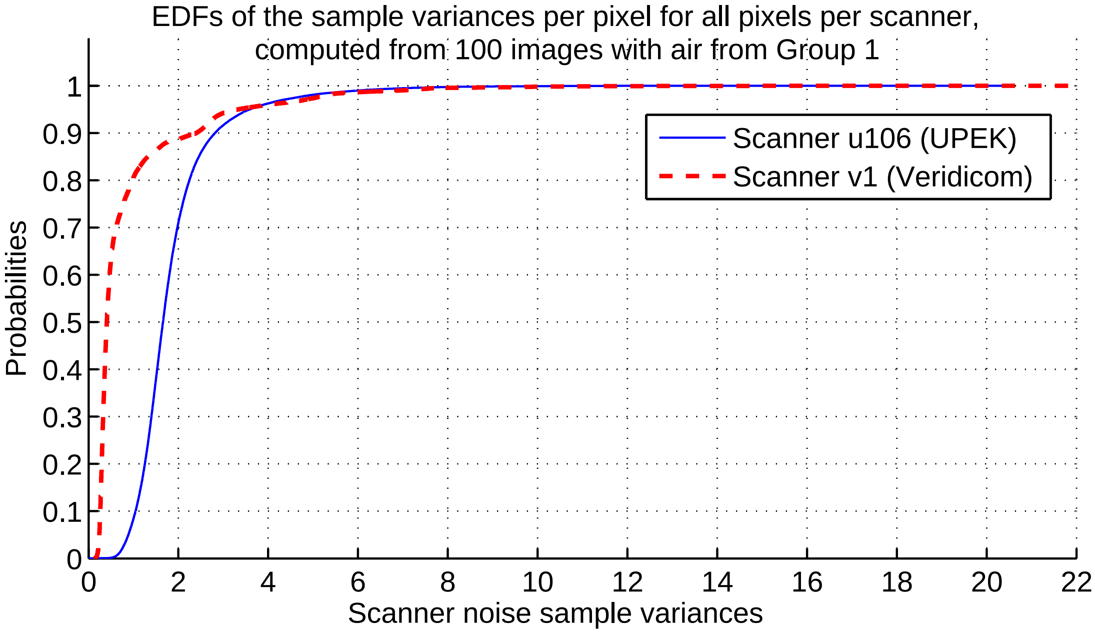


Figure 5.41: EDFs of the sample variances of the temporal scanner noise for 2 scanners

The detailed results are provided in Table 6.6 in the appendix, and in summary:

- Signal Model A (UPEK scanners): the sample variance per scanner ranges from 1.392 to 2.035, with average variance across all scanners **1.785**;

- Signal Model B (Veridicom scanner): the sample variance for the scanner we have is **0.885**.

Finally, to characterize the scanner noise in time, besides the probability distribution, we also need to study, at least, how much correlated in time the observations are. This, unfortunately, is not possible with our setup (hardware and software) and the currently acquired images because the time interval between two consecutive image acquisitions varies and cannot be controlled sufficiently precisely in order to render an accurate estimation of the temporal autocorrelation function of the scanner noise. The reason we did not put any further effort in quantifying the temporal correlation of the scanner noise is because it is not important – since the advanced algorithms use a single image to estimate the scanner pattern, of greater importance is not the temporal, but the spatial correlation, and generally the spatial characteristics, of the scanner noise, which is discussed in the next section.

5.4.2.2 Scanner Noise: Spatial Characteristics

Quantifying the scanner noise in time domain is important, but its characteristics in space, i.e., across the scanner platen area in a single image, are even more important as our objective is to estimate the scanner pattern from a single image. The upper plot of Figure 5.42 shows an image with air; it is visible that the area with largest row indices (close to 360) and especially close to the edges are consistently brighter than the other areas, suggesting that even with a perfect predetermined object (air), there are gradient effects and other anomalies in the pixel values. This is shown in the lower plot: for the row indices above about 250 for column 120 (roughly the middle column in the image), the pixel values exhibit an upward trend. On the other hand, for the edge column 1, the trend is downward and present across most rows.

For this reason, computing the sample mean along each column and subtracting it from the pixels in a column will not yield an accurate estimate of the scanner noise (nor of the scanner pattern). However, by averaging many pixel values

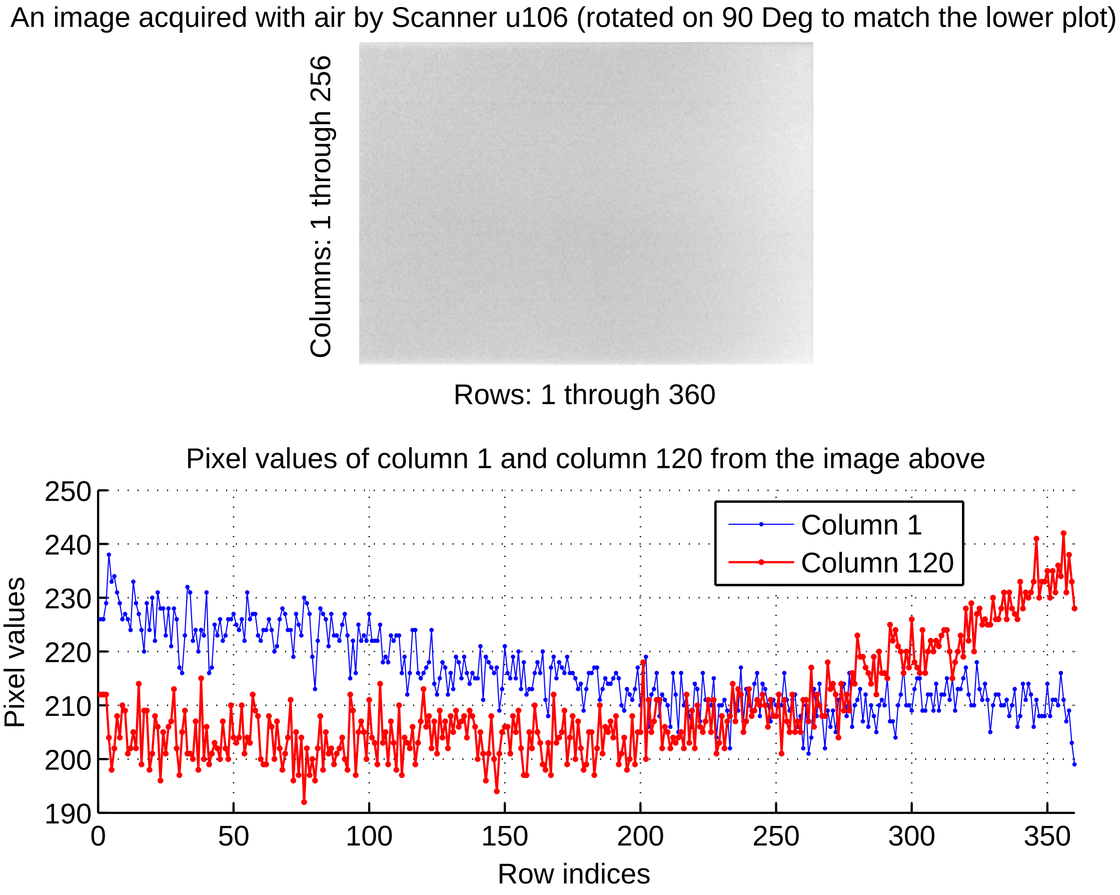


Figure 5.42: Image acquired with air by Scanner u106 and two columns of it

$g^{(po)}(i,j)$ (see Expression 5.16) from images acquired with air sequentially by one and the same scanner, we can compute the best estimate of the scanner pattern $s(i,j)$ because the sample mean over time of the scanner noise $n(i,j,t)$ at each pixel will tend to 0 (the law of large numbers) as with respect to time, the scanner noise is (assumed) a zero-mean random process. Thus, if $g_k^{(po)}(i,j)$ is the pixel value of the k-th image acquired with a particular scanner, then the estimate of the scanner pattern $\hat{s}(i,j)$ at row index i and column index j is:

$$\hat{s}(i,j) = \frac{1}{K}\sum_{k=1}^{K} g_k^{(po)}(i,j) \qquad (5.17)$$

It is important to note that this scanner pattern estimate is only valid for the particular group of images it is estimated from (and for the date and time at which these images were acquired); therefore, K in Expression 5.21 in our case is 100. As already discussed in Section 5.4.2.1, by averaging 100 images, the standard error of the sample mean is about *sqrt(sample variance/sample size)* $\approx \sqrt{2/100} \approx 0.141$, which believe is sufficiently low (the last expression uses 2 as the worst-case scanner noise variance in time domain). Finally, the scanner noise estimate at each pixel (and for each image) is (Expression 5.18):

$$\hat{n}_k(i,j) = g_k^{(po)}(i,j) - \hat{s}(i,j) \tag{5.18}$$

The upper plot of Figure 5.43 shows one column in an image acquired with air and the scanner pattern estimate, and the lower plot shows the scanner noise estimate computed by subtracting the former two. An important observation can be made here: it seems that the scanner noise estimate does not occupy a discrete, finite set of values, i.e., the effect of quantization seems to be absent. This can be explained with the fact that the scanner pattern estimate is an average and thus, unlike the pixel values in an image, it is not over a discrete set of values. This is also visible from Figure 5.44, which shows the EDFs of the scanner noise estimate for two columns (column 120 is the same as in Figure 5.43). It is also important to note (a) the very close similarity between the two EDFs and (b) the outliers that are present in both curves: the very small negative values in column 120 and the very large positive values in column 1.

To compare the EDF of the scanner noise estimate with the Gaussian distri-

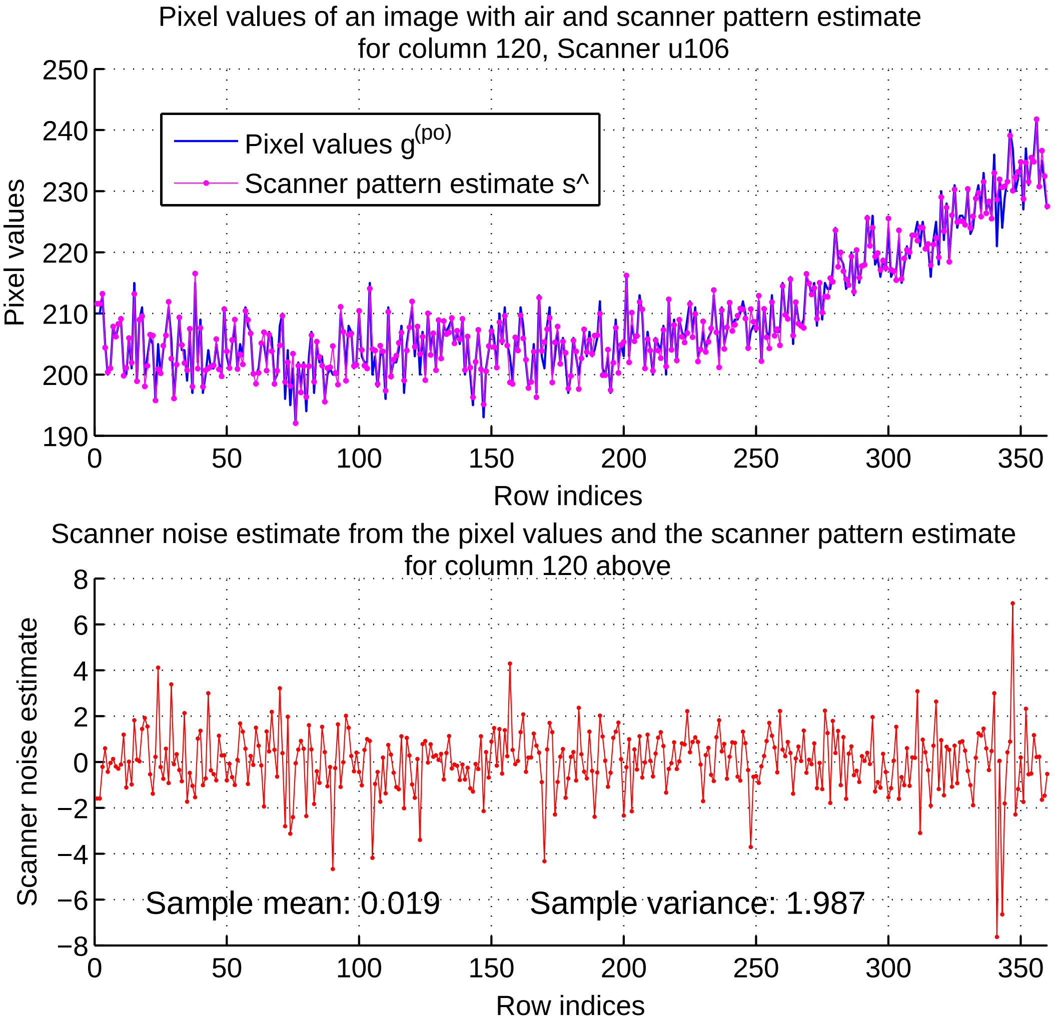


Figure 5.43: Scanner pattern and scanner noise estimates from an image with air

bution, in Figure 5.45 we show the probability plot for column 120 and a Gaussian distribution fitted to it. The extremely tight fit in the middle of the EDF is truly remarkable. The outliers beyond 3σ ($= 4.228$) on both sides around the sample mean (which is very close to 0) are also visible – the two smallest (negative) values and the largest (positive) value. What is peculiar, however, are the values between 2σ ($= 2.819$) and 3σ ($= 4.228$), again on both sides of the sample mean, both circled

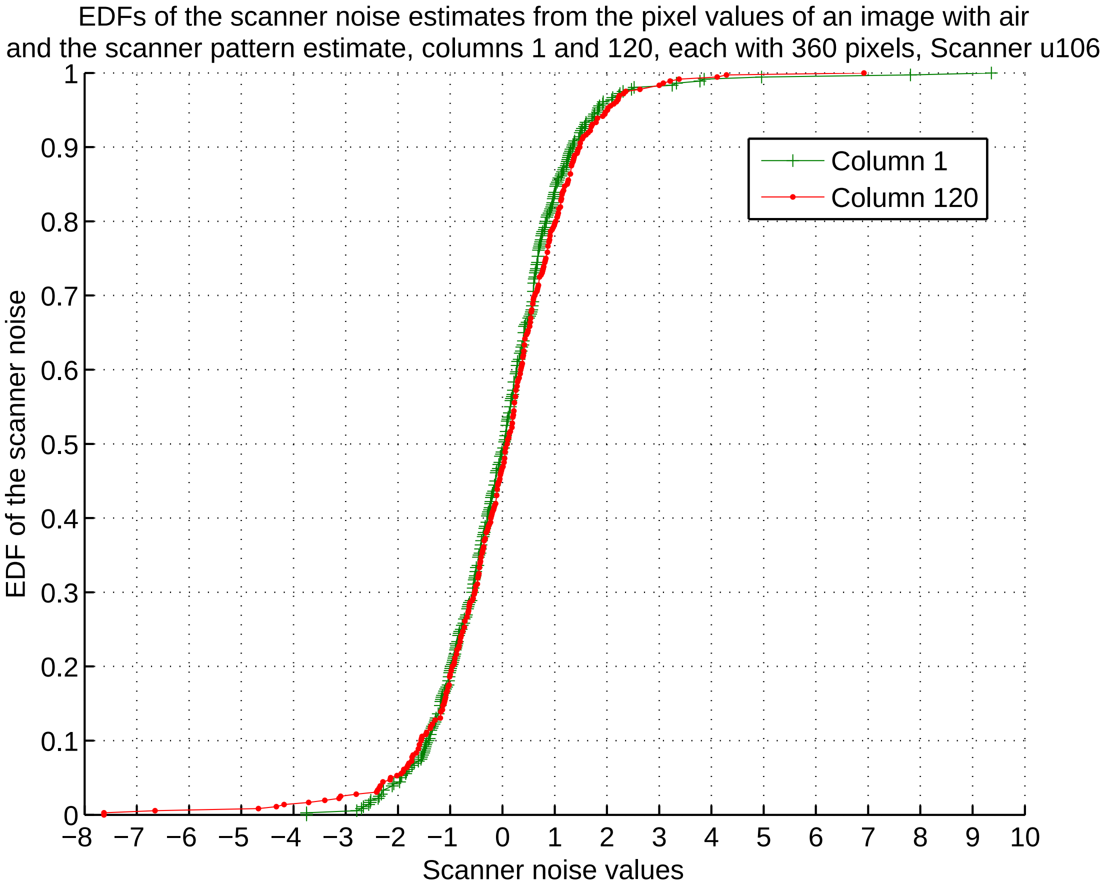


Figure 5.44: EDFs of the spatial scanner noise estimates for two columns (u106)

in the figure, that deviate too much from the Gaussian fit and contribute to heavy tails. (Note: here we use the term “heavy tails” very loosely, in a sense that these tails are above the tails of the Gaussian distribution, which is of importance for us, not that these tails are asymptotically above any decaying exponent). We could not identify the cause for these heavy tails, but considering the otherwise very good fit to the Gaussian distribution and that they are typically beyond $\pm 2\sigma$, we believe they are not of a concern.

Nevertheless, we quantified this deviation from the Gaussian distribution as a

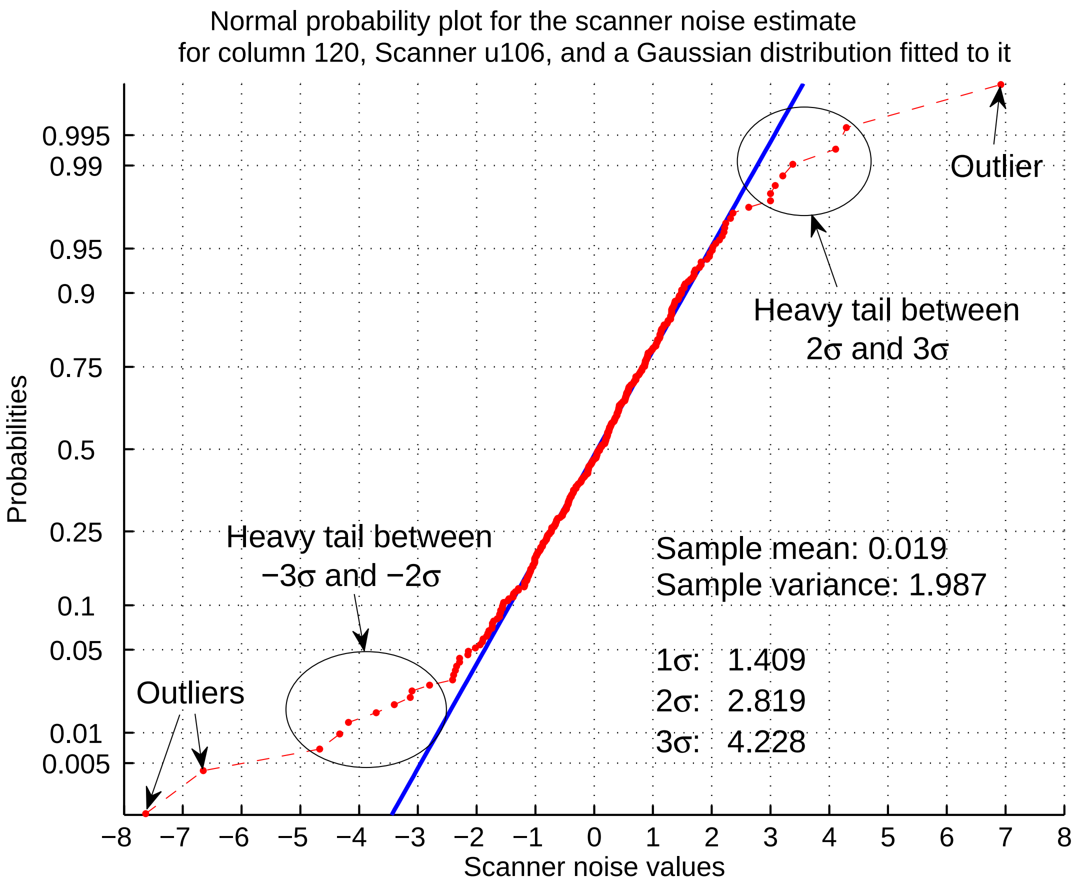


Figure 5.45: Normal probability plot of the scanner noise estimate for column 120

percentage of the columns (in an image) for which the null hypothesis can be rejected by running four hypothesis tests. Figure 5.46 shows the EDFs of the p-values of these tests performed on the scanner noise estimate for all columns of a single image (with air); the 3σ trimming removes the outliers from the data. The χ^2 test, which is the least powerful of the four tests, rejects the smallest percentage of the columns – 5.4% and 16.4% of the columns at 0.01 and 0.05 significance levels, respectively. As the observations in the tails are grouped in the two outer bins, thus effectively make the χ^2 test insensitive to the tails, which in turn suggests that if ignoring the heavy tails, the scanner noise estimates along columns are overwhelmingly Gaussian. Here we did not use the centered χ^2 test because the scanner noise estimates are not explicitly quantized (as explained earlier) and therefore forcing the centers of the bins is not necessary (and may even decrease the accuracy of the test).

The next more powerful test, the Lilliefors test, which is essentially a Kolmogorov-Smirnov test with estimated distribution parameters, rejects 7.8% and 21.9% percent of the columns at 0.01 and 0.05 significance levels, respectively, with both percentages being relatively close to those of the χ^2 test. This suggests that the effect of quantization, although implicitly still present in the scanner noise estimate, does not have significant impact on this EDF-based hypothesis test and hence the latter may be used here. Known to have higher sensitivity in the tails than the Lilliefors/Kolmogorov-Smirnov test, the Anderson-Darling test apparently detects the heavy tails and rejects 2 to 3 times more columns than the χ^2 test: 15.6% and 36.7% for the 0.01 and 0.05 significance levels, respectively.

Finally, the moment-based Jarque-Bera test rejects the largest percentage of columns – 22.7% and 42.2% for the 0.01 and 0.05 significance levels, respectively, most probably because of the effect of the heavy tails on the sample kurtosis (and also possibly on the sample skewness) and thus significantly increasing the Jarque-Bera test statistic. Because of this and because the Jarque-Bera test is known as one of the most powerful tests, we chose it as one of the tests for the scanner noise estimate; as a second test, we chose χ^2 as it is arguably the most liberal test and is least sensitive to heavy tails (and outliers). These two tests, in some sense, give the upper and the lower bound on the percentage of columns for which a Gaussian assumption would be inaccurate.

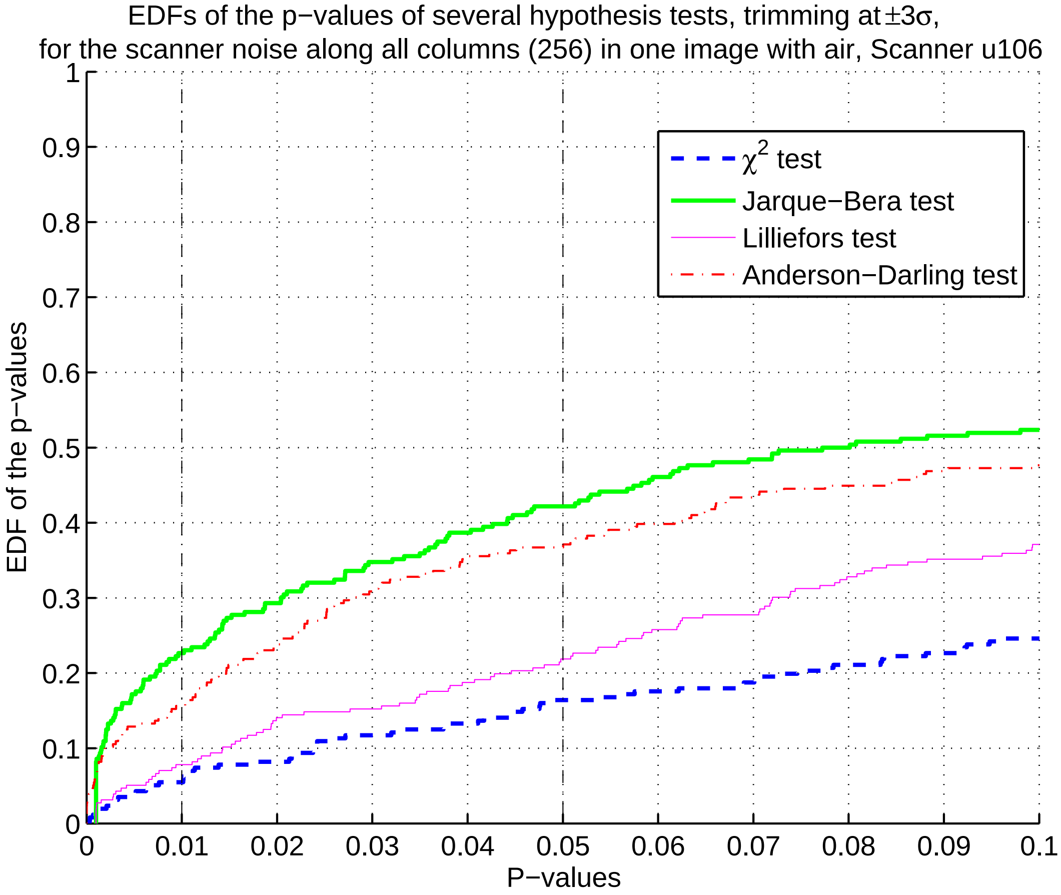


Figure 5.46: EDFs of the p-values of 4 hypothesis tests for the spatial scanner noise

The EDF values of the p-values of the Jarque-Bera and the χ^2 tests at 0.01 and 0.05 significance levels for the scanner noise estimate on all columns for 100 images acquired with air (in one group) by Scanner u106 are shown in Figure 5.47; the 3σ trimming rule is used to exclude outliers from the samples. We also computed an average EDF value of the p-values for each test at each significance level (also shown in the figure) as an average measure; the EDF values appear to vary considerably and using their minima or maxima may clearly lead to inaccurate conclusions. These averages corroborate our earlier observation that the Jarque-Bera test rejects about 3 times more columns than the χ^2 test at 0.01 significance level and about 2 times more columns at 0.05 significance level.

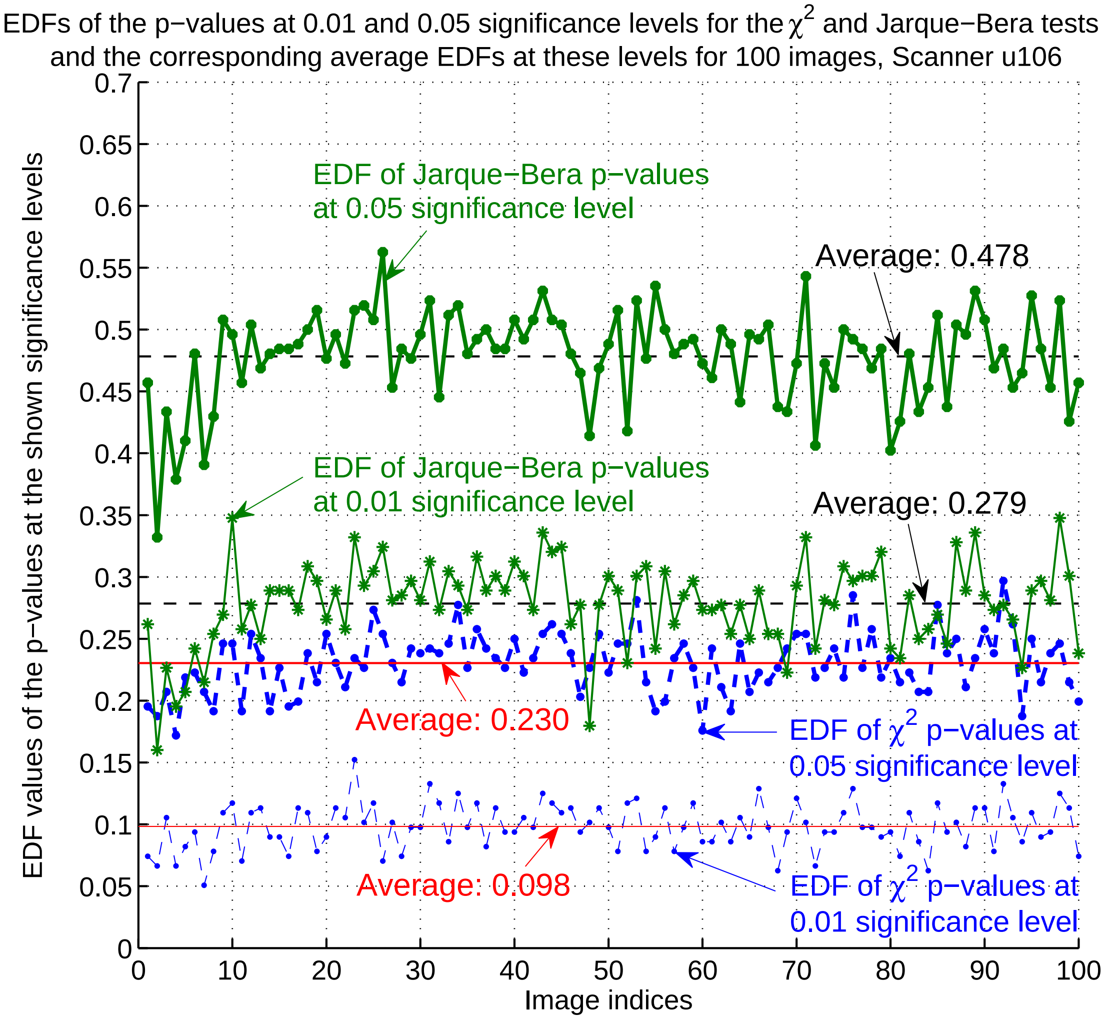


Figure 5.47: EDFs of the p-values at 0.01 and 0.05 significance for χ^2 & Jarque-Bera

Next, we ran the χ^2 and Jarque-Bera tests on all scanners and for all three groups of images, computed the average EDF values of the p-values at 0.01 and 0.05 significance levels for each group of 100 images, and took the maximum of the three averages (as the worst case) as representative for each scanner. The results are shown in Table 6.9 and Table 6.10 in the appendix, Section 6.4.2.3. For the UPEK scanners, based on the χ^2 test, the percentage of columns for which the null hypothesis can be rejected ranges from 7.0% to 14.4% at 0.01 significance level and from 17.5% to 29.6% at 0.05 significance level; both ranges are very similar to the ranges of the percentage of pixels for which the much more conservative Jarque-Bera test rejects the null hypothesis for the scanner noise in time at the same significance levels (see Section 5.4.2.1). The Jarque-Bera test for the scanner noise estimate in space along columns, however, rejects much larger percentage of columns, from 15.4% to 36.2% at 0.01 significance level and from 32.8% to 56.4% at 0.05 significance level. As already discussed, we believe this high percentage of rejection is due to heavy tails, which are detected by the Jarque-Bera test (but not by the χ^2 test), and they are not of a concern because the percentage of pixels for which the scanner noise deviates from normality are beyond $\pm 2\sigma$ or only about 4-5%.

For the Veridicom scanner, the χ^2 test rejects the null hypothesis for 16.2% and 28.9% of the columns at 0.01 and 0.05 significance levels, respectively, which is comparable to the percentages for the UPEK scanners. However, the percentage of columns for which the Jarque-Bera test rejects the null hypothesis here are well above those for the UPEK scanners – 51.5% and 64.8% at 0.01 and 0.05 significance levels, respectively, which we attribute to the much coarser quantization of the scanner noise for the Veridicom scanner.

Estimation of the Variance of the Scanner Noise in Space

For the scanner noise variance in space, we computed the sample variance for the scanner noise estimate along each column and averaged it for all columns in an image. We chose to compute the variance along one dimension instead of for two dimensions because the advanced algorithms process one-dimensional sequence of pixels (e.g., columns or rows). Furthermore, the difference between the computation of the sample variance in the two cases (one and two dimensions) is in the sample mean, subtracted from the value of the scanner noise estimate at each pixel. The sample mean of the scanner noise estimate when computed along one dimension is typically very close to zero, but it is larger in magnitude than the typical sample mean computed across all pixels (two-dimensionally), thus leading to a slightly larger sample variance in the one-dimensional case. We choose columns over rows because:

- We observed that on average the sample variance computed along columns is consistently higher than the variance computed along rows, thus providing a more conservative estimate for the actual scanner noise variance;

- The absolute relative error between the two types of computation (along columns and along rows) for a single image is below 1%;

- The preferred mode of operation of the advanced algorithms is columnwise.

Next, after computing the scanner noise sample variance per image, we averaged these variances for all 100 images in each of group (and per scanner). Then we averaged these three sample variances for each scanner, and finally, all these average variances per scanner we averaged across all scanners, yielding the average scanner noise variance in space. The detailed results are provided in the appendix, Table 6.8, and in summary:

- Signal Model A (UPEK scanners): the sample variance per scanner ranges from 1.343 to 1.991, with average variance across all scanners **1.735**;
- Signal Model B (Veridicom scanner): the sample variance for the scanner we have is **0.869**.

Both average sample variances are very close to the averages for the scanner noise estimate in time (see Section 5.4.2.1).

In conclusion, the distribution of the scanner noise both in time and in space, although overall close to Gaussian, exhibits considerable deviations from normality, which can be due to heavy tails, outliers, and coarse quantization. Even when these effects are factored out from the hypothesis tests, the percentage of pixels or columns for which the Gaussian assumption can be rejected is not negligible. For this reason, we chose methods and designed algorithms that are robust against significant violations of the Gaussian assumption. A confirmation that these deviations are not a problem for our advanced algorithms is their excellent performance. If the signals were truly Gaussian, the algorithms would be optimal, which is desired, but of greater importance for us it that the algorithms are robust (and also implementable).

5.4.3 Scanner Pattern

Because of the presence of the scanner noise, which is a time-varying random process, it is only possible to compute an estimate of the scanner pattern. Such an estimate can be computed from a single image or, preferably, from multiple images acquired with a predetermined object, i.e., not with a fingertip of a person, applied to the scanner platen. These images have to be acquired within a short time period (within several minutes to several tens of minutes).

As already discussed, by averaging many pixel values $g^{(po)}(i,j)$ (see Expressions 5.15 and 5.16) acquired sequentially with one and the same fingerprint scanner we obtained the best estimate of the scanner pattern $s(i,j)$ because the average over time of the scanner noise $n(i,j,t)$ at each pixel will tend to 0 (the law of large numbers as with respect to time; the scanner noise is a zero-mean random process). Thus, if $g_k^{(po)}(i,j)$ is the pixel value of the k-th image acquired with a particular fingerprint scanner, then the estimate of the scanner pattern $\hat{s}(i,j)$ at row index i and column index j is:

$$\hat{s}(i,j) = \frac{1}{K}\sum_{k=1}^{K} g_k^{(po)}(i,j) \tag{5.19}$$

where K is the number of images used for averaging. K can be as small as 10, but, to improve the accuracy of the estimate, we acquired 3 groups of images, each group containing 100 images. The time interval between acquiring Group 1 and Group 2 images was significant (about 1 year and 8 months), the time between acquiring Group 2 and Group 3 images – small (about a couple of days).

The histogram of the scanner pattern estimate from Group 1 images for column 100 of Scanner u103 is shown in Figure 5.48. Besides the main lobe, the histogram shows also a “spread” in the values, which suggests that assuming the distribution of the scanner pattern (in its absolute values) to be Gaussian will be incorrect.

The upper plot of Figure 5.49 shows the scanner pattern estimates, for one

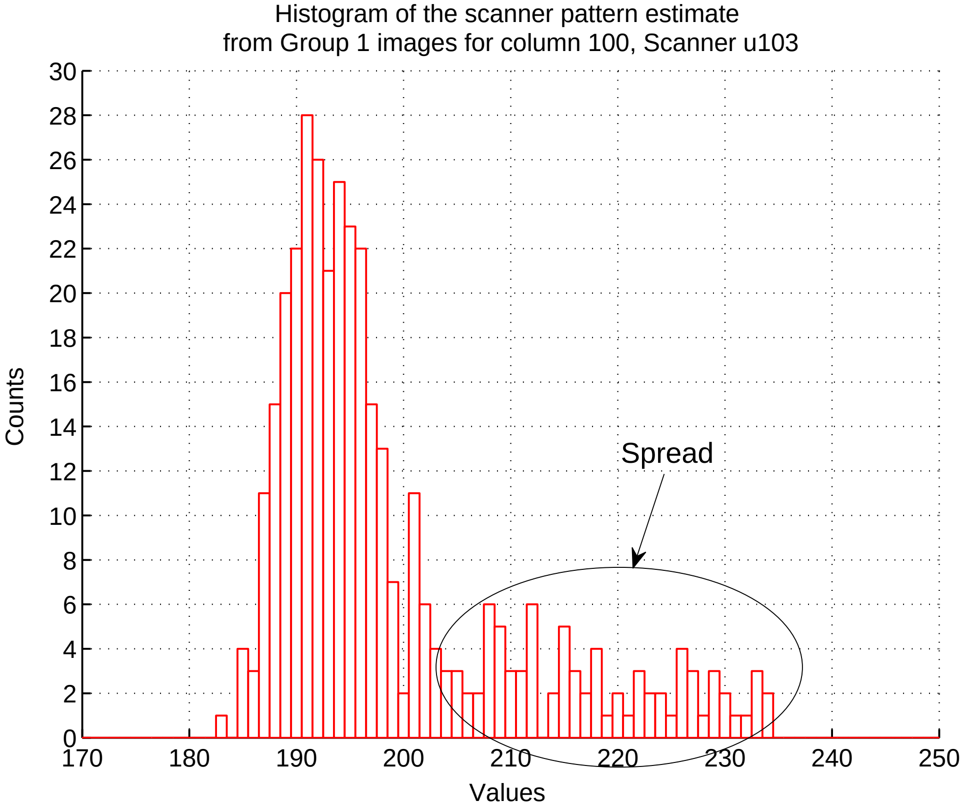


Figure 5.48: Histogram of the scanner pattern estimate for 1 column, Scanner u103

and the same column (100) of one and the same scanner (u103), computed from the three groups of images. Two important observations can be made from it:

1. **Variable mean.** Well visible is the downward trend in the scanner pattern estimates from the first rows to rows of about 200. After that, in the last about 100 rows, the trend reverses, turning steeply upwards and leading to increase by about 40 (from about 190 to over 230). These types of trends are most probably due to gradient effects in the semiconductor elements across the area of the chip. The practical implication for us is that, assuming the scanner pattern is a random process/field, its mean changes (so the process

is nonstationary). This can also explain the spread in the histogram in Figure 5.48.

2. **Offset.** Another observation in the same plot is that, although apparently very similar, the three estimates are slightly different; even their sample means (shown in the upper left corner) differ with about 0.9 from each other. Two of the differences between the estimates are shown in the lower plot on the same figure. The difference between Group 1 and Group 2 estimates is indeed about 1 for most of the rows (and is consistently larger than 0), which can be attributed to the significant time period between the days the two groups of images were acquired, during which the scanner pattern could have indeed changed. It is surprising, however, the about-twice larger difference between Group 1 and Group 3 estimates in comparison with the difference between Group 1 and Group 2, despite the fact that Group 2 and Group 3 images were acquired within 3 days (also visible from the difference between their means on the upper plot). It is also important to note that, although relatively consistent, these differences are not constant for all pixels (as row indices in the figure), and that these variations in the differences are too large to be attributed solely to the inaccuracy in the scanner pattern estimate due to the presence of the scanner noise. For our practical purposes, however, we can call this effect “offset” and can assume it is relatively constant for all pixels in an image.

In sum, because of the variable mean and this (nonconstant) offset, even within short time periods, the absolute value of the scanner pattern (estimate) may create problems for the signal processing and clearly cannot be used as a persistent characteristic of the scanner. (The temperature dependence is another problem which similarly leads to an offset; it is discussed later in this section).

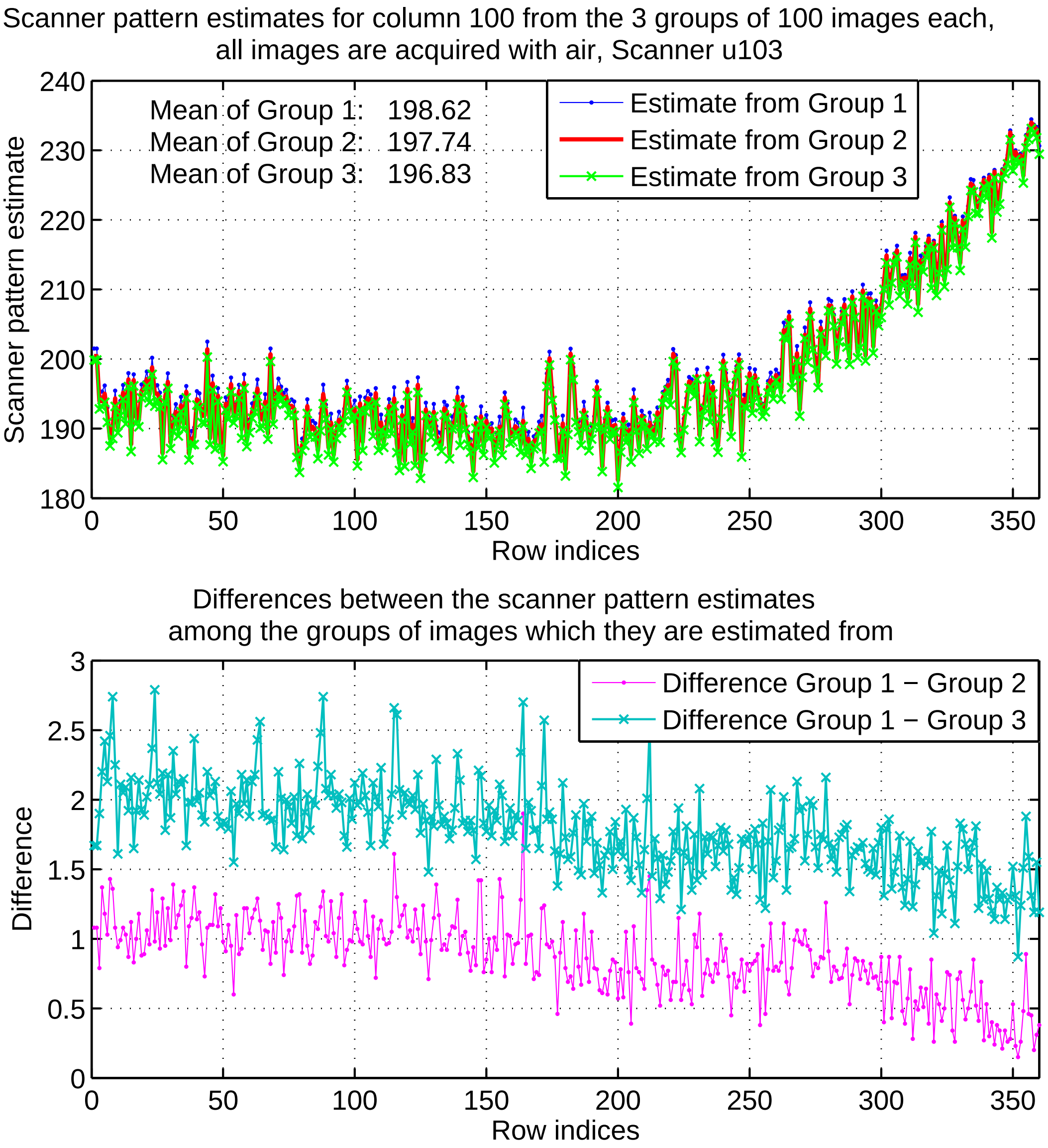


Figure 5.49: Scanner pattern estimates for column 100 from the 3 groups of images (100 images each; each image acquired with air) and their differences, Scanner u103

All these observations led us to the idea to view the scanner pattern as having two components (parts):

a. One component that slowly varies in space but may (considerably) change over time in the long term and also under different environmental conditions and other factors. This component is essentially the mean $\mu_s(i,j)$ of the scanner pattern. Our objective is to remove $\mu_s(i,j)$ from consideration for the scanner authentication because it is not reproducible and cannot serve as a persistent characteristic of each scanner;

b. A second component that rapidly varies in space but is relatively invariant in time (in both the short and the long term) and under different environmental conditions. This variable part $s_v(i,j)$ of the scanner pattern is reproducible and can serve as a persistent characteristic of each scanner. Therefore, our objective is to estimate $s_v(i,j)$ and use it to authenticate the scanners.

Thus, the scanner pattern at pixel (i,j) is the sum of these two components:

$$s(i,j) = \mu_s(i,j) + s_v(i,j) \tag{5.20}$$

We observed that by filtering the scanner pattern estimate $\hat{s}(i,j)$ (see Expression 5.21), for example, with a (possibly noncausal) moving-average filter and then subtracting it from $\hat{s}(i,j)$, we can obtain an estimate of its variable part $\hat{s}_v(i,j)$:

$$\hat{s}_v(i,j) = \hat{s}(i,j) - \mathcal{F}\{\hat{s}(i,j)\}, \tag{5.21}$$

where $\mathcal{F}\{.\}$ denotes the moving-average filtering. This variable part $\hat{s}_v(i,j)$ of the scanner pattern estimate is sufficiently persistent and can be used to identify the scanner. The moving-average filter, in effect, computes the local (sample) mean (estimate) of the scanner pattern, and by subtracting this local mean from the scanner pattern, we obtain the variable part of the scanner pattern; this whole operation is essentially high-pass filtering. Using low-pass filters $\mathcal{F}\{.\}$ other than a moving-average filter is also possible, but we chose the moving-average filter because its filtering is sufficient for our purposes and because it is extremely simple. On the other hand, it is certainly possible to instead use directly a high-pass. The filter $\mathcal{F}\{.\}$ we selected and use for illustration throughout this section is a 11-tap noncausal moving average filter (with 5 zeroes at $2\pi/11 * [1\ 2\ 3\ 4\ 5]$). The filter is one dimensional and works column-wise on the pixels; for the pixels close to the ends of the column (and signal), we pad the signal with its replicas symmetrically about the first and last pixels (the replica padding is explained later).

These variable parts $\hat{s}_v(i, j)$, computed using this 11-tap moving-average filter along columns, of the three scanner pattern estimates $\hat{s}(i, j)$ of Figure 5.49 are shown in Figure 5.50. The scale of the upper plot in Figure 5.49 is much larger than that of the upper plot in Figure 5.50, yet the three variable parts $\hat{s}_v(i, j)$ almost completely overlap. The two differences between them, shown in the lower plot of the same figure, rarely go beyond ±0.5 (in absolute units). The corresponding correlation coefficients are 0.99842 and 0.99778, either one of which is 1 for all practical purposes. Thus, with a simple moving-average filtering, we eliminated both the variable mean and the offset problems, obtaining essentially a persistent and reproducible representation of the scanner pattern.

In studying the statistical characteristics of the scanner pattern, we used the

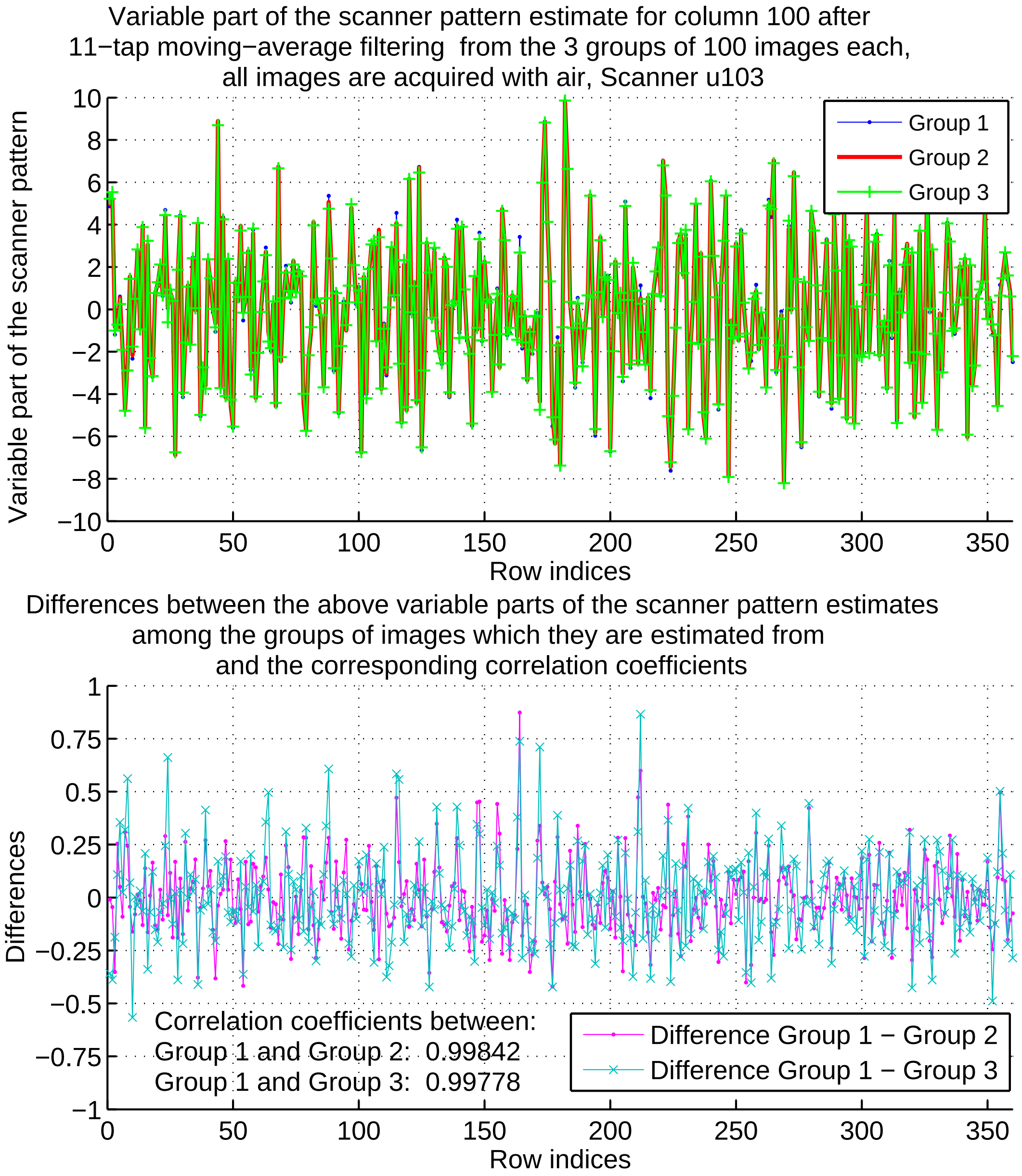


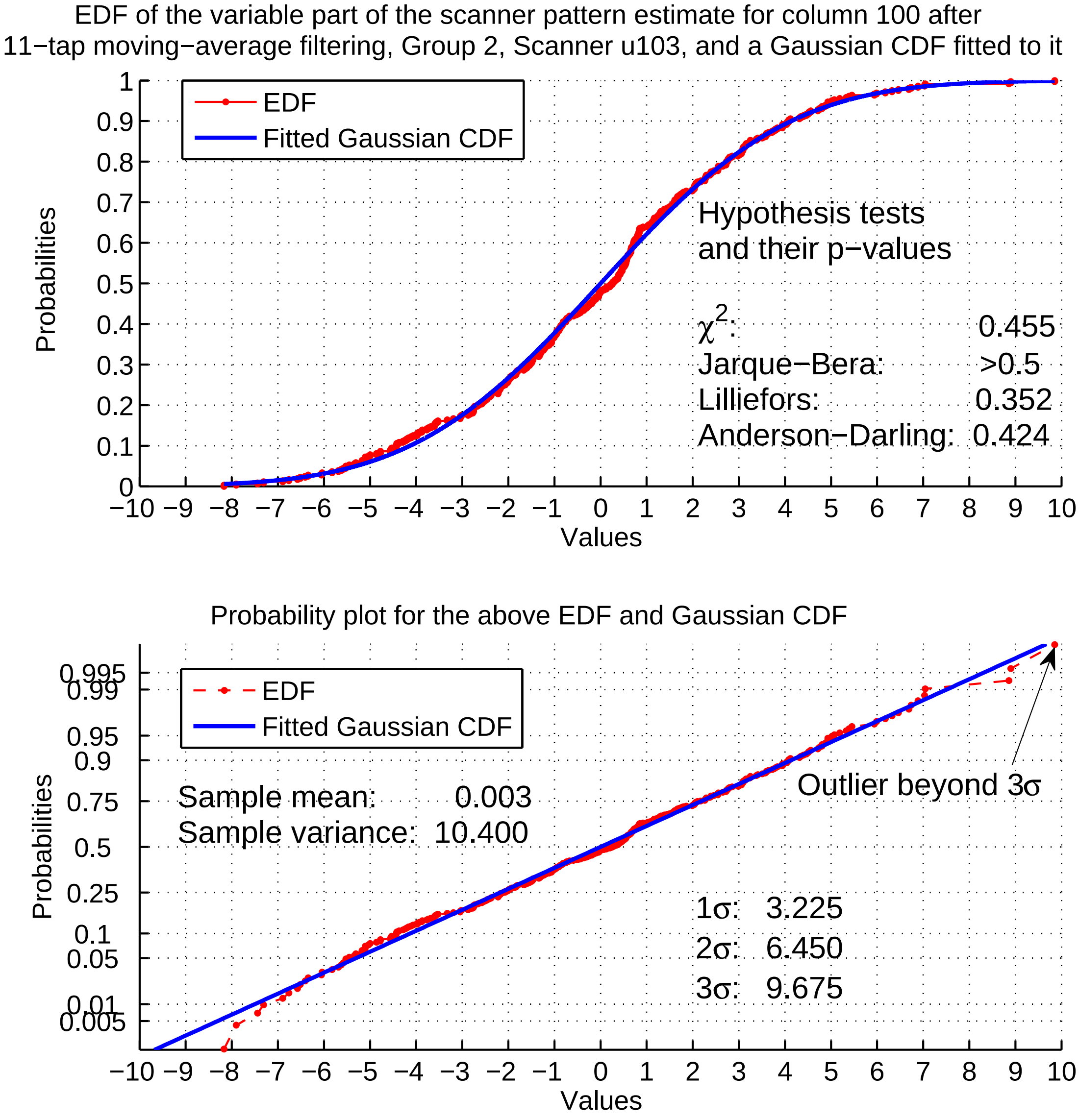
Figure 5.50: Variable parts, obtained with a 11-tap moving-average filter, of the scanner pattern in Figure 5.49 and their differences and correlation coefficients

same approach as for the scanner noise. Figure 5.51 shows the EDF of the variable part of the scanner pattern estimate for column 100 (of Scanner u103, computed from Group 2 images) by using a 11-tap moving-average filter; this is the same signal shown in the upper plot of Figure 5.50. A Gaussian CDF is fitted to it using as parameters the sample mean and the sample variance, shown in the lower plot. As visible from both plots, the distribution is very close to Gaussian.

To quantify the Gaussian hypothesis, we ran four tests (χ^2, Jarque-Bera, Lilliefors, and Anderson-Darling), the p-values of all of which, shown in the upper plot, are larger than 0.35 and thus well above 0.05, therefore failing to reject the null hypothesis. While the large p-values of the χ^2 and Jarque-Bera tests are to be expected in case of so good visual fit, it is surprising that the EDF-based Lilliefors and especially the Anderson-Darling test also fail to reject the null hypothesis. This suggests that the quantization, which was a serious problem for these tests when performed on the scanner noise, here appears to have very limited impact. One of the reasons for this is that the range of the variable part of the scanner pattern is sufficiently large with respect to the quantization step. However, there are two more veiled reasons: (1) the signal under test is the variable part of the scanner pattern, which means after the moving average filtering, and (2) it stems from the scanner pattern estimate, which means after averaging 100 images (in the group). Therefore, it is no surprise that after these two operations, no distinct quantization levels are present.

While studying this variable part of the scanner pattern, we also observed that although rare, it do sometimes contain outliers beyond $\pm 3\sigma$. Figure 5.51 shows one such outlier, the only one in this particular signal; the outlier is only slightly farther than 3σ away from the sample mean. Because of such outliers, we also use the "3σ trimming" when performing hypothesis tests on the scanner pattern. (Note:

none of the four hypothesis tests in the previous paragraph trimmed any outliers, yet they all failed to reject the null hypothesis.)



EDF of the variable part of the scanner pattern estimate for column 100 after 11−tap moving−average filtering, Group 2, Scanner u103, and a Gaussian CDF fitted to it

Probability plot for the above EDF and Gaussian CDF

Figure 5.51: EDF of the variable part of the scanner pattern estimate for column 100, a Gaussian CDF fit to it, and hypothesis tests (Group 2, Scanner u103)

We quantified the deviation from the Gaussian distribution as the percentage of columns for which the null hypothesis can be rejected. Figure 5.52 shows the

EDFs of the p-values of four hypothesis tests performed on the variable part $\hat{s}_v(i, j)$ of the scanner pattern, estimated from the images in Group 1, for all columns; the 3σ trimming removes the outliers from the sample. We look at the EDFs at 0.01 and 0.05 significance levels. The χ^2 and the Lilliefors tests reject a very similar percentage of columns: χ^2 rejects 1.6% and 4.7% at each significance level, respectively, and Lilliefors – 1.2% and 4.7%, respectively. Since we know that the χ^2 test is relatively weak and that the EDF-based Lilliefors test is weaker than the Anderson-Darling test, we will not use any one of the former two. The similarly EDF-based Anderson-Darling test is known to be more sensitive in the tails; here it rejects 1.2% and 6.6% of the columns at the two significance levels, respectively. For this reason, we chose the Anderson-Darling test as the worst case guidance. Somewhat surprisingly, the moment-based Jarque-Bera rejects a much smaller percentage of columns: 1.6% at 0.05 significance level and rejects no columns at 0.01 significance level. This suggests that the sample skewness and kurtosis of an overwhelming number of columns is in line with the Gaussian distribution; this can also be inferred from the apparent absence of heavy tails (see Figure 5.51). Therefore, we use Jarque-Bera test as another, liberal test for normality. These two tests, in some sense, give the upper and the lower bound on the percentage of columns for which a Gaussian assumption would be inaccurate.

Next, we ran the Jarque-Bera and the Anderson-Darling tests on all scanners and computed the p-value for each column of the variable part $s_v(i, j)$, computed with an 11-tap moving-average filter, of the scanner pattern as estimated from each group of images. Then we followed our standard procedure: computed the EDFs of

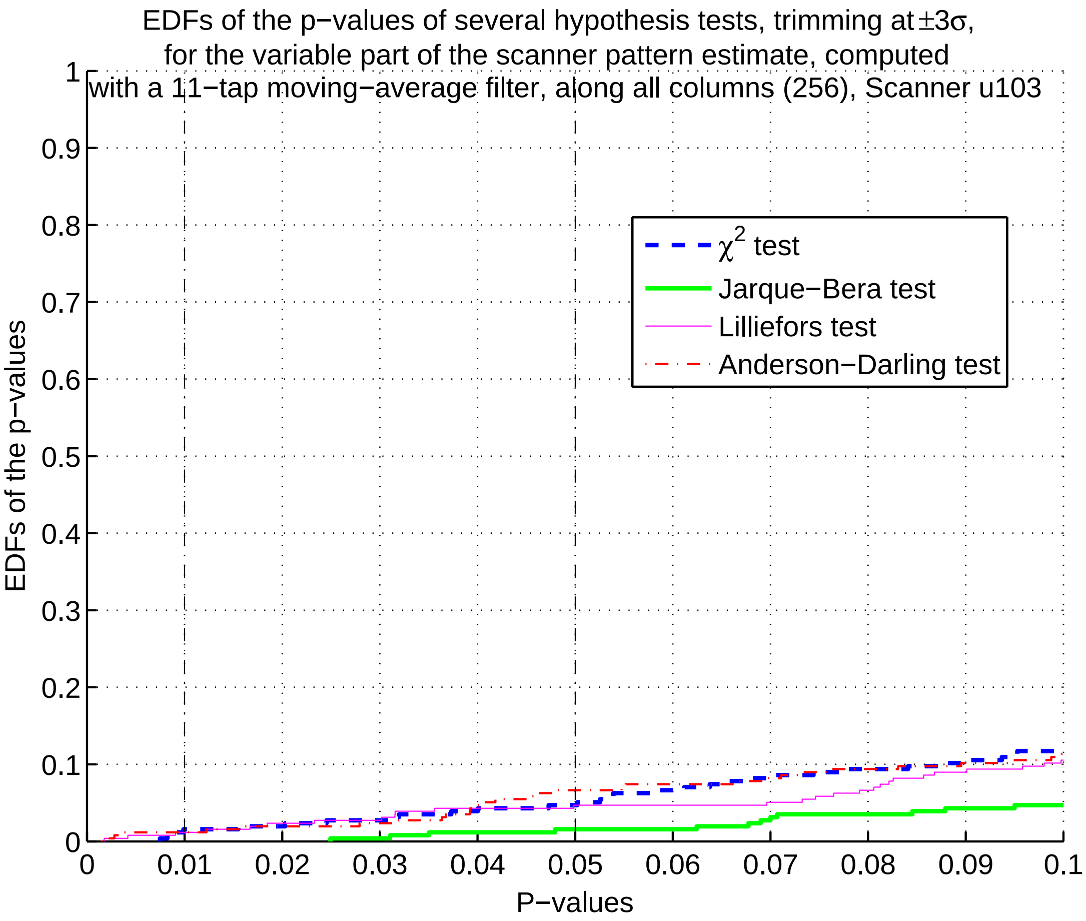


Figure 5.52: EDFs of the p-values of four hypothesis tests along all columns for the variable part of the scanner pattern estimate, Scanner u103

the p-values at 0.01 and 0.05 significance levels for each of the 3 groups and took the maximum of the three (as the worst case) as representative for each scanner. The results are shown in the appendix, Tables 6.12 and 6.13. The Jarque-Bera test, for either scanner type, fails to reject the null hypothesis for hardly any column: rejected are at most 0.4% of the columns (which is 1 column out of all 256 columns) at 0.01 significance level and at most 3.5% of the columns at 0.05 significance level; for the Veridicom scanner, the percentage is even lower: 1.6%. This essentially means that the sample skewness and kurtosis are in line with those of a Gaussian distribution. The percentage of columns rejected by the Anderson-Darling test for the UPEK scanners is somewhat larger, but still very low: at most 2.3% at 0.01 significance level and in the range from 3.9% to 8.2% at 0.05 significance level. For the Veridicom scanner, the percentages are 2.3% and 5.5% at 0.01 and 0.05 significance levels, respectively. This was expected as we already saw the absence of heavy tails; what is surprising, however, is that for the scanner pattern, the quantization step is sufficiently small as to not influence the test statistic, both for the UPEK and the Veridicom scanners.

One important consideration, which will become clear in the discussion of the advanced algorithms, is that since the advanced algorithms use a single image (even containing a fingerprint pattern), the signal processing modules "see" not only the "clean" scanner pattern, but the combination of the scanner pattern and the scanner noise (along the fingerprint pattern where present). Moreover, separating the variable part $s_v(i,j)$ of the scanner pattern from the scanner noise $n_k(i,j)$ is virtually impossible as both have zero-mean approximately Gaussian distributions and, therefore, their sum has also a (zero-mean) approximately Gaussian distribution. Thus, in essence, the advanced algorithms estimate and use (for matching) the composite signal: the variable part of the scanner pattern and the scanner noise: $s_v(i,j) + n_k(i,j)$.

For this reason, we ran the Jarque-Bera and the Anderson-Darling tests also on the variable part (i.e., $s_v(i,j) + n_k(i,j)$) of the pixel values $g^{(po)}(i,j)$ along columns, computed with an 11-tap moving average filter, for each image in each group of images (similarly to the temporal scanner noise analysis). Then we computed the EDFs of the p-values at 0.01 and 0.05 significance levels for each image and averaged these EDFs within each group of 100 images. Finally, we took the maximum of the three averages (as the worst case) as representative for each scanner. The results are shown in the appendix, Tables 6.15 and 6.16. There are small differences (per scanner) with respect to the results for the case only with the variable part of the scanner pattern, but for practical purposes the conclusion is the same: with very high confidence, the distribution can be assumed Gaussian. In summary, for the Jarque-Bera for the UPEK scanners, there is no difference at 0.01 significance level; at 0.05 significance level, the percentage of columns **on average** ranges from 1.1% to 2.0% and is slightly lower than for the scanner-pattern-only case because here we have the average EDF values (at the particular significance levels) among the 100 images. For the Veridicom scanner, at 0.05 significance level, the percentage is slightly higher (2.5%) but still negligible. The percentages of rejected columns by the Anderson-Darling test for the UPEK scanners are similarly close to the scanner-pattern-only case: at most 1.7% on average at 0.01 significance level and in the range from 4.0% to 6.9% on average at 0.05 significance level. For the Veridicom scanner, the percentages of rejected columns is also slightly higher at 0.05 significance level: 6.1% against 5.5% for the scanner-pattern-only case.

In conclusion, the variable part of the scanner pattern can be very safely assumed to have a Gaussian distribution as the percentage of columns for which this assumption can be rejected is negligible. Furthermore, despite the considerable percentage of columns for which the spatial scanner noise cannot be assumed Gaussian, the variable part of the composite signal {scanner pattern and spatial scanner noise} can also be very safely assumed Gaussian. Nevertheless, similarly to the scanner noise, here some deviations from normality are also present, in particular outliers and possibly heavy tails, for which reason they were excluded from the hypothesis tests. For this reason, we chose methods and designed algorithms that are robust against significant violations of the Gaussian assumption; a confirmation that these deviations are not a problem for our advanced algorithms is their excellent performance. If the signals were truly Gaussian, the algorithms would be optimal, which is desired, but of greater importance for us it that the algorithms are robust (and also implementable).

Scanner Pattern Dependence from Temperature

We observed that the scanner pattern depends significantly on the temperature, which is not surprising as it is related to the gain of an amplifier. We studied this dependence by acquiring images with air at various temperatures; details about the thermal tests are given in the appendix, Section 6.2. Because of the strong thermal dependence of the scanner pattern and the difficulty in controlling precisely the temperature of the scanner platen, it was not possible to acquire a set of images at exactly the same temperature (for other than the room temperature), from which images to estimate the scanner pattern. For this reason, for these temperatures we use directly the pixel values $g^{(po)}$, which contain both the scanner pattern and the scanner noise.

Figure 5.53 illustrates the thermal dependence of the scanner pattern. The upper plot shows column 100 (of Scanner u103) with the pixel values $g^{(po)}$ at low temperature (about 5 ^{o}C), the scanner pattern estimated from 100 images with air at room temperature, and the pixel values $g^{(po)}$ at high temperature (about 40 ^{o}C); the plot shows only rows 260 through 360 for better visibility. The lower plot shows the variable part of these signals, computed by filtering with an 11-tap moving-average filter. Despite the scanner noise, present in the variable parts of the pixel values $g^{(po)}$, it is remarkable how close to each other these variable parts are. Their close similarity can also be seen in the very large correlation coefficients: 0.922 and 0.960 for the two temperatures and the room temperature and 0.885 for the low and high temperatures. The correlation coefficients are computed for all 360 rows, not only for the shown 100 rows.

For the UPEK scanners, the scanner pattern shifts down when lowering the temperature (and the pixel values become smaller), and shifts up when increasing the temperature (and the pixel values become larger). A temperature difference of 34 ^{o}C changes the scanner pattern with about 30 grayscales levels, and we expect that a temperature change of 40 ^{o}C (from 0 ^{o}C to 40 ^{o}C) will change the scanner pattern with about 35-40 grayscale levels, which is considerable. It is, therefore, required that the algorithms be robust against such variations, as our advanced algorithms are.

Scanner Pattern Dependence from Water

It is well known that wet (as well as dry) fingertips create problems for the

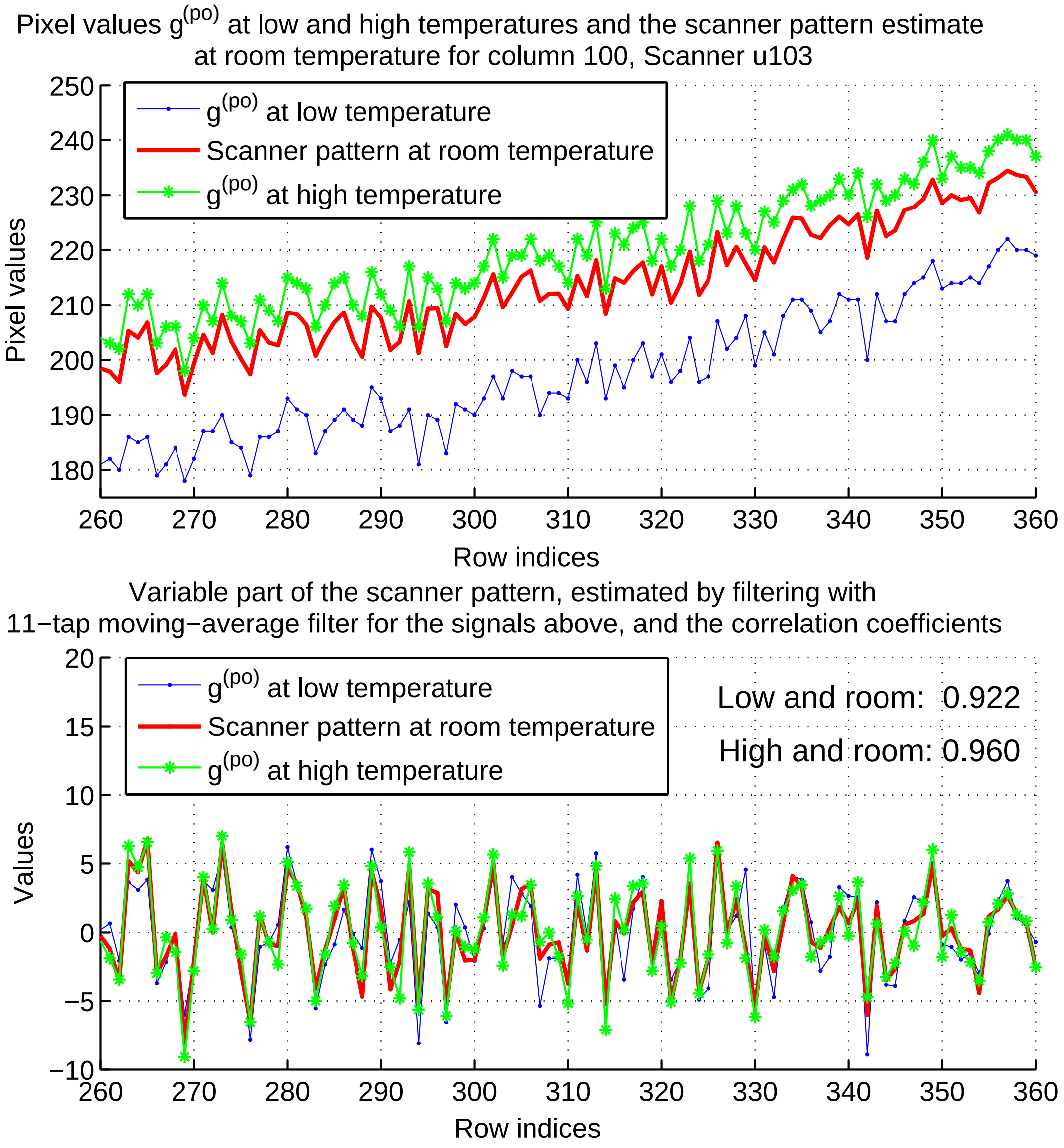


Figure 5.53: Thermal dependence of the scanner pattern (and noise) at 3 temperatures, their differences, and the correlation coefficients (column 100, Scanner u103)

fingerprint authentication. Although our signal to be authenticated is different, we observed that the scanner pattern changes when water, especially in abundant quantities, is present on the scanner platen. One would expect that water, similarly to air, would also be an ideal predetermined object as (a) water is homogeneous, (b) water naturally covers the scanner platen uniformly because water is a liquid, and (c) water adheres tightly to the scanner platen. Water is indeed a predetermined object (and generally could be used for scanner pattern estimation), but its effect on the scanner pattern proved to be very different from that of air and using it as predetermined object for estimating the scanner pattern has be done with great care.

Water affects the scanner pattern in a similar way as lowering temperature does – it shifts the scanner pattern down – but with two major differences:

- The shift is not uniform (or not even nearly uniform) across the platen area: close to the platen edges, the scanner pattern decreases much more than the scanner pattern in the middle does.

- The shift is much larger in magnitude; it can even be over 100 grayscale levels.

These effects of water are illustrated in Figure 5.54. The scanner pattern estimate in the figure is computed by averaging 100 images with air. The pixel values $g^{(w)}$ are when water covers completely the scanner platen, from edge to edge.

Although the absolute scanner pattern changes significantly, its variable part, remains substantially unchanged. The variable parts of the signals of Figure 5.54, computed by filtering with a 11-tap moving average filter, are shown in the upper plot of Figure 5.55. Although the variable parts seems to also be considerably different, the correlation coefficients (shown in the lower plot) between the signals are very close to 1: 0.907 between the first image with water and the scanner pattern estimate and 0.856 between the second image with water and the scanner pattern

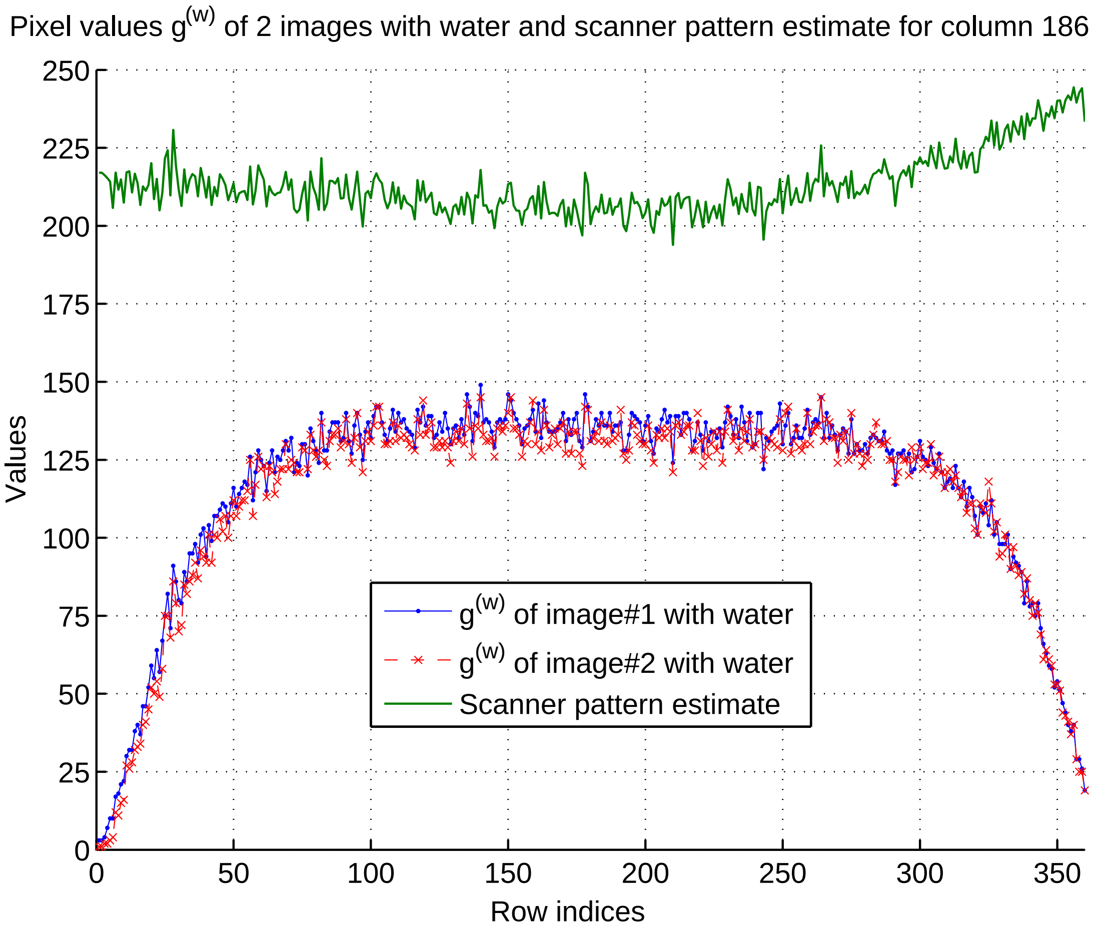


Figure 5.54: Effect of water on the scanner pattern: pixel values of two images with water and the scanner pattern estimate, all for column 186

estimate. The correlation coefficient between the two images with water is also very high: 0.832. The correlation coefficients are computed for all 360 rows, not only for the shown 100 rows. The lower plot also shows the differences between the variable parts, from which it is clear that the differences for certain pixels can be considerable.

Obviously, this peculiar effect of water on the scanner pattern requires that this problem be considered and solved, as our advanced algorithms do.

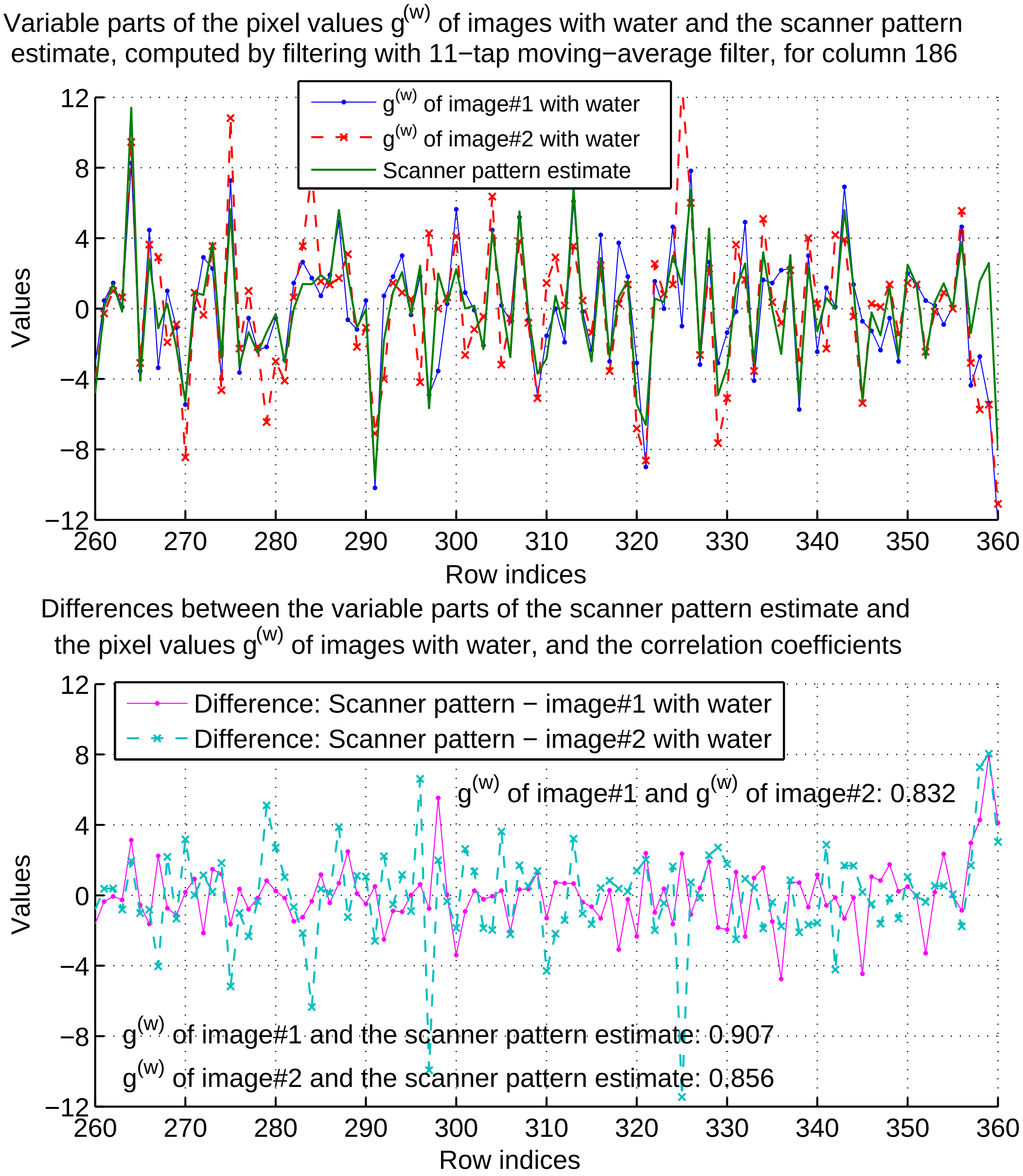


Figure 5.55: Effect of water on the scanner pattern: the signals of Figure 5.54, filtered with an 11-tap moving-average filter, their differences, and correlation coefficients

Estimation of the Scanner Pattern Variance

The main conclusion of the preceding discussions is that the signal of interest is the variable part of the scanner pattern. The statistical characteristic that quantifies this variable part is the variance σ_s^2 of the scanner pattern, and it needs to be either known or computed from image(s). Since the scanner pattern is a random field, for which only a finite amount of data is available, and because of the presence of the scanner noise, it is only possible to compute an estimate of σ_s^2. To do so, as already explained, we first need to compute the scanner pattern estimate $\hat{s}(i,j)$ at each pixel by pixel-wise averaging images with air.

One approach to estimating the scanner patter variance is to first compute the (global) sample mean $\bar{s}$, which is an estimate for the mean μ_s, of the scanner pattern:

$$\bar{s} = \frac{1}{I.J} \sum_{i=1}^{I} \sum_{j=1}^{J} \hat{s}(i,j) \tag{5.22}$$

where I is the number of rows and J is the number of columns in the image. The estimate $\hat{\sigma}_s^2$ of the scanner pattern variance σ_s^2 can then be computed with:

$$\hat{\sigma}_s^2 = \frac{1}{I.J} \sum_{i=1}^{I} \sum_{j=1}^{J} (\hat{s}(i,j) - \bar{s})^2 \tag{5.23}$$

Instead of the biased estimate in Expression 5.23, it is also possible to compute the unbiased estimate by dividing by $(I-1)(J-1)$ instead of by $I.J$, although in our case the difference between the two will be small because I and J are very large (several hundreds). The problem with this approach is that, as already demonstrated (see Figure 5.49), the mean μ_s of the scanner pattern is not constant, and it also depends on the temperature and the moisture, which may result in an overly conservative (large) estimate for the variance σ_s^2 and thus possibly to suboptimal overall performance.

Therefore, a better approach is to compute local estimates $\hat{\mu}_s(i,j)$ for the mean of the scanner pattern at each pixel (at row index i and column index j) by averaging the pixel values in blocks of pixels:

$$\hat{\mu}_s(i,j) = \frac{1}{L.R} \sum_{l=-\lfloor \frac{L}{2} \rfloor}^{\lfloor \frac{L-1}{2} \rfloor} \sum_{r=-\lfloor \frac{R}{2} \rfloor}^{\lfloor \frac{R-1}{2} \rfloor} \hat{s}(i+l, j+r) \tag{5.24}$$

where the integers L and R define the block over which the local estimate is computed, with best results achieved for L and R in the range from about 5 to about 20. When the index $(i+l)$ (or the index $(j+r)$) falls outside the image boundaries, the size of the block is reduced to accommodate this by decreasing L (or R, respectively).

However, in the context of the one-dimensional signal processing (e.g., pixels in columns or rows) of the advanced algorithms, we computed local estimates $\hat{\mu}_s(i,j)$ of the mean μ_s in one dimension (instead of in two dimensions). This essentially is filtering the sequence of pixels with the moving-average filter as detailed earlier. This, in general, can be done along rows, along columns, or along any one-dimensional cross section of the image, but since the preferred mode of operation of the advanced algorithms is columnwise, we computed the local means columnwise. Hence, computing $\hat{\mu}_s(i,j)$ at each pixel with row index i and column index j can be done by averaging the neighboring pixels in the column j, and Expression 5.24 reduces to:

$$\hat{\mu}_s(i,j) = \frac{1}{L} \sum_{l=-\lfloor \frac{L}{2} \rfloor}^{\lfloor \frac{L-1}{2} \rfloor} \hat{s}(i+l, j) \tag{5.25}$$

We used $L = 11$, but any L in the range from about 5 to about 20 can give good estimates.

The estimate $\bar{s}$ for the scanner pattern mean and the local estimates $\hat{\mu}_s(i,j)$ also depend on the signal model, and for Signal Model A they are in the range from about 150 to about 220 and for Signal Model B – in the range from about 200 to about 250. The estimates $\bar{s}$ and $\hat{\mu}_s(i,j)$ also depend on the particular scanner and the conditions under which they are estimated (e.g., temperature and moisture), and the local estimates $\hat{\mu}_s(i,j)$ also varies across the image.

Next, the sample variance $\hat{\sigma}_s^2$ is computed using the local means $\hat{\mu}_s(i,j)$:

$$\hat{\sigma}_s^2 = \frac{1}{I.J}\sum_{i=1}^{I}\sum_{j=1}^{J}(\hat{s}(i,j) - \hat{\mu}_s(i,j))^2 \tag{5.26}$$

Here, similarly to computation of the local means $\hat{\mu}_s(i,j)$, instead of the two-dimensional average, we computed one-dimensional sample variances $\hat{\sigma}_s^2(j)$ along each column j:

$$\hat{\sigma}_s^2(j) = \frac{1}{I}\sum_{i=1}^{I}(\hat{s}(i,j) - \hat{\mu}_s(i,j))^2, \tag{5.27}$$

which $\hat{\sigma}_s^2(j)$ afterwards are averaged to compute the sample variance $\hat{\sigma}_s^2$ for the particular scanner:

$$\hat{\sigma}_s^2 = \frac{1}{J}\sum_{j=1}^{J}\hat{\sigma}_s^2(j). \tag{5.28}$$

Similarly to the computation of the variance of the scanner noise in space, the difference between the computation of the sample variance of the scanner pattern in the two cases (one-dimensional and two-dimensional) is in the mean, global or local, subtracted from the scanner pattern estimate $\hat{s}(i,j)$ at each pixel. According to our observations, the difference is sufficiently small to be neglected. Furthermore, the two-dimensional computation takes significantly more time, which may be a problem if the scanner pattern variance is estimated by the system during enrolment.

After computing the sample variances of the scanner pattern, estimated by averaging within each group of 100 images, we averaged these three sample variances to compute the average scanner pattern variance $\hat{\sigma}_{s,m}^2$ for each scanner m. Finally, we computed the average scanner pattern variance $\hat{\sigma}_s^2$ across our batch of M scanners:

$$\hat{\sigma}_s^2 = \frac{1}{M} \sum_{m=1}^{M} \hat{\sigma}_{s,m}^2 \tag{5.29}$$

The detailed results are provided in the appendix, Table 6.11, and in summary:

- Signal Model A (UPEK scanners): the sample variance per scanner ranges from 12.459 to 19.317, with an average variance across all scanners **15.423**;
- Signal Model B (Veridicom scanner): the sample variance for the scanner we have is **8.645**.

An alternative to computing an average scanner pattern variance across a set of scanners is to compute $\hat{\sigma}_s^2$ for the particular scanner and then adjust the algorithm parameters accordingly, and this can be done during the scanner enrolment; only about 10 images acquired with air are sufficient to yield an accurate estimate $\hat{\sigma}_s^2$. Using a predetermined object, however, has a downside as it requires that during the scanner enrolment, two groups of images be acquired: one group with a predetermined object and another group with user's fingerprint, thus potentially increasing the computational time for enrolment and, much worse, weakening the security because the legitimate scanner may be replaced between the acquisitions of the two groups of images.

Computing and using an accurate estimate of the scanner pattern variance is important, but the advanced algorithms are sufficiently robust, i.e., the overall performance remains relatively unchanged, against wide deviations of the estimate $\hat{\sigma}_s^2$ from the true value. Therefore, of greater importance for is not the variance of the scanner pattern alone, but the variance of the sum of the scanner pattern and the spatial scanner noise. Thus, instead of estimating the scanner pattern $\hat{s}(i,j)$ by averaging images with air (in Expression 5.19), we used the pixel values of a single image $g^{(po)}(i,j)$ instead of $\hat{s}(i,j)$ to compute the local mean $\hat{\mu}(i,j)$ in Expression 5.25. This $\hat{\mu}(i,j)$ we in turn used to compute the sample variances $\hat{\sigma}(j)$ for each column j:

$$\hat{\sigma}^2(j) = \frac{1}{I}\sum_{i=1}^{I}(g^{(po)}(i,j) - \hat{\mu}(i,j))^2, \tag{5.30}$$

which $\hat{\sigma}^2(j)$ afterwards were averaged to compute the sample variance $\hat{\sigma}^2$ of a particular scanner:

$$\hat{\sigma}^2 = \frac{1}{J}\sum_{j=1}^{J}\hat{\sigma}^2(j). \tag{5.31}$$

The detailed results are provided in the appendix, Section 6.14, and in summary:

- Signal Model A (UPEK scanners): the sample variance per scanner ranges from 13.678 to 20.952, with an average variance across all scanners **17.00**;

- Signal Model B (Veridicom scanner): the sample variance for the scanner we have is **9.436**.

It is good to note that these variances are about 1.2 to 2.0 higher than the corresponding scanner pattern variances $\hat{\sigma}_s^2$, which is in line with the variance of the sum of two Gaussian random variables.

In conclusion, the scanner pattern, viewed as a two-dimensional random process, i.e., a random process dependent on two independent spatial variables, can be well approximated by a Gaussian random field, i.e., with the two-dimensional random variable having a Gaussian distribution $N(\mu_s, \sigma_s^2)$, where μ_s is the mean and σ_s^2 is the variance of the scanner pattern. The random field is not necessarily stationary in the mean, i.e., the mean μ_s may change across one and the same image (e.g., as a gradient effect) and among different images acquired with the same fingerprint scanner under different environmental conditions, e.g., under different temperatures or different moistures. For each fingerprint scanner, the variations of the scanner pattern around this mean do not change significantly and are relatively stable under different conditions (thus, the variance σ_s^2 is relatively constant); these persistent variations around the (possibly variable) mean determine the part of the scanner pattern that we use for scanner authentication.

Scanner Pattern Spatial Dependence

Studying the scanner pattern spatial dependence proved to be much more difficult than anticipated, even for the simplest possible measure of dependence – its second order statistics, i.e., the correlation within the signal.

Initially, we tried to study the PSD of the scanner pattern by two approaches: (1) using the DFT of the autocorrelation function of the variable part of the scanner pattern (along columns) and (2) averaging DFTs of these variable parts. However, neither approach yielded conclusive results. We did observe spectral peaks that can reach several dBs above the average PSD, but the frequencies of these peaks are not constant and vary across columns, even adjacent ones, and it is difficult to attribute them to consistent patterns (per scanner or across scanners), especially ones that can be exploited and used systematically in the advanced algorithms. Rather, the PSD looks more like the PSD of a noise (filtered by a high-pass filter, explained below), and the peaks can be also attributed to artifacts of the PSD estimation (e.g., from the finite signal length). The only definite observation is that, unlike for the UPEK scanners, the scanner pattern of the Veridicom scanner has a consistent peak at frequency very close to π rad.

We already saw some indirect evidence in Figure 5.6 (which shows the 2D DFT of image blocks with air) – the spectral power for the Veridicom scanner is concentrated along the two main frequency axes, implying correlation within the scanner pattern along columns and rows, and particularly closely to π rad. There is no such power concentration in the 2D DFT for the UPEK scanner as it is nearly uniformly distributed in frequency (the sample means of the image blocks have been subtracted before computing the DFT, so there is no DC component).

Nevertheless, these initial studies led us to the conclusion that there is some correlation in the scanner patterns, which, however, we could not systematize for the purpose of using it. To quantify it relatively and integratively, we looked at the problem somewhat unconventionally.

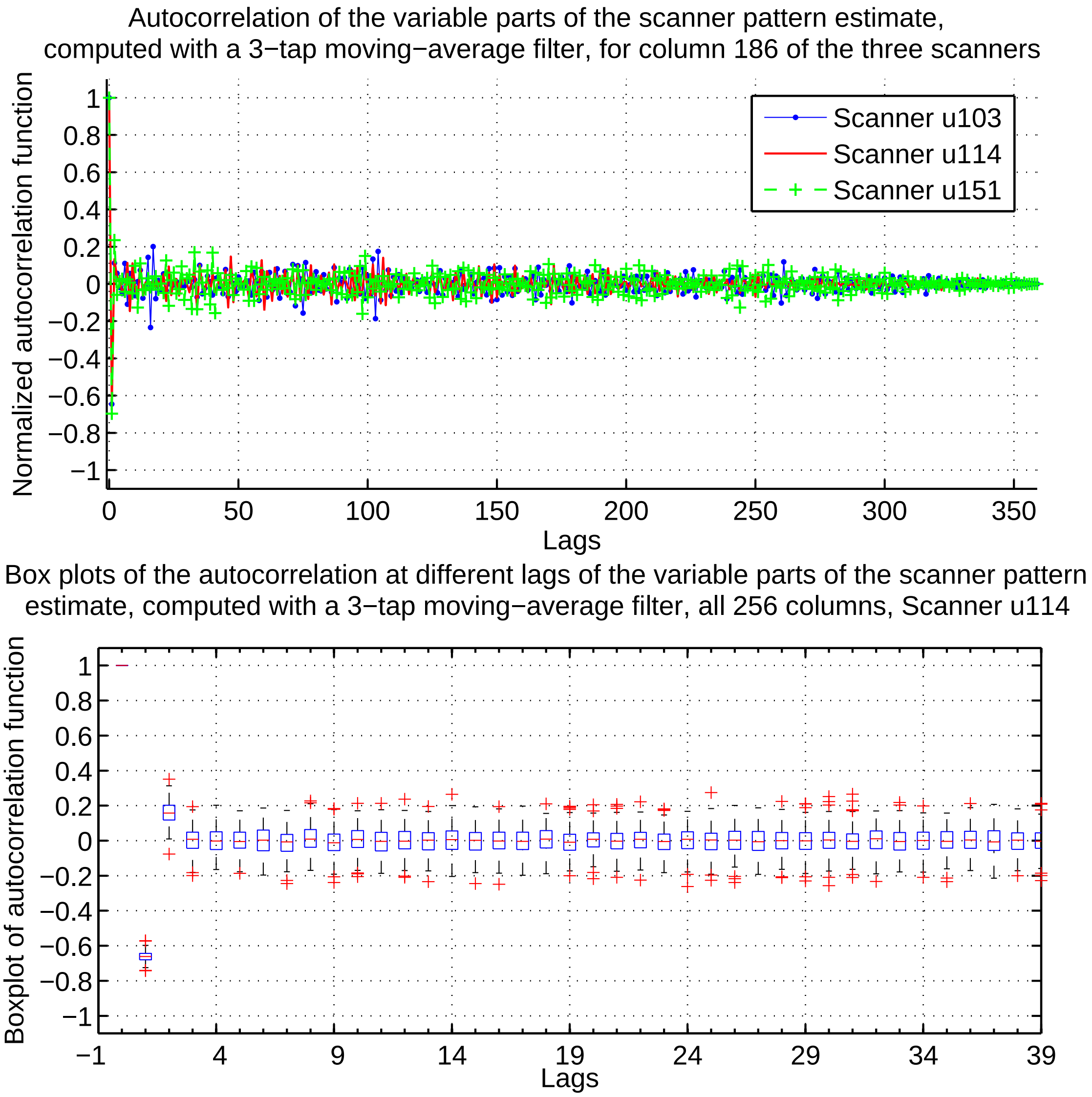
First, computing directly the autocorrelation (or autocovariance) function of a column (or a row) gives an extremely inaccurate estimate of the actual signal dependence because of the variable mean, especially when this variable mean is due to gradient effects, and is therefore not useful. To compute the variable part of the scanner pattern, in the preceding discussions we used a moving-average filter. Any kind of filtering, however, introduces correlation artifacts because the impulse response of the filter is convolved with the signal, which process essentially self-correlates the signal on a local basis, i.e., within the span of the filter impulse response. For this reason, we chose the simplest possible filter – a 3-tap (noncausal, symmetric) moving-average filter – and then subtracted the average from the signal. Applying this filter to a signal with pixel values even perfectly independent from each other introduces expected dependence at lag 1 as the local mean, which is subtracted from the current pixel, depends on the two pixels adjacent to the current pixel. Therefore, when using such a filter to compute the variable part of the scanner pattern, the autocorrelation estimate at lag 1 will be inherently inaccurate and cannot be relied upon for making conclusions. (Note: we show only positive lags because the signals are real valued and hence the autocorrelation function is real and symmetric. In addition, since the sample mean of the variable part of the scanner pattern estimate is very close to 0, the autocorrelation function is essentially the autocovariance function.)

The upper plot of Figure 5.56 shows the autocorrelation functions (normalized to 1 at lag 0) of the variable part of the scanner pattern, computed with a 3-tap moving average filter, of one column (186), estimated from 100 images with air, for three UPEK scanners. Although quantifying the level of correlation in a single metric is difficult, it is visible that the correlation for all lags, except for the first 2 lags (better visible on the lower plot), is small and contained in magnitude to 0.2, and for most lags, at most 0.1 in magnitude. The large values at lag 1 have already been explained as an artifact of the moving-average filtering. Unexpected is only the consistently somewhat large values at lag 2, which does suggest the presence of correlation. The autocorrelation functions along rows have similar behavior, with the difference being the larger values at lag 1.

The lower plot on the same figure shows the box plots of the normalized autocorrelation functions for all 256 columns of the scanner pattern estimate for Scanner u114, depicted only for the first 39 lags. The interquartile range (IQR), or the difference between the upper and the lower quartiles, which is 50% of the values, is the boxes in blue; the median (the second quartile) is in red. The whiskers of the box plots are at 1.5 IQR outside the lower and the upper quartiles, which in case of Gaussian distribution cover $\pm 2.7\sigma$ (or 99.3% of the values). For lags other than lag 1 and 2, we can make three observations:

- The median of the normalized autocorrelation function at each lag is very close to 0. The medians at the subsequent lags also appear to be random, not changing according to a pattern;
- 50% of the normalized autocorrelation at each lag is contained within ± 0.06, which corresponds to a range of 0.12, which is very small;
- Except for the few outliers, the range covered by the whiskers is within ± 0.2, which corresponds to a range of 0.4.

All this suggests that while the scanner pattern (along columns) for the UPEK scanners may exhibit some correlation, this correlation is limited and justifies our assumption for being largely uncorrelated.


Autocorrelation of the variable parts of the scanner pattern estimate, computed with a 3−tap moving−average filter, for column 186 of the three scanners

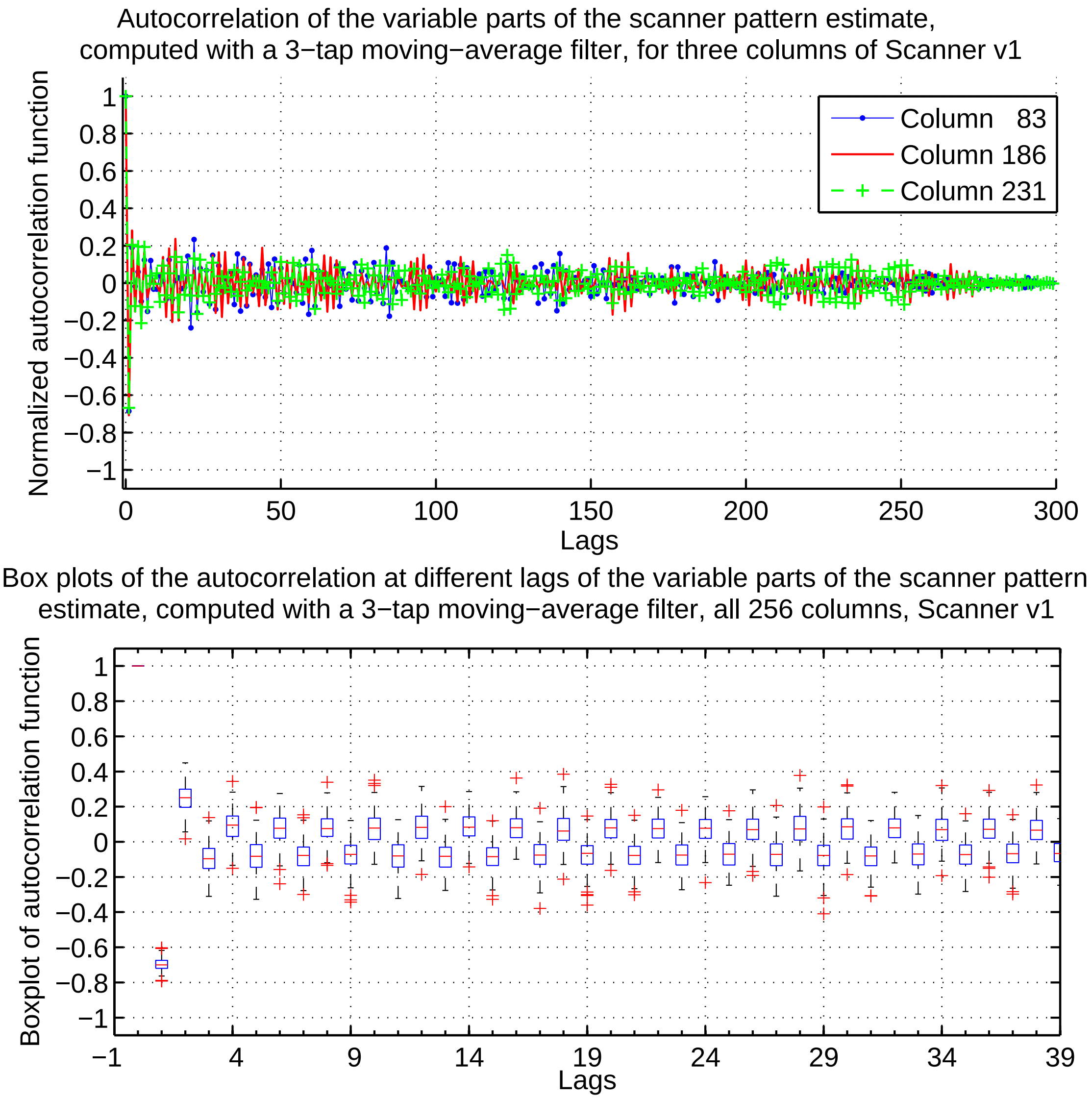
Box plots of the autocorrelation at different lags of the variable parts of the scanner pattern estimate, computed with a 3−tap moving−average filter, all 256 columns, Scanner u114

Figure 5.56: Autocorrelation function of the variable part of the scanner pattern estimate along columns for three UPEK scanners

We did the same study for the Veridicom scanner (v1). The upper plot of Figure 5.57 shows the autocorrelation functions (also normalized to 1 at lag 0) of the variable part of the scanner pattern, computed with a 3-tap moving average filter, of three columns (83, 186, and 231), estimated from 100 images with air, for

Scanner v1. Although here the normalized autocorrelation is contained in magnitude to 0.3, it is markedly larger than 0.1 at most lags. The lower plot of the same figure shows the autocorrelation function for all 256 columns for the first 39 lags. While the IQR and the whisker's coverage here are practically the same as those of the UPEK scanners, the medians here consistently oscillate between about -0.07 and 0.07, alternating between adjacent pixels, which corroborates the observation made from the PSD studies of a peak close to π rad. The autocorrelation function along rows is very similar to the one along columns.

In summary, after studying the scanner pattern, directly by computing autocorrelation functions and indirectly, we believe that the random field of the variable part of the scanner pattern can be modeled as white, i.e., either its one-dimensional or its two-dimensional autocorrelation function can be well approximated by a Dirac delta function, one-dimensional or two-dimensional, respectively. Nevertheless, the fact that the autocorrelation functions are not perfect Dirac delta functions needs to be kept in mind, as we did while designing the algorithms. Moreover, the accuracy of this model approximation depends on the particular signal model. The accuracy is also different along the two main axes of the two-dimensional autocorrelation function due to specifics in the hardware (and possibly software) implementation of the fingerprint scanner type and model in question, most probably to its sensing technology and addressing of its sensing elements (see Section 5.3). In particular, for the UPEK scanners (Signal Model A), the autocorrelation function along columns is typically closer to the Dirac delta function than the autocorrelation function along rows is. For the Veridicom scanner (Signal Model B), the autocorrelation function


Autocorrelation of the variable parts of the scanner pattern estimate, computed with a 3−tap moving−average filter, for three columns of Scanner v1

Box plots of the autocorrelation at different lags of the variable parts of the scanner pattern estimate, computed with a 3−tap moving−average filter, all 256 columns, Scanner v1

Figure 5.57: Autocorrelation function of the variable part of the scanner pattern estimate along columns for the Veridicom scanner

along columns is as close to the Dirac delta function as it is along rows.

Amount of Information in the Scanner Pattern

Quantifying accurately the amount of information (in number of bits) that uniquely identifies each individual scanner and that is contained in the scanner pattern is for further study as it proved particularly difficult because it depends on many and diverse factors, including:

- the inherent correlation within the scanner pattern itself;
- the correlation incurred by the filtering (moving-average or Wiener) of the advanced algorithms;
- the effect of quantization;
- the scanner noise, which cannot be removed from the scanner pattern as it is estimated from a single image containing a fingerprint in it;
- the residual distortion from the fingerprint (discussed later) that remains in the scanner pattern estimate.

As for the correlation in the variable part $s_v(i, j)$ of the scanner pattern, we believe that estimating the joint PDF of the scanner pattern (in order to compute the entropy rate) will be particularly difficult and we will have to use simplified models. The amount of information also depends on the number of pixels for which the scanner pattern is estimated after masking (in the advanced algorithms), which number is random for the area scanners but fixed for the swipe scanners.

Nevertheless, as a very simple and idealized model, we used a Gaussian process in which the pixel values are uncorrelated (and therefore independent) and compute the (differential) entropy of the scanner pattern per pixel as:

$$h(s) = \frac{1}{2} log_2(2\pi e \sigma_s^2) \approx 2.047 + \frac{1}{2} log_2(\sigma_s^2), \quad \text{in bits} \tag{5.32}$$

The Gaussian model also gives the upper bound as its entropy is the largest among all random variables of equal variance. In Table 5.11 below we provide the minimum, the average, and the maximum number of bits per pixel according to this model. Assuming that at least 100 pixels are required to identify a scanner (this is the minimum number of pixels required to overlap so that a matching score can be reliably computed in the advanced algorithms for the area scanners), then the upper bound for 100 pixels in this ideal case is roughly 350 to 400 bits. For the area scanners, however, the number of pixels used can be much larger, possibly tens of thousands, potentially giving at least one order of magnitude more bits. The only real limitation is for the UPEK swipe scanners (and for any swipe scanner) as the number of pixels there is very limited, only about 150 to 200, and thus the entropy is limited to about 550 to 800 bits, again in the ideal case.

	Parameter	**Min**	**Average**	**Max**
Signal Model A	variance σ_s^2	12.459	15.423	19.317
	bits/pixel	3.87	**4.02**	4.18
Signal Model B	variance σ_s^2	8.501	8.645	8.763
	bits/pixel	3.59	**3.60**	3.61

Table 5.11: Differential entropy of the scanner pattern per pixel in the ideal case

Because of the correlation within the scanner pattern (of each individual scanner), the pixel values are not independent and therefore the effective number of unique bits will be smaller. For example, if the correlation affects 2 adjacent pixels, as our observation for the UPEK areas scanners generally is, the number of unique bits very roughly can be assumed to be twice smaller, which is a correction factor only to compensate for the correlation.

A fundamental shortcoming of this approach for computing the number of bits is that what matters for the randomness is not the entropy of the scanner pattern *per se*, but the degrees of freedom in the scanner pattern across the whole population of scanners. This will require estimating the scanner pattern of many scanners (in the order of one hundred, at least) and study the distribution of (the variable part of) the scanner pattern for each individual pixel across the scanner population. Then, again a correlation coefficient or a another, different but appropriate similarity metric can be used to compute the degrees of freedom of the scanner pattern. We leave this research for further study because, besides the significant effort, most probably it will also require details about the specific application which will use these bits and exploit their randomness.

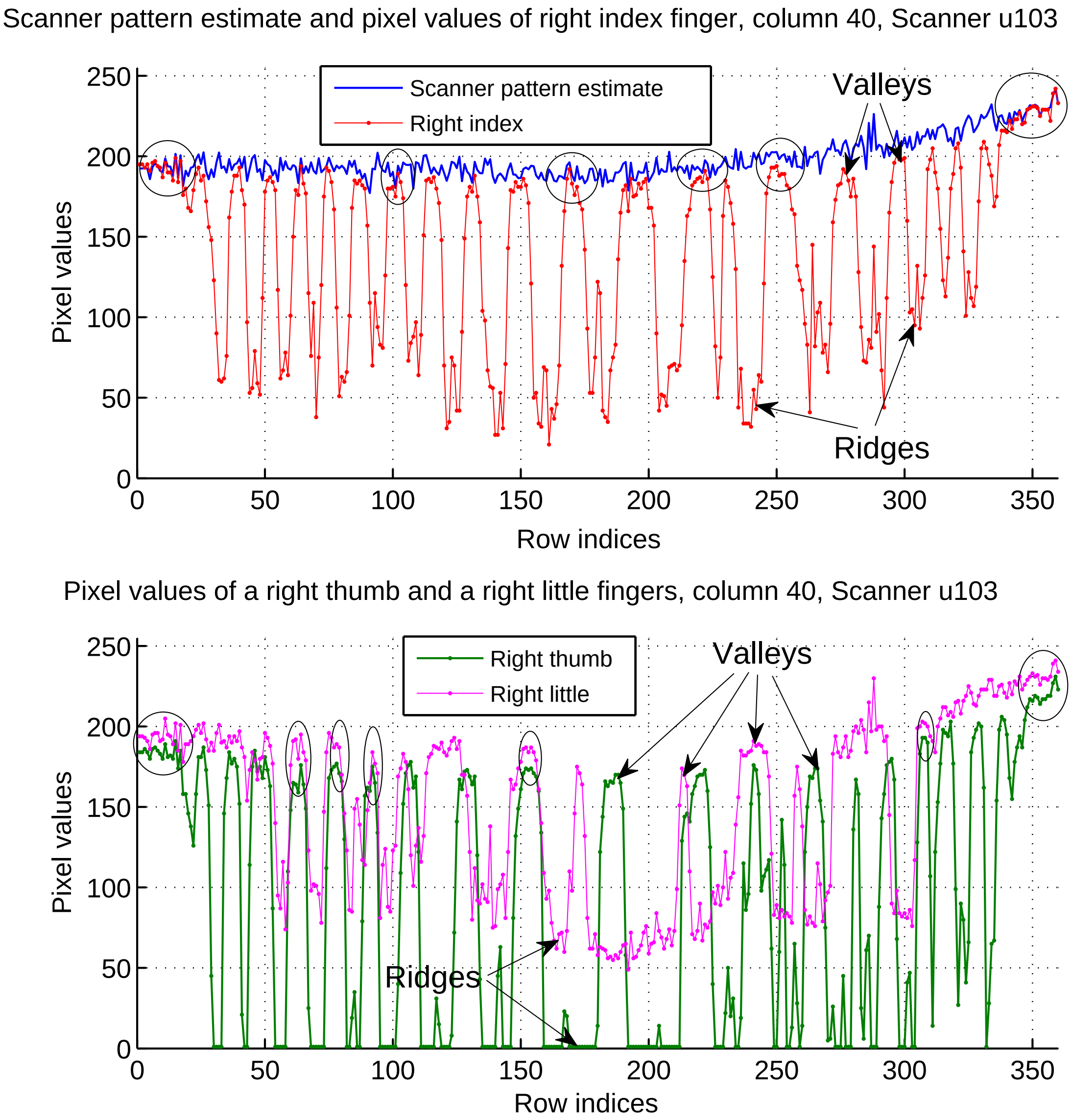
5.4.4 Fingerprint Pattern

In the discussion that follows, we will use the term *fingerprint pattern* to refer to the two-dimensional function $f(i, j)$ as defined by Expression 5.9 (for Signal Model A) and in Expression 5.12 (for Signal Model B).

The surface of the fingertip skin (as well as its subsurface) is a sequence of ridges and valleys. This surface is read by the fingerprint scanner and represented as a two-dimensional signal. Along with other imperfections introduced by the fingerprint scanner in this representation, the acquisition process may also include nonlinear transformations, such as (a) a projection of the three-dimensional fingertip onto the two-dimensional scanner platen and (b) a sensing process that reads the ridges and valleys and converts these readings into electrical signals, which signals are further processed and converted into a digital image. As result of such nonlinear transformations, the fingerprint pattern may become a nonlinear function of the actual surface (and/or subsurface) of the fingertip skin.

For our purposes, the fingerprint pattern $f(i,j)$, in each of the its two dimensions, can be roughly viewed as one dominant, single-frequency oscillation along its harmonics. The frequency of this oscillation depends on the width of the ridges and valleys, which are specific for each individual. This frequency also depends on the particular type of finger – it is known that thumbs typically have much wider ridges and deeper valleys than little fingers do (of the hands of one and the same person). Also typically, index fingers have narrower ridges and valleys than thumbs, and wider than little fingers. This frequency also depends on the gender (male fingers typically have wider ridges and valleys than female fingers) and on the age (adults usually have wider ridges and valleys than children do). Finally, this frequency may even vary within one and the same fingerprint.

The upper plot of Figure 5.58 shows the scanner pattern estimate and the pixel values (in grayscale levels) of an image containing a fingerprint (of an index finger) for one and the same column and one and the same UPEK scanner. The regions corresponding to valleys and to ridges are also shown. As already explained, because of the capacitive sensing, the ridges have smaller pixel values than the valleys have. The circles on the plot are over the regions where the pixel values of the image are close to the scanner pattern estimate. These regions are either of valleys or where the fingertip is not present at all, i.e., near the edges of the scanner platen (for small and high row indices). In this particular plot, the pixel values of the image with fingerprint is almost the same as the estimate of the scanner pattern, and therefore, not only are the variable parts of the scanner pattern the same, but also the absolute values of the pixels are.


Scanner pattern estimate and pixel values of right index finger, column 40, Scanner u103

Pixel values of a right thumb and a right little fingers, column 40, Scanner u103

Figure 5.58: Scanner pattern estimate and pixel values of 3 fingers, Scanner u103

The lower plot of Figure 5.58 shows the pixel values of one image containing a thumb finger and one image containing a little finger, both for the same column and scanner as in the upper plot. Here, however, the absolute levels of the two signals differ, yet their variable parts are very similar to each other in certain regions – again near the edges (for small and high row indices) and in the overlapping regions with valleys (both marked with circles). One very important observation can be made here: in this particular example, 31% of the pixel values of the column from the image with a thumb are constant and equal to 1 (the images never contain 0 as a pixel value), i.e., the scanner becomes saturated and the signal "clips." In contrast, for the index and little finger, all pixel values are well above 20.

We observed that this saturation ("clipping") can be due to several reasons: (a) strong pressure of the fingertip to the scanner platen (typical for thumb fingers as they are stronger), (b) wide ridges (also typical for thumbs and to some extent for the middle fingers as they are naturally larger than the other fingers), and (c) highly moisturized fingers. While this saturation of the signal to 1 is not a problem for the fingerprint recognition (because the signal will eventually be binarized anyway), extracting the scanner pattern from such saturated regions is impossible because it is simply not present there. Therefore, it is very important to be able to locate and exclude such regions from further processing, as the advanced algorithms are designed to do.

We did not do a detailed study on the frequency the fingerprint pattern because it cannot be relied upon for our purposes. As visible from both plots of Figure 5.58, besides the smooth valleys (and ridges) and nice periodic segments, there are also many irregularities, in particular "spikes," which can be due to many and various factors. Nevertheless, as a very approximate model, we decided that a frequency of about 0.63 radians per pixel is sufficiently representative for modeling purposes in the context of our algorithm design.

The range of $f(i,j)$ is $(0, 1]$. The pixel values $g(i,j)$ for the two general types of regions of the fingertip skin, ridges and valleys, taking into account that $s(i,j) \gg 1$ for either signal model, are approximately as follows:

- In the regions with ridges, $f(i,j)$ is close to 1. Hence:
  - for Signal Model A: $g^{(r)}(i,j) \approx \frac{s(i,j)}{1+s(i,j)} + n(i,j,t) \approx 1 + n(i,j,t)$
  - for Signal Model B: $g^{(r)}(i,j) \approx \frac{s(i,j)}{1+1} + n(i,j,t) \approx \frac{s(i,j)}{2} + n(i,j,t)$
- In the regions with valleys, $f(i,j)$ is close to 0. Hence:
  - for Signal Model A: $g^{(v)}(i,j) \approx \frac{s(i,j)}{1+0} + n(i,j,t) \approx s(i,j) + n(i,j,t)$
  - for Signal Model B: $g^{(v)}(i,j) \approx \frac{s(i,j)}{1+0} + n(i,j,t) \approx s(i,j) + n(i,j,t)$

Therefore, in the regions with valleys, for either signal model we have (see Expression 5.15):

$$g^{(v)}(i,j) \approx s(i,j) + n(i,j,t) = g^{(po)}(i,j) \tag{5.33}$$

And this is essentially what Figure 5.58 illustrates. Identifying the regions where this approximation holds sufficiently accurately is the basis of our approach.

5.5 Advanced Algorithms

In this section we present the signal processing steps, henceforth referred to as *modules*, and their modes of operation; the set of all modules and their modes we collectively call *advanced algorithms*. We developed the advanced algorithms based on our understanding of the image acquisition process, summarized in the signal models we created for it, and the initial work on the three scanner authentication scenarios. The advanced algorithms actually refer to two groups of algorithms: one group for the area scanners of UPEK/Authentec and Veridicom/Fujitsu and another one for the swipe scanners of UPEK/Authentec. Two important elements of the algorithms are the signal inversion and the Wiener filter, both discussed next.

Signal Inversion

Expressions 5.9 and 5.12 model the relationship between the scanner pattern $s(i,j)$ and the fingerprint pattern $f(i,j)$. Because of the division and the multiplication operations in them, directly separating $s(i,j)$ from $f(i,j)$ is difficult. To simplify this complex relationship, we propose using inversion. The pixel values $g(i,j)$ thus get inverted and we define a new signal $h(i,j)$ as:

$$h(i,j) \triangleq \begin{cases} \frac{1}{g(i,j)} & \text{for } g(i,j) \neq 0 \\ 1 & \text{for } g(i,j) = 0 \end{cases} \tag{5.34}$$

This inversion applied to Signal Model A transforms the relationship between the scanner pattern and the fingerprint pattern differently than when applied to Signal Model B, but the final result of the inversion for the regions with valleys is very similar for the two signal models because (a) the scanner noise $n(i,j,t)$ is much weaker than the scanner pattern $s(i,j)$, which makes the effect of the scanner noise negligible, and (b) the value of the scanner pattern $f(i,j)$ in the regions with valleys is close to 0. Thus, for Signal Model A, we have:

$$h(i,j) = \frac{1}{\frac{s(i,j)}{1+s(i,j)f(i,j)} + n(i,j,t)} \approx \frac{1}{s(i,j)} + f(i,j) \tag{5.35}$$

and for Signal Model B:

$$h(i,j) = \frac{1}{\frac{s(i,j)}{1+f(i,j)} + n(i,j,t)} \approx \frac{1}{s(i,j)} + \frac{f(i,j)}{s(i,j)} \tag{5.36}$$

Since the mean μ_s of the scanner pattern $s(i,j)$ is much larger (at least one order of magnitude) than its standard deviation σ_s, the variations of the scanner pattern $s(i,j)$ are small in comparison with its mean μ_s. Therefore, in Expression 5.36 $\frac{f(i,j)}{s(i,j)} \approx \frac{f(i,j)}{\mu_s}$. This means that the fingerprint pattern $f(i,j)$ is essentially simply scaled down by a constant factor μ_s, but its waveform shape as such is preserved; we refer to this scaled down version of the fingerprint pattern as $f'(i,j)$. Hence, using this approximation, Expression 5.36 becomes similar to Expression 5.35 and is:

$$h(i,j) \approx \frac{1}{s(i,j)} + \frac{f(i,j)}{\mu_s} = \frac{1}{s(i,j)} + f'(i,j) \tag{5.37}$$

Moreover, since in the regions with valleys, $1 \gg f(i,j)$, then $\frac{1}{s(i,j)} \gg \frac{f(i,j)}{s(i,j)}$, essentially making the second term in Expression 5.36 negligible, further simplifying to $h(i,j) \approx \frac{1}{s(i,j)}$ in the regions with valleys.

Because of its importance for the analysis that follows, we also define the signal $t(i,j)$ as the inverse of the scanner pattern:

$$t(i,j) \triangleq \frac{1}{s(i,j)} \tag{5.38}$$

Therefore, by applying the signal inversion of Expression 5.34, the multiplicative relationship between $s(i, j)$ and $f(i, j)$ in Expressions 5.35 and 5.36 becomes transformed into the sum of two terms, one of which represents the scanner pattern and the other one – the fingerprint pattern, thus making their separation possible using simple signal processing:

$$h(i, j) \approx t(i, j) + f(i, j) \tag{5.39}$$

As a next step, we developed a Gaussian approximation for the inverse of a Gaussian random variable (derived in detail in the appendix, Section 6.5), according to which $t(i, j)$ has approximately a Gaussian distribution $N(\mu_t, \sigma_t^2)$ with:

$$\mu_t = \frac{1}{\mu_s} \qquad \text{and} \qquad \sigma_t^2 = \frac{\sigma_s^2}{\mu_s^4} \tag{5.40}$$

This approximation is sufficiently accurate when $\mu_s > 100$ and $\mu_s \gg \sigma_s$, both of which hold true for either signal model. Note: because of the inevitable presence of (spatial) scanner noise, which we have neglected in the current discussion about the inversion, of importance for the signal processing of the pixels in the valleys is not the variance σ_s^2 of the scanner pattern alone, but the variance of the combined scanner pattern and spatial scanner noise. Since the scanner pattern and the spatial scanner noise can both be assumed Gaussian, their sum is also Gaussian (verified for the real signals earlier), and therefore, the inversion approximation is still applicable, but has to be used with the sum of their variances, i.e., $\sigma_s^2 + \sigma_n^2$.

In summary, the problem of separating the scanner pattern and the fingerprint pattern, which are in complex relationship with each other, is reduced to separating a Gaussian signal from an additive and roughly sinusoidal signal, which can be done in a straightforward and computationally-efficient way. Two downsides of this inversion, however, are that: (a) the inversion may require care when implementing it in digital precision-limited systems, e.g., with fixed-point arithmetic, because of possible roundoff errors, and (b) it may also create other types of nonlinear effects.

We also observed that even without the inversion, in the regions with valleys, i.e., where $f(i,j) \approx 0$, for both signal models (see Expression 5.33), the scanner pattern can also be relatively easily extracted because:

$$g(i,j) \approx s(i,j). \tag{5.41}$$

However, $f(i,j)$ is as close to 0 as to make the approximation in Expression 5.41 sufficiently accurate only for very small part of the pixels in an image containing a fingerprint. In this case, therefore, the subsequent signal processing modules may use only that part of the image for which the approximation in Expression 5.41 is sufficiently accurate or use a larger part of the image where the approximation is not. Our study showed that either approach degrades the overall performance of the algorithm but is still possible and can be used in certain cases.

Wiener Filter

Herewith we incorporate a summary of the theory of Wiener filters as presented in [Lim 1989] for the one-dimensional case. Let the signal $p(k)$ and the additive noise $q(k)$, where k is an integer, are two zero-mean second-order stationary discrete-time random processes, linearly independent of each other, and the noisy observation is $r(k) = p(k) + q(k)$. The objective is to find that linear time-invariant (or alternatively, space-invariant) filter with a possibly infinite and possibly non-causal impulse response $b(k)$ such that the linear estimate $\hat{p}(k)$ given the observation $r(k)$, i.e., $\hat{p}(k) = r(k) * b(k)$, is closest to the signal $p(k)$ in mean-square error sense: $E[|p\,(k) - \hat{p}\,(k)|^2]$. The discrete-time Fourier transform of the linear time-invariant filter $b(k)$ that minimizes the mean square error is:

$$B\,(\omega) = \frac{S_p\,(\omega)}{S_p\,(\omega) + S_q\,(\omega)}$$

where $S_p\,(\omega)$ and $S_q\,(\omega)$ are the power spectral densities of the signal $p\,(k)$ and the noise $q\,(k)$, respectively, and ω is the angular frequency. If $p\,(k)$ and $q\,(k)$ are Gaussian random processes, then the Wiener filter is also the optimal nonlinear mean-square error estimator.

In essence, the Wiener filter preserves the high SNR frequency components and suppresses the low SNR frequency components. Let $\rho\,(\omega) \triangleq \frac{S_p(\omega)}{S_q(\omega)}$ be the SNR in function of the frequency. Then the Wiener filter transfer function is: $B\,(\omega) = \frac{\rho(\omega)}{\rho(\omega)+1}$. At the frequencies where the signal is much stronger than the noise, i.e., where $\rho\,(\omega) >> 1$, the transfer function is $B\,(\omega) \approx 1$, and the observation $r\,(k)$ passes through the filter almost unchanged. On the other hand, the Wiener filter almost completely suppresses, i.e., $B\,(\omega) \approx 0$, the frequency components at which the signal is much weaker than the noise, i.e., where $\rho\,(\omega) \approx 0$. If the signal $p\,(k)$ has a nonzero mean μ_p and the noise $q\,(k)$ has a nonzero mean μ_q, then they have to be subtracted from the observation $r\,(k)$ before filtering it.

When the impulse response $b(k)$ of the Wiener filter changes in function of the local characteristics of the signal being processed, the filter becomes time variant (or alternatively, space variant). Therefore, instead of using constant (for all indices k)

power spectral densities for the signal and the noise, they can be estimated locally; furthermore, the expected values of the signal and the noise can be estimated locally as well. Depending on how these quantities are estimated, many variations are possible, but the simplest option is when the local power spectral densities of both the signal and the noise are assumed constant in function of the frequency, i.e., the signal and the noise are both "white." When the signal and the noise are zero mean, their power spectral densities are equal to their (local) variances: $S_p\left(\omega\right) = \sigma_p^2$ and $S_q\left(\omega\right) = \sigma_q^2$, where σ_p^2 and σ_q^2 are the variances of the signal and the noise, respectively. In this case, the frequency response of the Wiener filter is constant in function of the frequency, and thus its impulse response is a scaled Dirac delta function:

$$b(k) = \frac{\sigma_p^2}{\sigma_p^2 + \sigma_q^2}\,\delta\left(k\right)$$

where $\delta\left(k\right)$ is the Dirac delta function. Moreover, the filtering also depends on the relative relationship between the local variance of the signal and the noise: where the signal local variance σ_p^2 is smaller than the noise local variance σ_q^2, the filter suppresses the noise and thus the filter output is approximately equal to the local mean of the signal. On the other hand, where the signal local variance σ_p^2 is larger than the noise local variance σ_q^2, the filter leaves the input signal almost unchanged. Since the signal (local) variance is not known and generally is difficult to be estimated, in practice an estimate for the variance of the noisy observation $r(k)$ is used instead because $\sigma_r^2 = \sigma_p^2 + \sigma_q^2$ (since $p(k)$ and $q(k)$ are assumed to be independent). Putting all things together yields the following expression for the estimate $\hat{p}(k)$ of the signal $p\left(k\right)$:

$$\hat{p}(k) = \mu_p\left(k\right) + \frac{\max(0, \sigma_r^2\left(k\right) - \sigma_q^2)}{\max(\sigma_r^2\left(k\right), \sigma_q^2)}\left(p\left(k\right) - \mu_p\left(k\right)\right) \tag{5.42}$$

where $\sigma_r^2\left(k\right)$ is the local variance of the observation $r(k)$, and $\mu_p\left(k\right)$ is the local mean of the signal $p\left(k\right)$, which is also equal to the local mean $\mu_r\left(k\right)$ of the observation $r(k)$ since the noise $q(k)$ is zero mean. Assumed to be known is only the variance σ_q^2 of the noise; $\sigma_r^2\left(k\right)$ and $\mu_r\left(k\right)$ (and thus also $\mu_p\left(k\right)$) are estimated from the observation $r(k)$. The output of the adaptive Wiener filter is the estimate $\hat{p}(k)$, which is a smoothed version of the signal $p\left(k\right)$.

5.5.1 Advanced Algorithms for the Area Scanners

Figure 5.59 shows a conceptual diagram of the sequence of signal processing modules in which the signal **g**, the image, is processed to produce the signal d, the scanner verification decision, along with the interface signals among them. The signals between the subsequent modules in Figure 5.59 represent only the main, not all, input and output signals of the modules. Next we describe the processing of each module and their different modes of operation as disclosed in [Ivanov and Baras US'952].

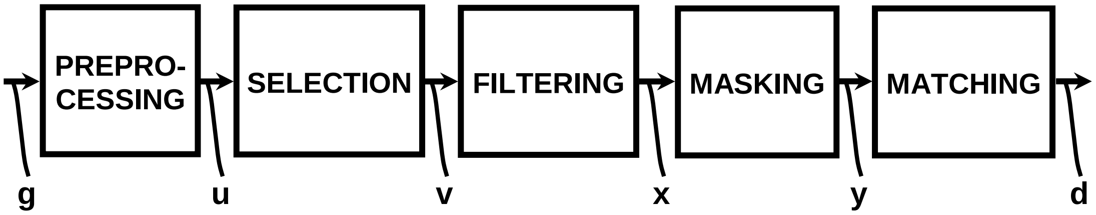


Figure 5.59: Conceptual diagram of operation of the modules for the area scanners

5.5.1.1 Preprocessing Module

The Preprocessing Module (see Figure 5.59) has the signal $g(i, j)$, i.e., the pixels of the image, as input and the signal **u**, a two-dimensional signal with the same size as **g**, as output. It has two modes of operation:

(a) Direct mode: $u(i, j) \;=\; g(i, j)$;

(b) Inverse mode: $u(i, j) \;=\; h(i, j) = \frac{1}{g(i,j)}$;

This mode implements the signal inversion discussed in the preceding section.

5.5.1.2 Selection Module

The Selection Module (see Figure 5.59) selects part of the pixels from its input signal $\mathbf{u}$ and produces this selection as output signal $\mathbf{v}$, which signal defines regions of interest. This selection is necessary because: (a) selecting part of the pixels from all pixels in the image in a particular order provides a signal with statistical characteristics that facilitates the subsequent signal processing and (b) using only some of all pixels minimizes the computations, therefore decreasing the required computational time of the whole algorithm or alternatively, relaxing the requirement for computational power.

One or many lines of pixels are selected from the two-dimensional input signal $\mathbf{u}$ to produce the one-dimensional signal output $\mathbf{v}$, and thus, all subsequent signal processing is one dimensional. The output signal $\mathbf{v}$ consists of b line segments, concatenated one after each other, with each line segment having N elements and each element being a pixel value. The selected line or lines can be columns of pixels, rows of pixels, or diagonal lines of pixels.

Columns of Pixels

Since the one-dimensional autocorrelation function of the variable part of the scanner pattern along columns can be assumed a Dirac delta function, then the pixel values when selected column-wise are close to a white sequence. This type of selection greatly facilitates the processing of the Filtering Module. The columns of pixels, as line segments, are concatenated in a sequence one after each other to form one line of pixels, which line becomes the output signal $\mathbf{v}$.

Let the input signal $\mathbf{u}$ be an image with N rows. If the first column of pixels to be included as the first line segment in the output signal $\mathbf{v}$ is column j of the input signal u, we denote this line segment as vector $\mathbf{v_1} = [u(1,j) \;\; u(2,j) \;\; \ldots \;\; u(N,j)]^T$. Next, if the second column of pixels to be included as the second line segment in the output signal $\mathbf{v}$ is column k of $\mathbf{u}$, we denote this line segment as vector $\mathbf{v_2} = [u(1,k) \;\; u(2,k) \;\; \ldots \;\; u(N,k)]^T$. The other vectors $\mathbf{v_3}, \; \mathbf{v_4}, \; \ldots \; \mathbf{v_c}$, where c is the number of columns to be concatenated, are formed in the same way, and their concatenation forms the output signal $\mathbf{v}$:

$$\mathbf{v} = \begin{bmatrix} \mathbf{v_1} \\ \mathbf{v_2} \\ \ldots \\ \mathbf{v_c} \end{bmatrix} \tag{5.43}$$

which is a column vector with $(c.N)$ elements. Each element of this vector $\mathbf{v}$ is a pixel value.

It is possible to select all columns in the image or only a few of them. In the latter case, it is recommended to have the columns selected nonadjacent and about evenly spaced across the whole image area because (a) this type of selection reduces the statistical dependence among the selected pixels and (b) using pixels from all regions of the image ensures processing a sufficient number of pixels with high quality estimates of the scanner pattern. For example, for Signal Model A, selecting about 10 nonadjacent columns, each column containing 360 pixels, may be sufficient. Thus, the output signal $\mathbf{v}$ may contain about 3,600 pixels or only about 4% of the total 92,160 pixels in an image with 360 rows and 256 columns, greatly reducing the number of computations and computational time. Furthermore, it is also possible that the columns contain only a subset of the rows of the image. For example, it is possible to exclude the first several and the last several rows. Thus, if using only rows from 25 through 245, the line segments for each selected column j are of the form: $[u(25,j)\ u(26,j)\ \ldots\ u(245,j)]^T$. The overall performance in this case may be higher because the pixels in the regions close to the boundaries of the image may experience edge effects that are difficult to mitigate.

Rows of Pixels

Since the one-dimensional autocorrelation function along rows may not be sufficiently close to a Dirac delta function, the pixels along rows exhibit non-negligible statistical dependence, and selecting and using them may degrade the overall performance. However, it is still possible to achieve good performance by selecting more pixels, i.e., a larger number of rows.

Similarly to the case with columns of pixels, the rows of pixels, as line segments, are concatenated in a sequence one after another to form one line of pixels, which line becomes the output signal $\mathbf{v}$. Thus, if row i of the input signal $\mathbf{u}$ becomes the first line segment, then $\mathbf{v_1} = [u(i,1)\ \ u(i,2)\ \ \ldots\ \ u(i,N)]^T$; if row k of the input signal $\mathbf{u}$ becomes the second line segment, then $\mathbf{v_2} = [u(k,1)\ \ u(k,2)\ \ldots\ u(k,N)]^T$; etc. When c number of rows are concatenated, the output signal $\mathbf{v}$ is a column vector with $(c.N)$ pixels:

$$\mathbf{v} = \begin{bmatrix} \mathbf{v_1} \\ \mathbf{v_2} \\ \ldots \\ \mathbf{v_c} \end{bmatrix} \tag{5.44}$$

Again, similarly to the case with columns of pixels, it is possible to select all rows in the image or only a few of them. In the latter case, it is recommended to have the rows selected nonadjacent and about evenly spaced across the whole image area. For example, for Signal Model A, we estimated that selecting about 20 rows (from the total of 300 rows) is sufficient to ensure good overall performance, meaning that only about 7% of all pixels of the image are processed, greatly reducing the number of computations and computational time. Furthermore, it is also possible that the rows contain only a subset of the columns of the image, e.g., the first several and the last several columns are excluded: if using only columns from 10 through 246, then the line segments for each selected row i are: $[u(i,10) \;\; u(i,11) \;\; \ldots \;\; u(i,246)]^T$.

Diagonal Lines of Pixels

Selecting diagonal lines from the input signal $\mathbf{u}$ is also possible. For example, the diagonal lines of pixels can be constructed by taking lines parallel to the main diagonal of the matrix that represents the input signal $\mathbf{u}$:

$$\mathbf{v_1} = \begin{bmatrix} u\,(4,1) \\ u\,(5,2) \\ u\,(6,3) \\ \cdots \\ u\,(259,256) \end{bmatrix}; \quad \mathbf{v_2} = \begin{bmatrix} u\,(14,1) \\ u\,(15,2) \\ u\,(16,3) \\ \cdots \\ u\,(269,256) \end{bmatrix}; \;\; \cdots \;\; \mathbf{v_{10}} = \begin{bmatrix} u\,(94,1) \\ u\,(95,2) \\ u\,(96,3) \\ \cdots \\ u\,(349,256) \end{bmatrix}.$$

The output signal $\mathbf{v}$ is then constructed by concatenating the column vectors $\mathbf{v_1}, \mathbf{v_2}$, $\ldots \mathbf{v_{10}}$.

Selecting pixels from the input signal $\mathbf{u}$ can be done in alternative ways as long as the selection is done so that: (a) these pixels include pixels from regions with valleys, (b) the fingerprint pattern contained in the resulting output signal $\mathbf{v}$ has a “smooth” waveform (e.g., no discontinuities), and (c) the statistical dependence in the scanner pattern contained in the resulting pixels is not significant.

Finally, the algorithm as specified above is with the Preprocessing Module coming before the Selection Module (see Figure 5.59) because this simplifies the exposition of the theory and principle of operation. However, it is trivial that they can be swapped, i.e., the Selection Module can precede the Preprocessing Module, because the processing in one of them is independent from the processing in the other one. When the two modules are swapped, the implementation is less computationally expensive because only (very small) part of all pixels in the image needs to be inverted.

5.5.1.3 Filtering Module

The Filtering Module (see Figure 5.59) has as input the signal $\mathbf{v}$ and as output the signal $\mathbf{x}$. It can be implemented in three ways: as a bypass implementation, with a low-pass filter, and with an adaptive Wiener filter.

In the bypass implementation, the input signal $\mathbf{v}$ is not modified and thus $\mathbf{x} = \mathbf{v}$. This implementation does not provide a very good overall performance, but it is the simplest one and can be used in certain very computational-limited systems.

In both the low-pass filter implementation and the adaptive Wiener filter implementation, the input signal $\mathbf{v}$ is processed to produce the output signal $\mathbf{x}$, which signal $\mathbf{x}$ contains the scanner pattern. Because of the signal inversion, this filtering can performed using simple signal processing and essentially comprises two operations: (1) a smoothing operation $\mathcal{F}(.)$ that smooths $\mathbf{v}$ and (2) a subtraction operation that subtracts thus smoothed signal $\mathcal{F}(\mathbf{v})$ from $\mathbf{v}$, and produces the output signal $\mathbf{x}$:

$$\mathbf{x} = \mathbf{v} - \mathcal{F}(\mathbf{v}) \tag{5.45}$$

In this way, the smoothing also removes the (variable) mean of the scanner pattern and yields only the variable part of it, from which another signal is derived and used in the Matching Module.

Padding and Windowing

The input signal $\mathbf{v}$, which is the output of the Selection Module, is a column vector consisting of the successive concatenations of c column vectors, with each vector representing different columns, rows, or diagonal lines and each vector having N pixels (see Expressions 5.43 and 5.44). Because of this construction, the signals in two adjacent vectors in v may be substantially different, creating a signal discontinuity at the place of the concatenation of the two vectors, which, in turn, may create unwanted artifacts. We propose three methods for mitigating these artifacts: computation shortening, replica padding, and constant padding, although using other methods is also possible. The replica padding method and constant padding method are specified below, while the computation shortening method is specific for the particular implementation of the Filtering Module and specified in the appropriate sections.

1. **Replica padding**

   Each vector $\mathbf{v_i}$, where i is an integer from 1 to c, is extended to include zero and negative indices and indices larger than N such that the added elements are symmetric about the first and the last indices of $\mathbf{v_i}$:

$$v_i\left(j\right) = v_i\left(2 - j\right) \qquad \text{for j from } \left(-\left\lfloor\frac{M}{2}\right\rfloor\right) \text{ to } 0 \tag{5.46}$$

$$v_i\left(j\right) = v_i\left(2N - j\right) \qquad \text{for j from } \left(N+1\right) \text{ to } \left(N + \left\lfloor\frac{M-1}{2}\right\rfloor\right) \tag{5.47}$$

The added elements in this extension can also be copies of the first and last elements, respectively, of the vector $\mathbf{v_i}$ in the same order as they appear in $\mathbf{v_i}$.

2. **Constant padding**

   Each vector $\mathbf{v_i}$, where i is an integer from 1 to c, is extended to include zero and negative indices and indices larger than N such that the added elements are set to constants. The constants can be the first or the last elements, respectively, as follows:

$$v_i\left(j\right) = v_i\left(1\right) \qquad \text{for } j \text{ from } \left(-\left\lfloor\frac{M}{2}\right\rfloor\right) \text{ to } 0 \tag{5.48}$$

$$v_i\left(j\right) = v_i\left(N\right) \qquad \text{for } j \text{ from } \left(N+1\right) \text{ to } \left(N + \left\lfloor\frac{M-1}{2}\right\rfloor\right) \tag{5.49}$$

   The constants can also be other numbers, such as 0 (the smallest minimum grayscale level), 127 (the middle grayscale level), 255 (the largest grayscale level), or any number in the range from 0 to 255. However, selecting constants that do not depend on the elements of the vector $\mathbf{v_i}$ may lead to degraded overall performance.

Thus, each vector $\mathbf{v_i}$ from the input signal $\mathbf{v}$ is processed separately, and the vectors that are the result of this processing are concatenated one after each other to form the output signal $\mathbf{x}$.

Incorporating such methods to avoid using pixels from one vector for signal processing in the vector adjacent to it may seem unjustified, but actually it may be quite important. Because applying a fingertip tightly in the regions around the boundaries of the scanner platen area (and of the image in this respect) is difficult, the pixels in these regions typically contain no fingerprint pattern. Hence, the estimate of the scanner pattern in these regions can be made very accurate if introduction of unwanted artifacts is avoided as specified above.

Another important aspect of the processing in this module is applying a windowing function. By multiplying the pixel values by a windowing function (for example, see $w(j)$ in Expression 5.51), the pixels close to the current index of the signal being processed have higher weight in the computation, thus controlling the level of the smoothing by placing larger weight on the pixels around the center pixel than on the distant pixels and thus reducing the effect of the latter.

The windowing function $w(j)$ of size M, for j an integer from $-\left\lfloor \frac{M}{2} \right\rfloor$ to $\left\lfloor \frac{M-1}{2} \right\rfloor$, can be a rectangular, triangular, Hann, Hamming, or a Gaussian window:

- A rectangular window (also known as Dirichlet window): $w(j) = 1$

- A triangular window (also known as Bartlett window): $w(j) = 2\left(1 - \frac{2}{M}|j|\right)$

- A Hann window (also known as Hanning or raised-cosine window): $w(j) = 1 + \cos\left(\frac{2j\pi}{M}\right)$

- A Hamming window: $w(j) = 2\left(0.54 + 0.46 \cdot \cos\left(\frac{2j\pi}{M}\right)\right)$

- A Gaussian window: $w(j) = 2 \cdot \exp\left(-\frac{1}{2}\left(\frac{j}{\frac{w_0(M-1)}{2}}\right)^2\right)$, where w_0 is a suitably chosen value below 0.5.

Using other windowing functions is also possible. The windowing function of choice has to satisfy is the normalization condition:

$$\sum w(j) \approx 1 \qquad \text{for all } j \text{ for which } w(j) \text{ is used.} \tag{5.50}$$

Low-pass Filter Implementation of the Filtering Module

The smoothing operation $\mathcal{F}(.)$ in this implementation is performed by a low-pass filter whose cutoff frequency, order, and attenuation in the different frequency bands are optimized for best performance. This low-pass filter can be a Butterworth, Chebyshev, elliptic, Bessel or another suitable filter. The filter may have a finite (FIR) or infinite (IIR) impulse response.

The low-pass filter we selected and describe here is the (possibly windowed) moving-average filter because of its extreme implementation simplicity and the corresponding excellent overall performance. Let the vectors $\mathbf{v_i}$, where i is an integer from 1 to c, are the vectors from the input signal $\mathbf{v}$ (see Expression 5.43). Generally, for a pixel with index k sufficiently far from the beginning and end of this vector v_i, i.e., such that the index $(k+j)$ does not address elements outside vector $\mathbf{v_i}$, the local mean $\mathbf{v_i^{(lm)}}$ is computed by:

$$v_i^{(lm)}(k) = \frac{1}{M} \sum_{j=-\left\lfloor \frac{M}{2} \right\rfloor}^{\left\lfloor \frac{M-1}{2} \right\rfloor} w(j) \, .v_i(k+j) \tag{5.51}$$

where M is a positive integer and determines the size of the moving-average window, $\mathbf{w}$ is a windowing function, and $\lfloor . \rfloor$ is the floor function. Preferably, M is selected to be odd so that the window is symmetric about index k, but selecting M to be even is also possible. Selecting M about 5 gives optimal results, but selecting values in the range from about 3 to about 7 is also possible. Using large values for M leads to better smoothing as a large number of pixels are taken in the sum, which, however, also makes the estimate of the scanner pattern in the neighborhood of transition regions between valleys and ridges less accurate. Using small values for M leads to worse smoothing as small number of pixels are taken in the sum (see Expression 5.51), which results in a less accurate estimate of the scanner pattern in the regions with valleys. Once the windowing function is selected, the size M of the moving-average window may need to be adjusted for achieving optimal overall performance.

For the pixels that are close to the beginning or the end of the vector $\mathbf{v_i}$, we propose three techniques for computing the local mean $\mathbf{v_i^{(lm)}}$, although using other techniques is also possible:

1. **Computation shortening**

   The sum in Expression 5.51 and the denominator in the coefficient in front of it are adjusted so that only elements of the vector $\mathbf{v_i}$ are used. Thus, for index $k \leq \left\lfloor \frac{M}{2} \right\rfloor$ or $k \geq \left(N - \left\lfloor \frac{M-1}{2} \right\rfloor + 1\right)$, the local mean vector $\mathbf{v_i^{(lm)}}$ is computed by:

$$v_i^{(lm)}(k) = \frac{1}{(j_{\max} - j_{\min} + 1)} \sum_{j=j_{\min}}^{j_{\max}} w_k(j) \, .v_i(k+j) \qquad (5.52)$$

   where $j_{\min} = \max\left(-\left\lfloor \frac{M}{2} \right\rfloor, 1-k\right)$ and $j_{\max} = \min\left(\left\lfloor \frac{M-1}{2} \right\rfloor N - k\right)$.

   In this case, the windowing function depends on the index k because the window is truncated and needs to be normalized such that the sum of its elements equals 1. Therefore, for all values of j from j_{min} to j_{max} as defined by Expression 5.52:

$$w_k\left(j\right) = \frac{w\left(j\right)}{D} \qquad \text{where} \;\; D = \sum_{j=j_{\text{min}}}^{j_{\text{max}}} w\left(j\right) \tag{5.53}$$

and $\mathbf{w}$ is the chosen windowing function. The computation shortening may lead to slight degradation in the accuracy of the local mean estimate for the pixels where it is applied to, but the distortion the computation shortening introduces is the smallest in comparison with the other techniques.

2. **Replica padding**

   Each vector $\mathbf{v_i}$ is extended to include zero and negative indices and indices larger than N as specified in Expressions 5.47. The added elements in this extension can also be copies of the first and last elements, respectively, of the vector $\mathbf{v_i}$ in the same order as they appear in $\mathbf{v_i}$. Then the local mean vector $\mathbf{v_i^{(lm)}}$ is computed using Expression 5.51.

3. **Constant padding**

   Each vector $\mathbf{v_i}$ is extended to include zero and negative indices and indices larger than N as specified in Expressions 5.49. Then the local mean vector $\mathbf{v_i^{(lm)}}$ is computed using Expression 5.51.

Once the local mean vectors $\mathbf{v_i^{(lm)}}$, where i is an integer from 1 to c, are computed, they are concatenated one after each other to form the local mean signal

$\mathbf{v^{(lm)}}$ (which is the smoothed signal $\mathcal{F}(\mathbf{v})$):

$$\mathbf{v^{(lm)}} = \begin{bmatrix} \mathbf{v_1^{(lm)}} \\ \mathbf{v_2^{(lm)}} \\ \dots \\ \mathbf{v_c^{(lm)}} \end{bmatrix}. \tag{5.54}$$

Finally, the output signal $\mathbf{x}$ of the Filtering Module in this implementation is the difference between the input signal $\mathbf{v}$ and the local mean signal $\mathbf{v^{(lm)}}$:

$$x(k) = v(k) - v^{(lm)}(k), \tag{5.55}$$

where k is the current pixel index, an integer from 1 to $(c.N)$.

Figures 5.60 and 5.61 in the Magnitude Masking Implementation for the Low-pass Filter (Section 5.5.1.4) show examples for signals processed with this implementation in both direct and inverse signal modes.

Adaptive Wiener Filter Implementation of the Filtering Module

Let the vectors $\mathbf{v_i}$, where i is an integer from 1 to c, be the vectors from the input signal $\mathbf{v}$ (see Expressions 5.43 and 5.44). Each vector $\mathbf{v_i}$ is processed separately in the following five steps:

I. **Computing the local mean**

The local means $\mathbf{v_i^{(lm)}}$ of the vectors $\mathbf{v_i}$ are computed exactly in the same way as the local means $\mathbf{v_i^{(lm)}}$ described in the Low-pass Filter Implementation in Section 5.5.1.3 and repeating that is not necessary. The only difference here is that for optimal results, the size of the moving-average window M should be about 3, although values in the range from about 2 to about 7 also yield good results. Here again, once the windowing function is selected, M may need to be adjusted for achieving optimal overall performance.

II. **Computing the local square**

The local squares $\mathbf{v_i^{(ls)}}$ of the vectors $\mathbf{v_i}$ are computed similarly to the local means $\mathbf{v_i^{(lm)}}$ above, with the difference being that $v_i(k+j)$ in the sums are replaced by $v_i^2(k+j)$. Nevertheless, for the sake of clarity, here below we repeat the specification.

Generally, for a pixel with index k sufficiently far from the beginning and end of this vector v_i, i.e., such that the index $(k+j)$ does not address elements outside vector $\mathbf{v_i}$, the local square $\mathbf{v_i^{(ls)}}$ is computed by:

$$v_i^{(ls)}(k) = \frac{1}{M} \sum_{j=-\left\lfloor \frac{M}{2} \right\rfloor}^{\left\lfloor \frac{M-1}{2} \right\rfloor} w(j) \,.v_i^2(k+j) \tag{5.56}$$

where M is a positive integer and determines the size of the moving-average window, $\mathbf{w}$ is a windowing function, and $\lfloor . \rfloor$ is the floor function. Preferably, M is selected to be odd so that the window is symmetric about index k, but selecting M to be even is also possible. Selecting M about 3 gives optimal results, but selecting values in the range from about 2 to about 7 is also possible. Once the windowing function is selected, the size M of the moving-average window may need to be adjusted for achieving optimal overall performance.

For the pixels that are close to the beginning or the end of the vector $\mathbf{v_i}$, we propose three techniques for computing the local square $\mathbf{v_i^{(ls)}}$, although using other techniques is also possible:

1. **Computation shortening**

   The sum in Expression 5.56 and the denominator in the coefficient in front of it are adjusted so that only elements of the vector $\mathbf{v_i}$ are used. Thus, for index $k \leq \left\lfloor \frac{M}{2} \right\rfloor$ or $k \geq \left(N - \left\lfloor \frac{M-1}{2} \right\rfloor + 1\right)$, the local square vector $\mathbf{v_i^{(ls)}}$ is computed by:

$$v_i^{(ls)}(k) = \frac{1}{(j_{\max} - j_{\min} + 1)} \sum_{j=j_{\min}}^{j_{\max}} w_k(j) \, .v_i^2(k+j) \qquad (5.57)$$

   where $j_{\min} = \max\left(-\left\lfloor \frac{M}{2} \right\rfloor, 1-k\right)$ and $j_{\max} = \min\left(\left\lfloor \frac{M-1}{2} \right\rfloor N-k\right)$.

   In this case, the windowing function depends on the index k because the window is truncated and needs to be normalized such that the sum of its elements equals 1. Therefore, for all values of j from $j_{\min}$ to $j_{\max}$ as defined by Expression 5.57:

$$w_k(j) = \frac{w(j)}{D} \qquad \text{where} \quad D = \sum_{j=j_{\min}}^{j_{\max}} w(j) \qquad (5.58)$$

   and $\mathbf{w}$ is the chosen windowing function. The computation shortening may lead to slight degradation in the accuracy of the local square estimate for the pixels where it is applied to, but the distortion the computation shortening introduces is the smallest in comparison with the other techniques.

2. **Replica padding**

   Each vector $\mathbf{v_i}$ is extended to include zero and negative indices and indices larger than N as specified in Expressions 5.47. The added elements in this extension can also be copies of the first and last elements, respectively, of the vector $\mathbf{v_i}$ in the same order as they appear in $\mathbf{v_i}$. Then the local square vector $\mathbf{v_i^{(ls)}}$ is computed using Expression 5.56.

3. **Constant padding**

   Each vector $\mathbf{v_i}$ is extended to include zero and negative indices and indices larger than N as specified in Expressions 5.49. Then the local square vector $\mathbf{v_i^{(ls)}}$ is computed using Expression 5.56.

III. **Computing the local variance vector**

For each pixel with index k, where k is from 1 to N, each element of the local variance vector $\mathbf{v_i^{(lv)}}$ is computed by:

$$v_i^{(lv)}(k) = v_i^{(ls)}(k) - \left(v_i^{(lm)}(k)\right)^2 \tag{5.59}$$

IV. **Computing the scaling coefficient vector**

For each pixel with index k, where k is from 1 to N, each element of the scaling coefficient vector $\mathbf{d_i}$ is computed by:

$$d_i(k) = \left(\frac{\max\left(0, v_i^{(lv)}(k) - \sigma_w^2\right)}{\max\left(v_i^{(lv)}(k), \sigma_w^2\right)}\right)^{\beta_w}, \tag{5.60}$$

where σ_w^2 is the Wiener filter variance and β_w is the Wiener beta coefficient. Since in Expression 5.60 the numerator is always smaller than the denominator, by raising the ratio to power $\beta_w > 1$, the scaling coefficient $d_i(k)$ becomes smaller than when $\beta_w = 1$. Conversely, by raising the ratio to power $\beta_w < 1$, the scaling coefficient $d_i(k)$ becomes greater than when $\beta_w = 1$. Therefore, the Wiener filter beta coefficient β_w controls the relative weight put on the scaling factor with respect to the difference between the local variance $v_i^{(lv)}(k)$

and the Wiener filter variance σ_w^2. We observed that when $\beta_w = 1$, the overall performance is good and the implementation is simple because there is no raising to power; however, other values of β_w can also be used, in particular $\beta_w = 2$. Also, generally, β_w need not be integer.

The Wiener filter variance σ_w^2 is a critically important parameter that determines the overall performance. Its optimal value is related to the scanner pattern variance σ_s^2 (and to its derived parameter σ_t^2 in case of inverse signal mode) because the level of the filtering effect of the Wiener filter directly depends on the value of σ_w^2. However, using the value of σ_s^2 (or σ_t^2) or its estimate to derive a value for σ_w^2 in a simplistic way is not recommended because:

(a) For the Wiener filter, the "signal" is the fingerprint pattern, while the "noise" is the combined effect of the scanner pattern and the (spatial) scanner noise;

(b) σ_w^2 is a tradeoff parameter that controls the relationship between the FAR and the FRR;

(c) Other factors, such as varying environmental conditions, may require adjustment of σ_w^2 in order to compensate for these variations.

Therefore, the best value for σ_w^2 is typically the result of optimization and tests, preferably with a great number of scanners of the same type and under different environmental conditions. When doing such optimization is not feasible, as a very approximate guideline, in direct signal mode, σ_w^2 can be set to the sum of the estimates for the scanner pattern variance σ_s^2 and the scanner noise variance σ_n^2. For the direct signal mode implementation of the Preprocessing Module, σ_w^2 can be set to about 30 for Signal Model A and to about 8 for Signal Model B. For the inverse signal mode implementation of the Preprocessing Module, σ_w^2 can be set to about $3 * 10^{-8}$ for Signal Model A and to about $4 * 10^{-9}$ for Signal Model B.

V. **Computing the smoothed signal**

For each pixel with index k, where k is from 1 to N, each element of the smoothed signal vector $\mathbf{v_i^{(s)}}$ is computed by:

$$v_i^{(s)}(k) = v_i^{(lm)}(k) + d_i(k) . \left(v_i(k) - v_i^{(lm)}(k) \right) . \qquad (5.61)$$

Once the smoothed signal vectors $\mathbf{v_i}^{(s)}$, where i is an integer from 1 to c, are computed, they are concatenated one after each other to form the smoothed signal $\mathbf{v^{(s)}}$:

$$\mathbf{v^{(s)}} = \begin{bmatrix} \mathbf{v_1^{(s)}} \\ \mathbf{v_2^{(s)}} \\ \cdots \\ \mathbf{v_c^{(s)}} \end{bmatrix} .$$

Finally, the output signal $\mathbf{x}$ of the Filtering Module in this implementation is the difference between the input signal $\mathbf{v}$ and the smoothed signal $\mathbf{v^{(s)}}$, corrected with the Wiener mean μ_w:

$$x(k) = v(k) - v^{(s)}(k) + \mu_w$$

where k is the current pixel index, an integer from 1 to $(c.N)$. In the preferred implementation, the Wiener filter mean μ_w is set to 0, but other values of μ_w

are also possible as μ_w can be used to compensate in case when fixed-valued offset is present so that the output signal $\mathbf{v}$ becomes zero mean. However, setting a non-zero value for μ_w may require adjustments in the thresholds of the various implementations of the Masking Module.

Figures 5.62 and 5.63 in the Magnitude Masking Implementation for the Adaptive Wiener Filter (Section 5.5.1.4) show examples for signals processed with this implementation in both direct and inverse signal modes.

5.5.1.4 Masking Module

The Masking Module (see Figure 5.59) marks as usable those pixels from the input signal $\mathbf{x}$ that contain sufficiently accurate estimate of the scanner pattern and as unusable the remaining pixels, producing a binary mask as the output signal $\mathbf{y}$.

When using the Low-pass Filter Implementation or the Adaptive Wiener Filter Implementation of the Filtering Module, the following observations regarding masking can be made for the four types of regions below:

(i) In the regions of transitions between a valley and a ridge, the adjacent pixels have very different grayscale values, which makes the local mean signal $\mathbf{v}^{(\mathbf{lm})}$ significantly different from the input signal $\mathbf{v}$. Consequently, the output signal $\mathbf{x}$ is large and considerably different from the scanner pattern, which makes using the output signal $\mathbf{x}$ as a scanner pattern estimate difficult. Therefore, such regions are marked not to be used, i.e., with mask value 0.

(ii) In the regions with ridges where either (a) the adjacent pixels have considerably different grayscale values (which is similar to case (i)), with this difference particularly pronounced when using the inverse signal model, or (b) the pixel values are saturated because of the particular fingerprint scanner implementation, thus all of these pixels have the same constant grayscale value, resulting in an output signal $\mathbf{x}$ being equal to 0, rendering it not useful as a scanner pattern estimate. Therefore, such regions are marked not to be used, i.e., with mask value 0.

(iii) In the regions where the adjacent pixels have grayscale values approximately equal to each other, very often such pixels actually are significantly corrupted by the scanner noise, which makes using the output signal $\mathbf{x}$ as a scanner pattern estimate difficult. Therefore, such regions are marked as not to be used, i.e., with mask value 0.

(iv) In the regions with valleys and in the regions without a fingerprint pattern being present (i.e., no fingertip skin touches the scanner platen), the difference between the input signal $\mathbf{v}$ and the local mean signal $\mathbf{v^{(lm)}}$ provides an accurate estimate of the scanner pattern. Therefore, such regions are marked as useful, i.e., with mask value 1.

The Masking Module has several implementations which are described next.

Threshold Implementation

When using the Bypass Implementation of the Filtering Module, the Masking Module is implemented as Threshold Implementation. In the direct signal mode of the Preprocessing Module, the pixel values in the regions with valleys are greater than the pixel values in the regions with ridges for both Signal Model A and Signal Model B (see Expression 5.33). Therefore, by comparing the values of each pixel with a threshold, a decision can be made as to whether the pixel can be used to estimate the scanner pattern or not. Thus, in the direct signal mode, for each pixel index k from 1 to $(c.N)$, the output signal $\mathbf{y}$ is computed by:

$$y\left(k\right)=\begin{cases}1 & \text{if } \ u\left(k\right)\geq\theta\\ 0 & \text{otherwise.}\end{cases} \tag{5.62}$$

where θ is the threshold value. The comparison can be done also in the inverse signal mode of the Preprocessing Module, in which case the condition for 1 in Expression 5.62 is replaced by $u\left(k\right)\leq\left(1/\theta\right)$.

The value of θ depends strongly on the overall level of the scanner pattern and to its mean and variance; methods for computing different estimates for them were described in Section 5.4.3. When an estimate $\bar{s}$ of the scanner pattern mean μ_s for a particular fingerprint scanner is available, then θ can be set fixed for all pixel indices k to about 90% of $\bar{s}$. When local estimates $\hat{\mu}_s\left(i,j\right)$ of the scanner pattern mean μ_s are available, then θ can be set separately for each index k to about 90% of the local estimate $\hat{\mu}_s\left(i,j\right)$ that corresponds to the pixel in question. When, in addition to estimates of the scanner pattern mean μ_s, an estimate $\hat{\sigma}_s^2$ of the scanner pattern variance σ_s^2 is also available, the threshold θ can be set to about $3\hat{\sigma}_s$ below the estimate ($\bar{s}$ or $\hat{\mu}_s\left(i,j\right)$, whichever is available) of the scanner pattern mean μ_s.

When no estimate of the scanner pattern mean μ_s is available, the threshold θ can be computed using the following method. First, the unique values of signal

$\mathbf{v}$ are stored in a vector $\mathbf{U_v}$ such that each element of the vector $\mathbf{U_v}$ represents each unique value of $\mathbf{v}$ and the elements of $\mathbf{U_v}$ are sorted in ascending order, with the first element of $\mathbf{U_v}$ being equal to the smallest value of $\mathbf{v}$ and the last element of $\mathbf{U_v}$ being equal to the largest value of $\mathbf{v}$. This is possible because the signal $\mathbf{v}$ is either equal to the signal $\mathbf{g}$, which is a discrete-valued (quantized) signal, e.g., its range is the integers from 0 to 255 when $\mathbf{g}$ is an 8-bit grayscale image, or to its inverse, the signal $\mathbf{h}$, which is also a discrete-valued signal when $\mathbf{g}$ is discrete valued. Furthermore, when $\mathbf{g}$ is an 8-bit grayscale image, the vector $\mathbf{U_v}$ has at most 256 elements in both cases.

Next, a new vector $\mathbf{D_v}$, derived from the vector $\mathbf{U_v}$, is computed by:

$$D_v\left(k\right) = U_v\left(k+1\right) - U_v\left(k\right) \tag{5.63}$$

where the index k runs such as to address all elements of the vector $\mathbf{U_v}$. The vector $\mathbf{D_v}$ contains differences between the values of adjacent elements of the vector $\mathbf{U_v}$ and thus it carries information about "gaps" in the unique pixel values of the image $\mathbf{g}$.

In implementations where the direct signal mode of the Preprocessing Module is used, the last Q elements of the vector $\mathbf{D_v}$ are then inspected. A large-value element in these Q elements of the vector $\mathbf{D_v}$, which element corresponds to a large difference between adjacent elements in the vector $\mathbf{U_v}$, may mark the separation between the values of two groups of pixels of the image: (1) a group of pixels that correspond to abnormally operating sensing elements (outliers), such as dead and defective pixels which are unique for every fingerprint scanner, and (2) a group of pixels that correspond to valleys in the image. Thus, the largest value of these Q

elements of the vector $\mathbf{D_v}$ can be considered as an indicator about a split between these two groups of pixels. Let the index of this largest value element be R. Then, the average of the elements with indices from 1 to $(R-1)$ is computed by:

$$\bar{d} = \frac{1}{(R-1)} \sum_{k=1}^{R-1} D_v\left(k\right)$$

If the value of the element $D_v\left(R\right)$ is over about 4 times greater than this average $\bar{d}$, i.e., if $D_v\left(R\right) > 4\bar{d}$, then the value of the element $U_v\left(R\right)$ indicates the peak value from which the threshold θ can be computed by:

$$\theta = U_v\left(R\right) - \Delta$$

where Δ is a suitably chosen constant, which is about 32 for Signal Model A and about 18 for Signal Model B, both for the case when the direct signal mode of the Preprocessing Module is used. When an estimate $\hat{\sigma}_s^2$ of the scanner pattern variance σ_s^2 is available, then the constant Δ can be set to about 6 times this estimate $\hat{\sigma}_s$. This factor 6 can be motivated by the observation that since (the variable part of) the scanner pattern $\mathbf{s}$ has a Gaussian distribution, then thus selected threshold θ can be thought to be about $3\sigma_s$ below the scanner pattern mean μ_s (which is unknown) and the peak value $U_v\left(R\right)$ to be about $3\sigma_s$ above the scanner pattern mean μ_s. Thus, all pixels that are about $\pm 3\sigma_s$ around the (unknown) scanner pattern mean μ_s are being marked as useful.

In implementations where the inverse signal mode of the Preprocessing Module is used, the processing is analogous. Since the inversion (Expression 5.34) transforms large numbers into small ones and small numbers into large ones, several adjustments have to be made: (i) the first Q elements of the vector $\mathbf{D_v}$ are inspected, not the last ones, (ii) a small-value element in these Q elements of the vector $\mathbf{D_v}$ corresponds to a large difference between pixels in the image $\mathbf{g}$, therefore the smallest value of these Q elements can be considered as an indicator about the split point, (iii) the inequality condition has the opposite direction, and (iv) the constant Δ can be set to about 6 times the value of σ_t, which σ_t is derived from the estimate $\hat{\sigma}_s^2$ of the scanner pattern variance.

In an alternative implementation where no estimates of the scanner pattern characteristics (mean μ_s and standard deviation σ_s) are available, the threshold θ can be set to about 185 for Signal Model A and to about 210 for Signal Model B and used in Expression 5.62. When the inverse signal mode of the Preprocessing Module is utilized, the condition in Expression 5.62 is replaced by $u\,(k) \leq (1/\theta)$ and the same values for the threshold θ are used.

The Threshold Implementation was inspired by Scenario B and Scenario C algorithms of our initial research (see Section 5.2.4.2). The Threshold Implementation is simple to implement, especially when the threshold θ is set without determination, but its performance under changing environmental conditions, such as temperature and moisture, may be suboptimal if the value of the threshold θ is not adjusted to changes of the mean of the scanner pattern due to the different environmental conditions.

Magnitude Masking Implementation for the Low-pass Filter

In this implementation, by comparing the magnitude (i.e., the absolute value) of the elements of the input signal $\mathbf{x}$ with two thresholds, a decision can be made as to whether the pixel can be used to estimate the scanner pattern or not. Thus, in the direct signal mode of the Preprocessing Module, for each pixel index k from 1 to $(c.N)$, the output signal $\mathbf{y}$ is computed by:

$$y\left(k\right)=\begin{cases}1 & \text{if } \ \varphi_{\min}\leq\left|x\left(k\right)\right|\leq\varphi_{\max}\\ 0 & \text{otherwise}\end{cases} \tag{5.64}$$

where $\varphi_{\min}$ and $\varphi_{\max}$ are two threshold values.

Figure 5.60 shows example signals $\mathbf{v}$ and $\mathbf{x}$ for one and the same column (#43) of an image with a fingerprint (thumb) and an image with air in an implementation using direct signal mode, a 3-tap moving-average filtering, and magnitude masking. For better visibility, only the first 60 rows (one-sixth of all) are shown in the figure. The reason for choosing one of the images with air instead of an image with another fingerprint is to better illustrate the process visually as when both images are with fingerprints, the commonly masked pixels in a single column are very few and sparsely distributed along the rows, and the figure becomes difficult to understand. An image with air contains both the scanner pattern and the scanner noise, and this is sufficient for illustrative purposes.

The moving-average filter, used to “denoise” $\mathbf{v}$, creates spikes in the signal $\mathbf{x}$ in the transition regions from valleys to ridges (and vice versa). As we already explained, extracting the scanner pattern from these transition regions is difficult because the scanner pattern is much weaker than the fingerprint pattern. On the other hand, in the regions with plateaus, i.e., in valleys and ridges as well as in areas where no fingertip is present, the signal $\mathbf{x}$ is weak. Therefore, by marking as not useful the pixels with magnitudes above the higher threshold φ_{max}, we discard

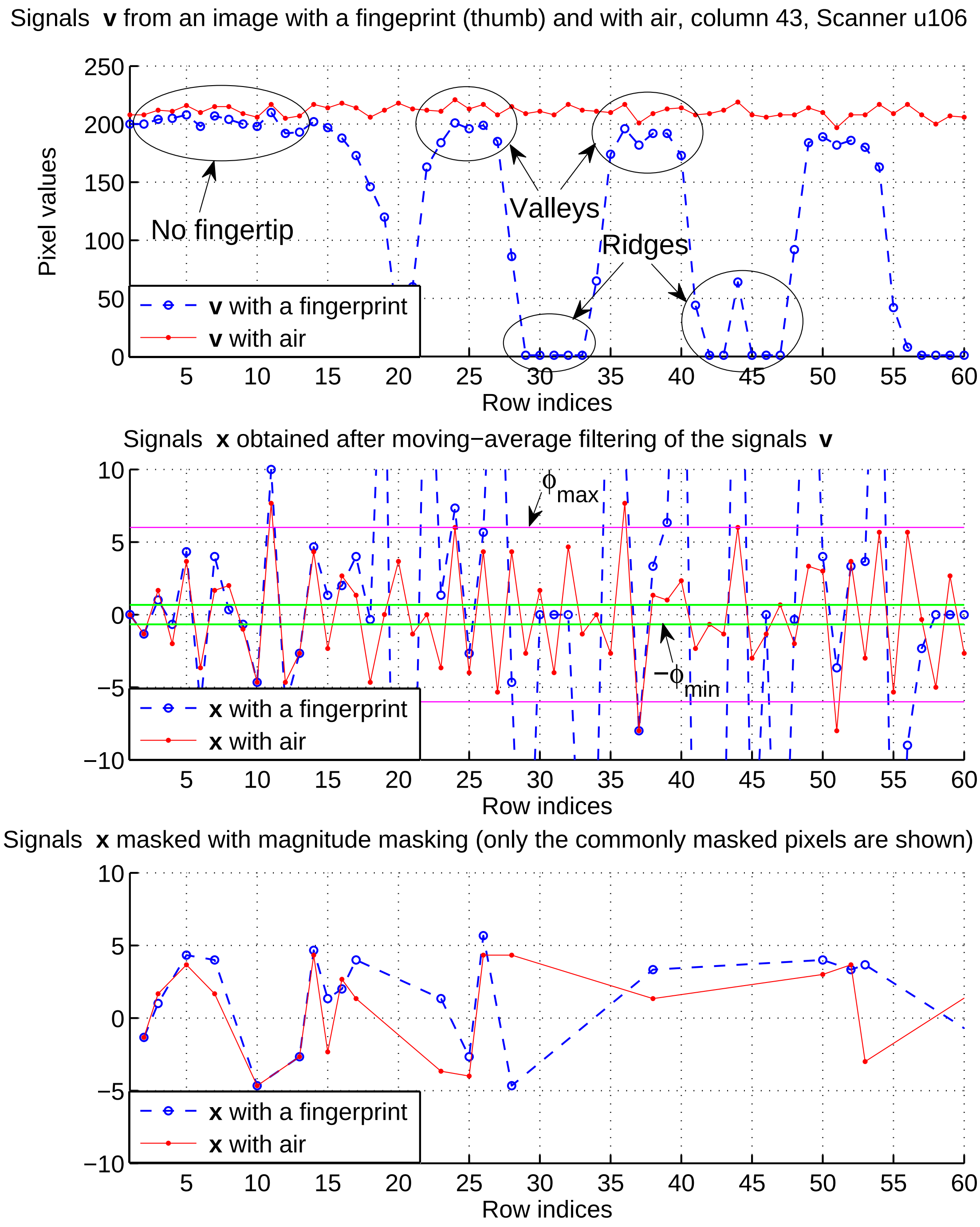


Figure 5.60: Advanced algorithms for area scanners: example signals in an implementation with moving-average filtering and magnitude masking in direct mode

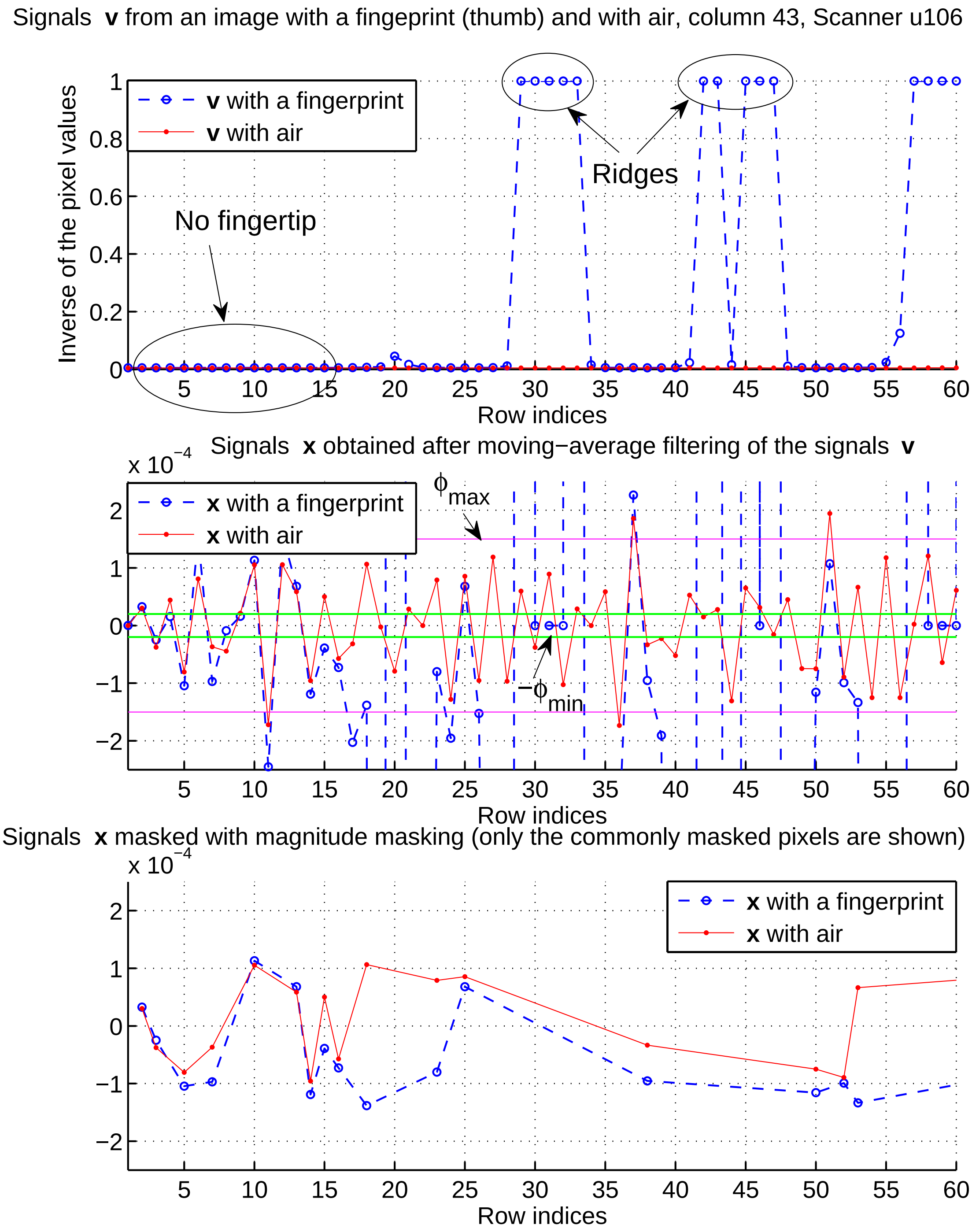
the pixels in the transitions, while retaining those in the plateaus. Because in the regions with valleys, the signal $\mathbf{x}$ is essentially scanner pattern and scanner noise, choosing the threshold φ_{max} as a function of their combined variance will retain considerable number of pixels containing the scanner pattern. However, of course, there will always be incorrectly discarded pixels as the scanner pattern can have much larger values than its (combined with the scanner noise) standard deviation, as it is visible from the figure (e.g., at rows #11 and #37).

Next, we observed that because of the saturation ("clipping") problem in some of the ridges (see the figure), no scanner noise can be extracted from them. The moving-average filtering in this case yields 0 in signal $\mathbf{x}$ because there is no "noise" at these pixels. The second observation is that sometimes pixels for which $\mathbf{x}$ has a very small magnitude contain predominantly noise because the scanner noise variance is much smaller than the scanner pattern variance and therefore in pixels with very small magnitudes, the scanner noise is more likely to dominate over the scanner pattern (because both are zero-mean Gaussians), in which case it is better not to use these pixels. These two observations led to the conclusion that a lower threshold φ_{min} is also necessary, which logically, is related to the scanner noise variance.

The lowest plot in Figure 5.60 shows the common pixels after masking both signals $\mathbf{x}$. As designed, these are only (a) pixels in the valleys and (b) pixels where no fingertip is present, which number in this case is much larger than the number of pixels in (a) because the segment of rows shown in the figure is close to the edge of the scanner platen. The total number of common pixels in the whole column (for all 360 rows) is 89, which is about 25% and considerably higher than in a typical case with two images with fingerprints. The correlation coefficient between these 89 common pixels is 0.6209.

Figure 5.61 shows exactly the same two columns of pixels from the same two images as in Figure 5.60. The moving-average filtering and the magnitude masking are also the same; the only difference is that here the two columns are processed in inverse signal mode. Because of the inversion, the ridges are now on top, and the valleys become too small to be discernible in the figure; the latter also applies to the signal **v** with air. The reasoning, however, remains exactly the same: the pixels in the transition regions are discarded and the pixels in signal **x** with too small magnitudes are discarded as well. One important note: the inversion does transform large values into small values (and vice versa), but for the signals we process (Gaussians with means much larger than their standard deviations), a signal with large **variance** still has large **variance** after the inversion (provided that the mean is unchanged) and a signal with small variance still has small variance after the inversion (for details, see Section 6.5 in the appendix). Therefore, the higher threshold φ_{max} still should be computed in function of both the scanner pattern and the scanner noise variances; the reasoning for the lower threshold φ_{min} is similar.

The lowest plot in Figure 5.61 shows the common pixels after masking both signals **x**. The total number of common pixels in the whole column (for all 360 rows) is much smaller (58) or about 16%, although the correlation coefficient is slightly higher (0.6749). We believe that this is due to the values of the thresholds, which have not been optimized. However, one can notice that the overall pattern remains roughly the same (but, of course, inverted).



Signals **v** from an image with a fingeprint (thumb) and with air, column 43, Scanner u106

Signals **x** obtained after moving−average filtering of the signals **v**

Signals **x** masked with magnitude masking (only the commonly masked pixels are shown)

Figure 5.61: Advanced algorithms for area scanners: example signals in an implementation with moving-average filtering and magnitude masking in inverse mode

Hereby we propose some guidelines for computing the thresholds ϕ_{min} and ϕ_{max}. However, we advise doing some experimentation and tests in order to find the values giving the best overall performance. In the direct signal mode of the Preprocessing Module, $\varphi_{\min}$ can be set to about half of the scanner noise standard deviation σ_n and $\varphi_{\max}$ to about the square root of the sum of the scanner pattern variance σ_s^2 and the scanner noise variance σ_n^2, i.e., to about $\sqrt{\sigma_s^2 + \sigma_n^2}$. Thus, $\varphi_{\min}$ can be set to about 0.67 for Signal Model A and to about 0.5 for Signal Model B. When an estimate $\hat{\sigma}_s$ of the scanner pattern standard deviation σ_s is available, $\varphi_{\max}$ can be set to about $\sqrt{\hat{\sigma}_s^2 + 1.8}$ for Signal Model A and to about $\sqrt{\hat{\sigma}_s^2 + 1}$ for Signal Model B. When no estimate of the scanner pattern standard deviation is available, $\varphi_{\max}$ can be set to about 5.5 for Signal Model A and to about 3.6 for Signal Model B.

In the inverse signal mode of the Preprocessing Module, $\varphi_{\min}$ can be set to about $\sigma_n/2\mu_s^2$ and $\varphi_{\max}$ to about $\sqrt{(\sigma_s^2 + \sigma_n^2)/\mu_s^4}$. Thus, when an estimate $\bar{s}$ of the scanner pattern mean μ_s for a particular fingerprint scanner is available, $\varphi_{\min}$ can be set to about $0.67/\bar{s}^2$ for Signal Model A and to about $0.5/\bar{s}^2$ for Signal Model B. When no estimate of the scanner pattern mean μ_s is available, $\varphi_{\min}$ can be set to about $1.8 \cdot 10^{-5}$ for Signal Model A and to about $0.95 \cdot 10^{-5}$ for Signal Model B. When an estimate $\hat{\sigma}_s^2$ of the scanner pattern variance σ_s^2 is available, $\varphi_{\max}$ can be set to about $\sqrt{(\hat{\sigma}_s^2 + 1.8)/\bar{s}^4}$ for Signal Model A and to about $\sqrt{(\hat{\sigma}_s^2 + 1)/\bar{s}^4}$ for Signal Model B. When no estimate of the scanner pattern variance σ_s^2 is available, $\varphi_{\max}$ can be set to about $1.36 \cdot 10^{-4}$ for Signal Model A and to about $0.68 \cdot 10^{-4}$ for Signal Model B.

Variance Masking Implementation for the Low-pass Filter

In this implementation, by comparing an estimate of the local variance of the input signal x with two thresholds, a decision can be made as to whether the pixel can be used to estimate the scanner pattern or not. Thus, in the direct signal mode of the Preprocessing Module, for each pixel index k from 1 to $(c.N)$, the output signal $\mathbf{y}$ is computed by:

$$y(k) = \begin{cases} 1 & \text{if } \gamma_{\min} \leq \left| v^{(lv)}(k) \right| \leq \gamma_{\max} \\ 0 & \text{otherwise} \end{cases} \tag{5.65}$$

where $\gamma_{\min}$ and $\gamma_{\max}$ are two threshold values. The signal $\mathbf{v^{(lv)}}$ is the local variance computed by concatenating the local variance vectors $\mathbf{v_i^{(lv)}}$, where the index i is from 1 to c:

$$\mathbf{v^{(lv)}} = \begin{bmatrix} \mathbf{v_1^{(lv)}} \\ \mathbf{v_2^{(lv)}} \\ \cdots \\ \mathbf{v_c^{(lv)}} \end{bmatrix}. \tag{5.66}$$

Each vector $\mathbf{v_i^{(lv)}}$ in Expression 5.66 is computed using Expression 5.59, wherein M, the size of the moving-average window, is chosen to be about 5 for computing the local mean vectors $\mathbf{v_i^{(lm)}}$ and the local square vectors $\mathbf{v_i^{(ls)}}$ as specified in the Adaptive Wiener Filter implementation of the Filtering Module.

In the direct signal mode of the Preprocessing Module, $\gamma_{\max}$ can be set to approximately the sum of the scanner pattern variance σ_s^2 and the scanner noise variance σ_n^2, and $\gamma_{\min}$ can be set to about 50% of $\gamma_{\max}$. When an estimate $\hat{\sigma}_s^2$ of the scanner pattern variance σ_s^2 is available, $\gamma_{\max}$ can be set to about $(\hat{\sigma}_s^2 + 1.8)$ for Signal Model A and to about $(\hat{\sigma}_s^2 + 1)$ for Signal Model B. When no estimate of the scanner pattern standard deviation is available, $\gamma_{\max}$ can be set to about 30 for

Signal Model A and to about 14 for Signal Model B.

In the inverse signal mode of the Preprocessing Module, $\gamma_{\max}$ can be set to about $(\sigma_s^2 + \sigma_n^2)/\mu_s^4$, and $\gamma_{\min}$ can be set to about 50% of $\gamma_{\max}$. Thus, when an estimate $\bar{s}$ of the scanner pattern mean μ_s for a particular fingerprint scanner and an estimate $\hat{\sigma}_s^2$ of the scanner pattern variance σ_s^2 are available, $\gamma_{\max}$ can be set to about $(\hat{\sigma}_s^2 + 1.8)/\bar{s}^4$ for Signal Model A and to about $(\hat{\sigma}_s^2 + 1)/\bar{s}^4$ for Signal Model B. When no estimates of the scanner pattern mean μ_s or standard deviation σ_s is available, $\gamma_{\max}$ can be set to about $1.87 \cdot 10^{-8}$ for Signal Model A and to about $0.46 \cdot 10^{-8}$ for Signal Model B.

Magnitude Masking Implementation for the Adaptive Wiener Filter

In regions where the signal $\mathbf{v}$ changes very little, such as in valleys or in areas with no fingerprint pattern, the adaptive Wiener filter suppresses these small changes (because the "signal-to-noise" ratio with respect to the fingerprint pattern is small and the Wiener filter treats the signal as containing predominantly "noise"). Thus, the output of the filter, the signal $\mathbf{x}$, which is the difference between the signal $\mathbf{v}$ and its smoothed version, gives an accurate estimate of the scanner pattern. Therefore, the magnitude of the signal $\mathbf{x}$ can be used as a criterion for the usefulness of the pixel in question: if this magnitude is sufficiently large, the corresponding pixel is marked as useful. On the other hand, in regions where the signal $\mathbf{v}$ undergoes significant changes, such as in transitions between a valley and a ridge, the adaptive Wiener filter leaves the signal $\mathbf{v}$ almost unaltered (because the "signal-to-noise" ratio with respect to the fingerprint pattern is large and the Wiener filter does not suppress the "noise"). Thus, the magnitude of the difference between the signal $\mathbf{v}$ and its smoothed version is close to 0, and therefore using the signal $\mathbf{x}$ may lead to inaccurate estimate of the scanner pattern. For this reason, these regions are marked not to be used, i.e., they are discarded.

This process is illustrated in Figure 5.62. The columns of pixels used as signals $\mathbf{v}$ are exactly the same as in the Magnitude Masking Implementation for the Low-pass Filter (see Figure 5.60). Three clusters of pixels on each of the three subplots are boxed to serve as examples for the pairs of pixels that are masked as useful. The determination for this usefulness is shown in the middle plot: the corresponding pairs of signals $\mathbf{x}$ have magnitudes above the threshold, i.e., this is the magnitude masking criterion.

The lowest plot shows the common pixels after masking both signals $\mathbf{x}$. The total number of common pixels in the whole column (for all 360 rows) is only 37 or only about 10% of all; however, the correlation coefficient is very high (0.9462). One of the explanation for this is that the scanner pattern here is estimated more accurately than in case of moving-average filtering.

Figure 5.63 shows the same process in inverse signal mode. The first plot is enlarged so that the valleys can be seen clearly. The three boxed clusters of pixels correspond to the same clusters of pixels in the direct mode in Figure 5.62, with the difference being two more pixels determined here as useful. The visual comparison of the accuracy of the scanner pattern estimate between the inverse and the direct mode (the third plots in the two figures) is in favor of the inverse mode. The total number of common pixels in the whole column (for all 360 rows) is 34 and slightly

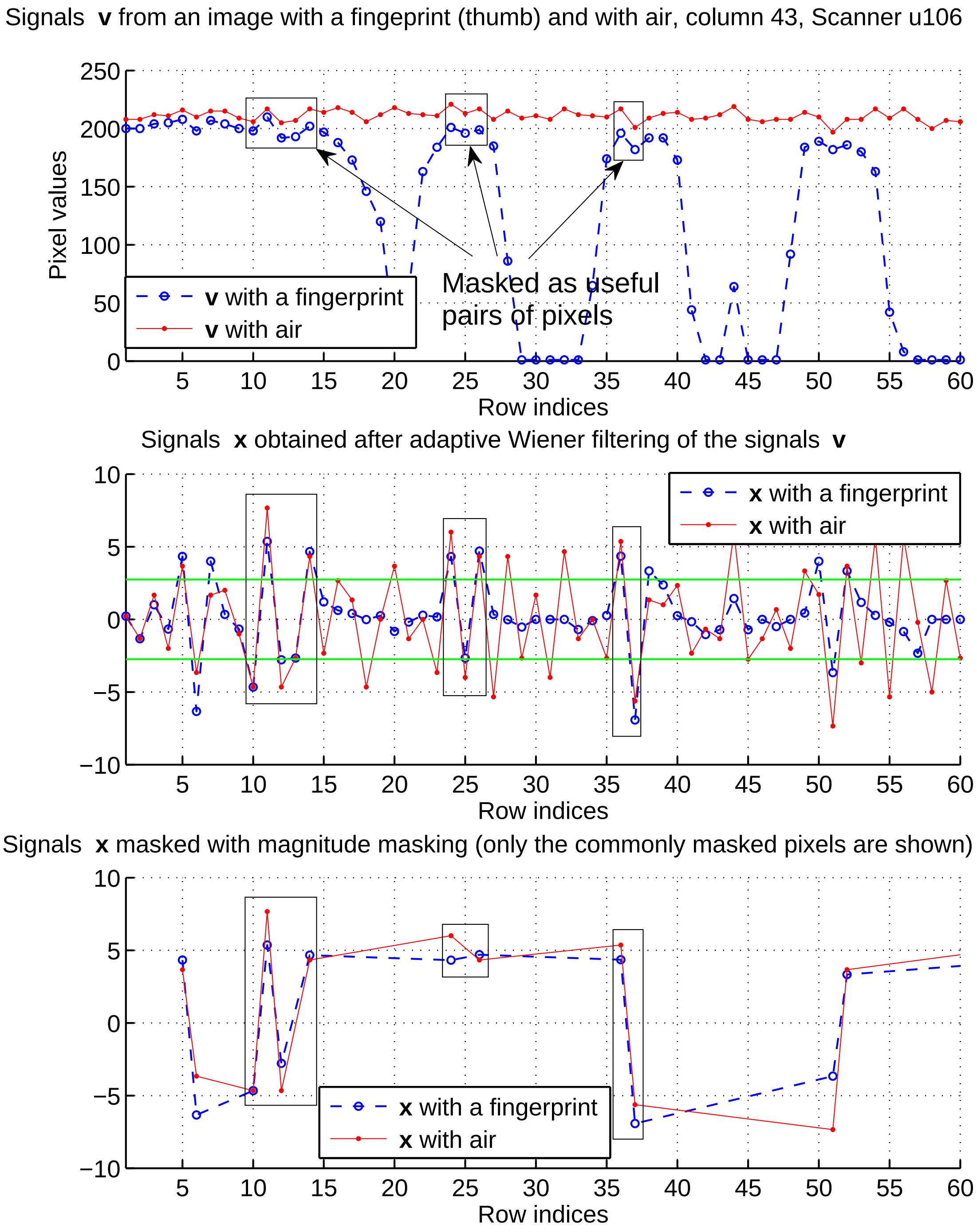


Figure 5.62: Advanced algorithms for area scanners: example signals in an implementation with adaptive Wiener filtering and magnitude masking in direct mode

less than that in the direct mode; the correlation coefficient is also slightly smaller (0.9413). However, the differences are too small to generalize from this particular example that the direct mode is better.

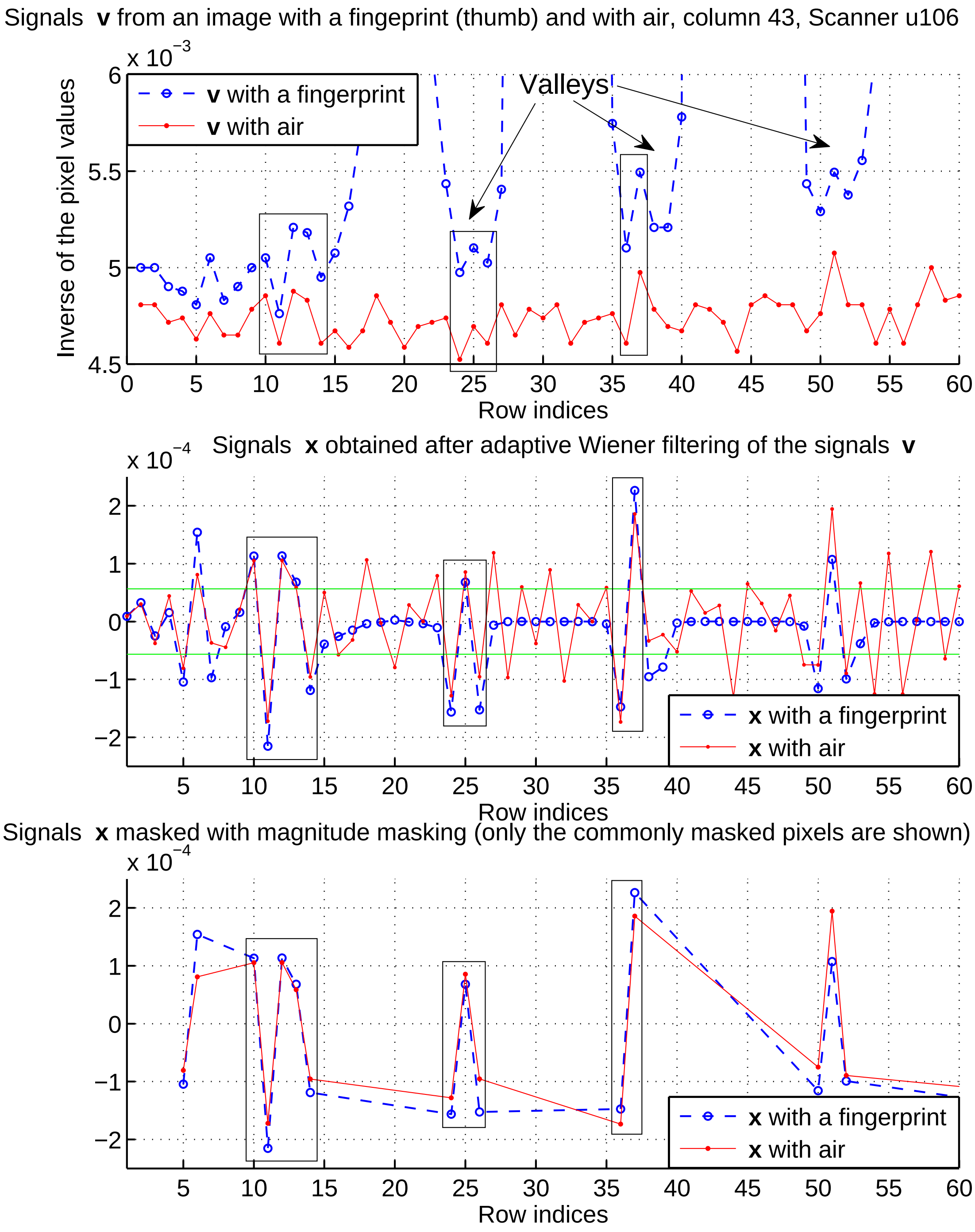
In summary, the magnitude masking for the Adaptive Wiener Filter implementation is as follows. For each pixel with index k from 1 to $(c.N)$, the output signal $\mathbf{y}$ is constructed by:

$$y(k) = \begin{cases} 1 & \text{if } \ |x(k)| \geq \alpha_w \cdot \sigma_w \\ 0 & \text{otherwise} \end{cases} \tag{5.67}$$

where σ_w is the square root of the Wiener variance as specified in the Adaptive Wiener Filter Implementation, and α_w is a suitably chosen scaling coefficient. For Signal Model A, α_w can be chosen to be about 0.50 in the direct signal mode and about 0.33 in the inverse signal mode. For Signal Model B, α_w can be chosen to be about 0.50 in either signal mode, direct or inverse.

Valley Masking Implementation for the Adaptive Wiener Filter

The objective of this mode of masking is determining the pixel indices of the regions with valleys directly from the signal $\mathbf{v}$. We propose a heuristics based on the observation that the derivative (gradient) of the signal $\mathbf{v_i}$ changes much less in the regions with valleys and ridges than in the regions of transitions from valleys to ridges and vice versa. Thus, by localizing the regions with small changes of the gradient, we identify the plateaus with valleys or ridges. The next step is discriminating between valleys and ridges in these plateaus, for which we observed that the distribution of the gradients of $\mathbf{v_i}$ in these plateaus can be of two different types:


Signals **v** from an image with a fingerprint (thumb) and with air, column 43, Scanner u106

Signals **x** obtained after adaptive Wiener filtering of the signals **v**

Signals **x** masked with magnitude masking (only the commonly masked pixels are shown)

Figure 5.63: Advanced algorithms for area scanners: example signals in an implementation with adaptive Wiener filtering and magnitude masking in inverse mode

1. **Unimodal distribution** for plateaus where no fingerprint pattern is present. In this case, all pixels can be used;

2. **Bimodal distribution** for plateaus with a fingerprint pattern. We observed that the pixels in the left (lower) distribution (mode) are typically plateaus with ridges, while the pixels in the right (higher) distribution (mode) are typically plateaus with valleys. We propose to identify the pixels in the latter plateaus by thresholding above a certain value.

The masking algorithm is as follows. First, for each vector $\mathbf{v_i}$ of the signal $\mathbf{v}$, where i is an integer from 1 to c, the gradient vector $\mathbf{v_i^{(g)}}$ is computed by:

$$v_i^{(g)}(k) = \frac{v_i(k+1) - v_i(k-1)}{2} \tag{5.68}$$

where k is an integer from 2 to $(N-1)$. The first and the last elements of the gradient vector $\mathbf{v_i^{(g)}}$ are computed by:

$$v_i^{(g)}(1) = v_i(2) - v_i(1) \tag{5.69}$$

$$v_i^{(g)}(N) = v_i(N) - v_i(N-1) \tag{5.70}$$

Let F_i be the set of all elements in $\mathbf{v_i}$ for which $\left|v_i^{(g)}(k)\right|$ is less than about 2 times the value of $\hat{\sigma}_s$ in direct signal mode or about 2 times the value of $\hat{\sigma}_t$ in inverse signal mode. This essentially is the criterion for localizing the plateaus with valleys and ridges. Next, a decision is made as to whether the distribution is unimodal or bimodal. Let μ_F be the mean value of the elements in F_i, m_{LF} be the mode of the histogram of those elements in F_i that are smaller than μ_F, and m_{RF} be the mode of the histogram of those elements in F_i that are greater than or equal to μ_F. The two cases are:

1. **Unimodal distribution.** If the difference $(m_{RF} - m_{LF})$ is smaller than about 2 times the value of $\hat{\sigma}_s$ in direct signal mode or about 2 times the value of $\hat{\sigma}_t$ in inverse signal mode, all elements in $\mathbf{v_i}$ can be used, i.e., $y\,(k) = 1$ for all k.

2. **Bimodal distribution.** Otherwise, $y\,(k)$ is set to 1 only for those k for which $\mathbf{v_i}\,(\mathbf{k})$ is greater than $(m_{RF} - \lambda)$, where λ is a predetermined value, which can be chosen to be about equal to $\hat{\sigma}_s$ in direct signal mode or to $\hat{\sigma}_t$ in inverse signal mode.

This algorithm is repeatedly applied to all vectors $\mathbf{v_i}$.

Threshold Masking Implementation for the Adaptive Wiener Filter

When using the Adaptive Wiener Filter Implementation of the Filtering Module, the Threshold Implementation of the Masking Module can be used in the same way as when using the Bypass Implementation of the Filtering Module.

5.5.1.5 Matching Module

The Matching Module (see Figure 5.59) computes a similarity score between the scanner patterns extracted from two images and produces a decision as to whether they are sufficiently similar or not.

Let $\mathbf{x_e}$ denote the output signal of the Filtering Module and $\mathbf{y_e}$ denote the output signal of the Masking Module when the input signal $\mathbf{g}$ is an image acquired during the scanner enrolment. Let $\mathbf{x_q}$ denote the output signal of the Filtering Module and $\mathbf{y_q}$ denote the output signal of the Masking Module when the input signal $\mathbf{g}$ is an image acquired during the scanner verification. Using the signals $\mathbf{x_e}$, $\mathbf{y_e}$, $\mathbf{x_q}$, and $\mathbf{y_q}$, the Matching Module:

(i) selects the common pixel indices marked as useful in the signals $\mathbf{y_e}$ and $\mathbf{y_q}$;

(ii) quantifies the similarity between the two signals $\mathbf{x_e}$ and $\mathbf{x_q}$ for these common pixel indices in a score;

(iii) produces a decision via the output signal d as to whether the two images have been acquired with the same fingerprint scanner by comparing this score with a threshold. When the output signal d is set to 1, this indicates scanner match; when it is set to 0, this indicates scanner non-match; and when it is set to (-1), this indicates that a decision on matching/non-matching cannot be made and a new query image must be acquired.

The selection of the common pixel indices marked as useful in the signals $\mathbf{y_e}$ and $\mathbf{y_q}$ produces the signal $\mathbf{y_m}$ so that:

$$y_m\left(k\right) = \begin{cases} 1 & \text{if } \ y_e\left(k\right) = 1 \ \text{ and } \ y_q\left(k\right) = 1 \\ 0 & \text{otherwise} \end{cases} \tag{5.71}$$

where the index k is an integer running from 1 to $(c.N)$. Let $\mathbf{D}$ be the set of all indices k for which $y_m\left(k\right) = 1$, and let N_D be the number of elements in this set $\mathbf{D}$.

If N_D is less than about 100, the Matching Module produces (-1) as the output signal d, which indicates that the number of common pixel indices is insufficient to compute a reliable similarity score and to make a decision thereof. In this case, acquiring a new query image is necessary.

Quantifying the similarity between the two signals $\mathbf{x_e}$ and $\mathbf{x_q}$ for the common pixel indices as computed in the signal $\mathbf{y_m}$ in a score can be done with the following three implementations.

Normalized Correlation Implementation

First, the norms of the signals $\mathbf{x_e}$ and $\mathbf{x_q}$ for the indices in the set $\mathbf{D}$ are computed:

$$\|\mathbf{x_e}\| = \sqrt{\sum_{k\in\mathbf{D}} |x_e(k)|^2} \qquad \|\mathbf{x_q}\| = \sqrt{\sum_{k\in\mathbf{D}} |x_q(k)|^2} \tag{5.72}$$

If either one of the norms $\|\mathbf{x_e}\|$ or $\|\mathbf{x_q}\|$ is equal to zero, d is set to 0 and no further computations are performed. Otherwise, the similarity score $z^{(nc)}$ is computed by:

$$z^{(nc)} = \frac{\sum_{k\in\mathbf{D}} x_e(k) \,.\, x_q(k)}{\|\mathbf{x_e}\| \,.\, \|\mathbf{x_q}\|} \tag{5.73}$$

The output signal d is then computed by comparing the similarity score $z^{(nc)}$ with a predetermined threshold:

$$d = \begin{cases} 1 & \text{if } \mathrm{z}^{(\mathrm{nc})} \geq \tau^{(\mathrm{nc})} \\ 0 & \text{otherwise} \end{cases} \tag{5.74}$$

The decision threshold $\tau^{(nc)}$ is the result of optimization and is in the range from about 0.4 to about 0.6.

Correlation Coefficient Implementation

First, the zero-mean signals $\tilde{\mathbf{x}}_\mathbf{e}$ and $\tilde{\mathbf{x}}_\mathbf{q}$ for the indices k in the set $\mathbf{D}$ are computed:

$$\tilde{x}_e\left(k\right) = x_e\left(k\right) - \frac{1}{N_D}\sum_{k\in\mathbf{D}} x_e\left(k\right) \qquad\qquad \tilde{x}_q\left(k\right) = x_q\left(k\right) - \frac{1}{N_D}\sum_{k\in\mathbf{D}} x_q\left(k\right) \tag{5.75}$$

where the index k runs through all elements in the set $\mathbf{D}$. The values of $\tilde{x}_e\left(k\right)$ and $\tilde{x}_q\left(k\right)$ for indices k that do not belong to the set $\mathbf{D}$ can be set to 0 or any other number because they will not be used in the computations that follow.

Next, the norms of the signals $\tilde{\mathbf{x}}_\mathbf{e}$ and $\tilde{\mathbf{x}}_\mathbf{q}$ for the indices k in the set $\mathbf{D}$ are computed:

$$\|\tilde{\mathbf{x}}_\mathbf{e}\| = \sqrt{\sum_{k\in\mathbf{D}} \left|\tilde{x}_e\left(k\right)\right|^2} \qquad\qquad \|\tilde{\mathbf{x}}_\mathbf{q}\| = \sqrt{\sum_{k\in\mathbf{D}} \left|\tilde{x}_q\left(k\right)\right|^2} \tag{5.76}$$

If either one of the norms $\|\tilde{\mathbf{x}}_\mathbf{e}\|$ or $\|\tilde{\mathbf{x}}_\mathbf{q}\|$ is equal to zero, d is set to 0 and no further computations are performed. Otherwise, the similarity score $z^{(cc)}$ is computed by:

$$z^{(cc)} = \frac{\sum_{k\in\mathbf{D}} \tilde{x}_e\left(k\right) .\ \tilde{x}_q\left(k\right)}{\|\tilde{\mathbf{x}}_\mathbf{e}\| . \|\tilde{\mathbf{x}}_\mathbf{q}\|} \tag{5.77}$$

The output signal d is then computed by comparing the similarity score $z^{(cc)}$ with a predetermined threshold:

$$d = \begin{cases} 1 & \text{if } \mathrm{z}^{(\mathrm{cc})} \geq \tau^{(\mathrm{cc})} \\ 0 & \text{otherwise} \end{cases} \tag{5.78}$$

The decision threshold $\tau^{(cc)}$ is the result of optimization and is in the range from about 0.4 to about 0.6.

Relative Mean Square Error Implementation

First, the norm of the signal $\mathbf{x_e}$ for the indices in the set $\mathbf{D}$ is computed:

$$\|\mathbf{x_e}\| = \sqrt{\sum_{k \in D} |x_e(k)|^2} \tag{5.79}$$

If the norm $\|\mathbf{x_e}\|$ is equal to zero, d is set to 0 and no further computations are performed. Otherwise, the similarity score $z^{(rmse)}$ is computed by:

$$z^{(rmse)} = \frac{\sqrt{\sum_{k \in \mathbf{D}} \left[x_e(k) - x_q(k)\right]^2}}{\|\mathbf{x_e}\|} \tag{5.80}$$

The output signal d is then computed by comparing the similarity score $z^{(rmse)}$ with a predetermined threshold:

$$d = \begin{cases} 1 & \text{if } \mathrm{z}^{(\mathrm{rmse})} \leq \tau^{(\mathrm{rmse})} \\ 0 & \text{otherwise} \end{cases} \tag{5.81}$$

The decision threshold $z^{(rmse)}$ is the result of optimization and is in the range from about 0.8 to about 1.1.

5.5.1.6 Using Multiple Images

All implementations described above can use a single image for the scanner enrolment and a single image for the scanner authentication, and this is preferred because (a) it requires the least number of computations and (b) it is the most secure as it determines if two images are taken with the same scanner or not without any additional images. However, variations are also possible. For example, it is typical for the biometric systems to capture three images and use them for enrolling the biometric information. Similarly, another implementation allows using multiple images for the scanner enrolment and/or multiple images for the scanner verification. This may improve the overall accuracy of the scanner authentication.

Let the number of enrolled images be E and the output signals of the Filtering Module and the Masking Module when the enrolled image with index r is being processed be $\mathbf{x_r}$ and $\mathbf{y_r}$, respectively. In the preferred implementation, the similarity scores for each pair consisting of one enrolled image and the query image are averaged and the resulting average similarity score is used to produce a decision. Thus, if the similarity score between the query image and the enrolled image with index r is denoted by z_r, computed using Expression 5.73, 5.77, or 5.80, then the average similarity score z_a is:

$$z_a = \frac{1}{E}\sum_{r=1}^{E} z_r. \tag{5.82}$$

Finally, the output signal d of the Matching Module is computed using Expression 5.74, 5.78, or 5.81, depending on which implementation of the Matching Module is used for computing the similarity scores z_r.

Another implementation computes an “average” enrolled scanner pattern from all enrolled images and uses this “average” enrolled scanner pattern in the Matching Module. First, the “average” mask $\mathbf{y_a}$ is computed by:

$$y_a(k) = \prod_{r=1}^{E} y_r(k) \tag{5.83}$$

where k is an integer running from 1 to $(c.N)$. Then the “average” scanner pattern is computed by:

$$x_a(k) = \frac{1}{N_a}\sum_{r=1}^{E} y_a(k).x_r(k) \tag{5.84}$$

where N_a is the number of elements in $\mathbf{y_a}$ for which $y_a\,(k) = 1$. Next, $\mathbf{x_a}$ is used instead of $\mathbf{x_e}$ and $\mathbf{y_a}$ is used instead of $\mathbf{y_e}$ in the Matching Module. The performance of this implementation may be suboptimal in certain cases because of two reasons: (1) since the signals $\mathbf{y_r}$ for different indices r (and thus different enrolled images) may be considerably different from one another, the "average" mask $\mathbf{y_a}$, which essentially is a logical AND of all $\mathbf{y_r}$, may have very few non-zero elements, which may result in fewer than sufficient number of pixels to be used in the Matching Module, and (2) the "average" signal $\mathbf{x_a}$ may become considerably distorted for some pixels and this may result in false scanner match or false scanner non-match decisions.

5.5.1.7 Combinations of Module Modes for the Area Scanners

All modes (implementations) of the {Selection Module, Filtering Module, Matching Module} can be used in combination with any of the modes of the modules that precede this current module in the conceptual signal flow diagram shown in Figure 5.59). Those of them that can be used in combination are suggested in the description of each module. However, different combinations of modes may provide different overall performance. Two well-performing combinations are described next.

Figure 5.64 shows the flowchart of one exemplary implementation using a single enrolled image $\mathbf{g_e}$, acquired and processed during the scanner enrolment, and a single query image $\mathbf{g_q}$, acquired and processed during the scanner verification. Although $\mathbf{g_e}$ and $\mathbf{g_q}$ are processed at different times, the consecutive processing steps are identical, and therefore we discuss them simultaneously. $\mathbf{g_e}$ and $\mathbf{g_q}$ are first processed by the

Preprocessing Module in its inverse signal mode. The output signals $\mathbf{u_e}$ and $\mathbf{u_q}$ are processed by the Selection Module selecting columns of pixels. The resulting signals $\mathbf{v_e}$ and $\mathbf{v_q}$ are processed by the Filtering Module with an adaptive Wiener filter. The Masking Module performs magnitude masking and produces the signals $\mathbf{y_e}$ and $\mathbf{y_q}$. Finally, the Matching Module computes the correlation coefficient and outputs the signal d, based on which a decision for scanner match or scanner nonmatch is made.

Figure 5.65 shows another exemplary implementation that uses moving-average filtering. The figure is self explanatory as the signals are the same and the operations are evident.

5.5.1.8 Performance

The advanced algorithms for the area scanners are scalable in performance (accuracy) as well as they are in complexity and in computational power. Our understanding is that the modes of each module should be chosen and the values of the parameters should be optimized once the available computational power (and other implementation constraints, e.g., memory and numerical precision) are set. Obviously, there is a tradeoff between performance and computational power/complexity. The performance of the advanced algorithms decreases gradually when decreasing the available computational power (e.g., when processing fewer pixels) and when choosing less complex modes. The algorithms also have a fail-safe guard: if the number of pixels to be used for computing the similarity score falls below a certain

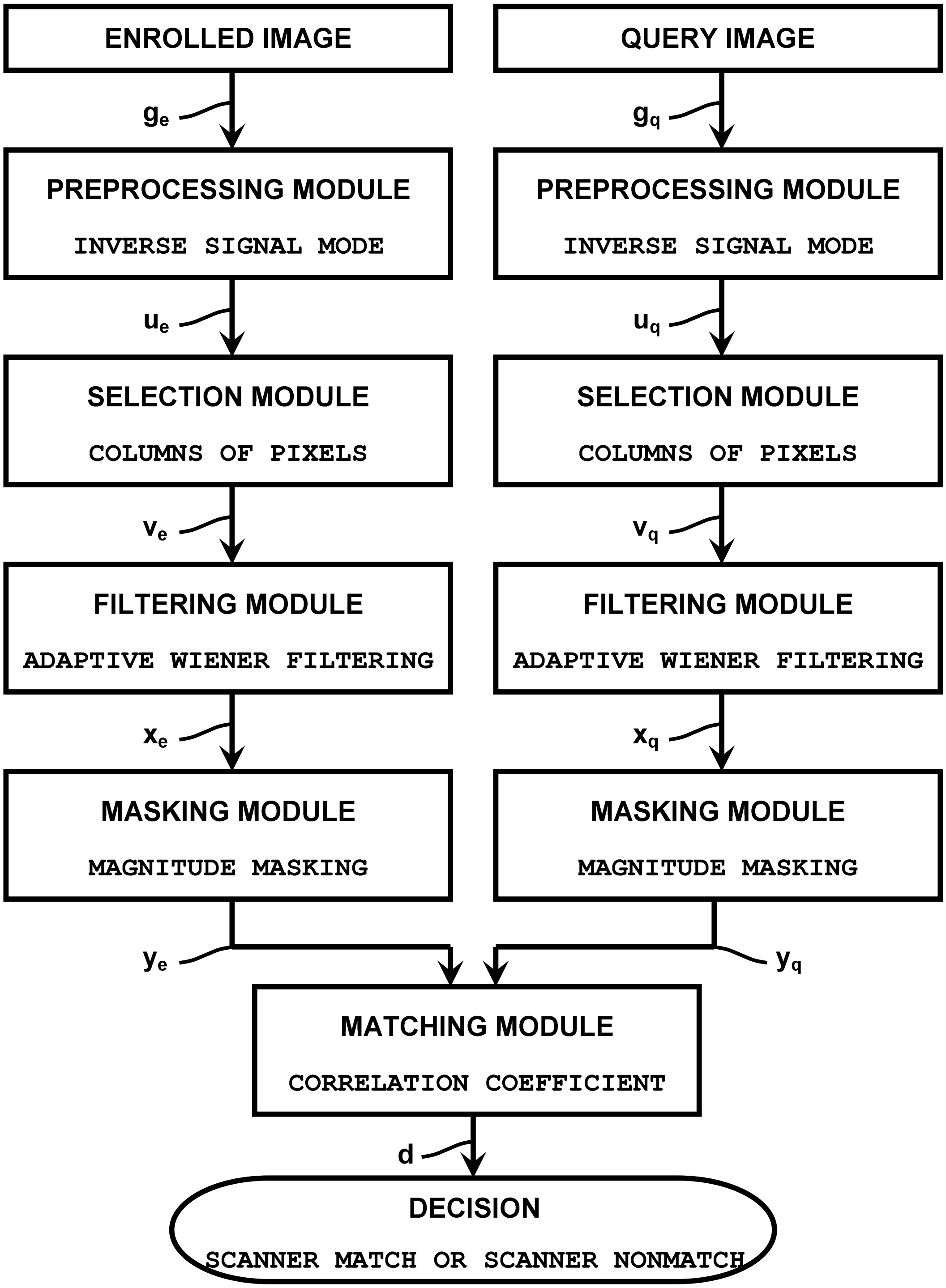


Figure 5.64: Advanced algorithms for area scanners: An exemplary implementation that uses Wiener filtering

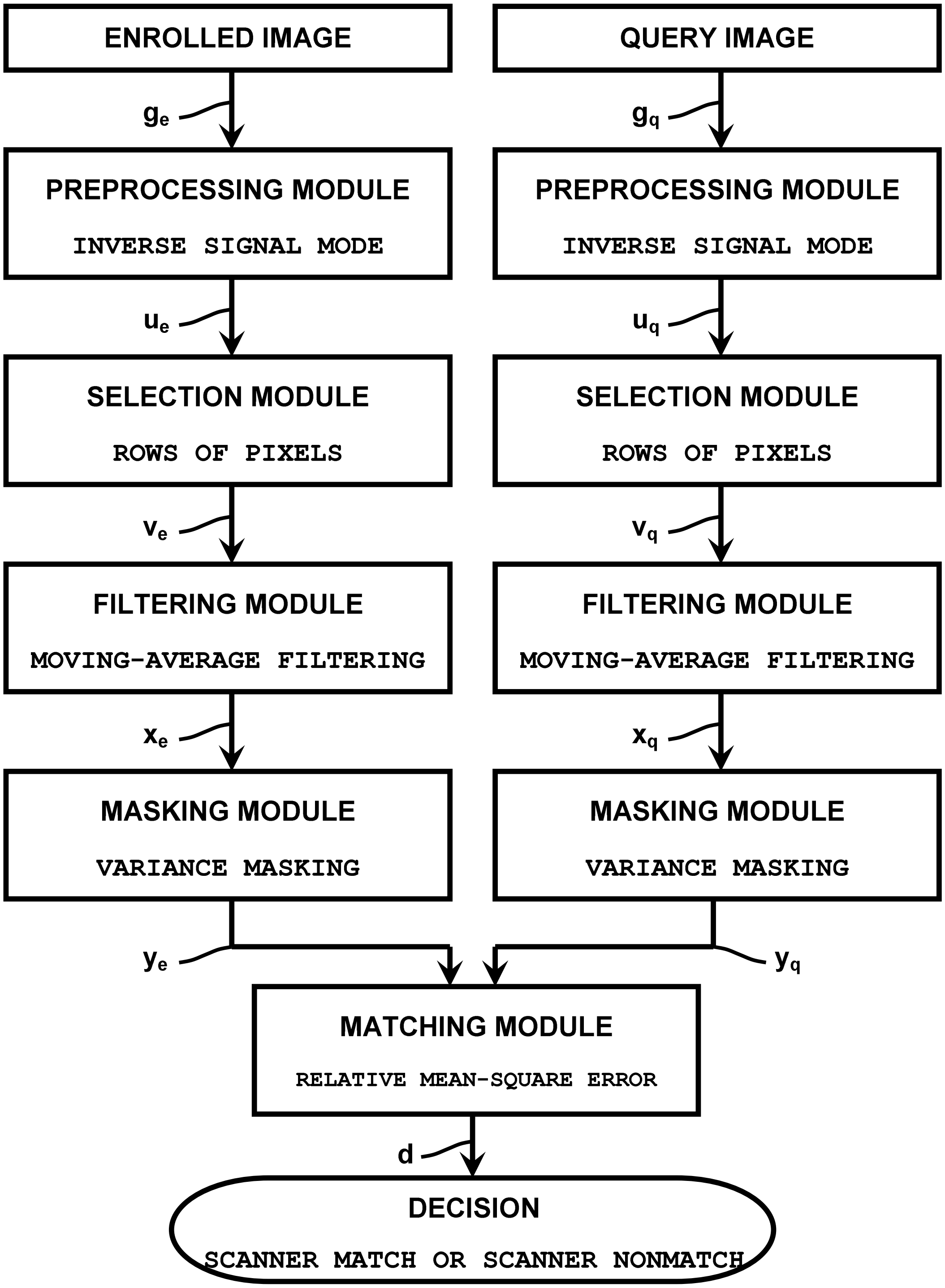


Figure 5.65: Advanced algorithms for area scanners: An exemplary implementation that uses moving-average filtering

value, the Matching Module flags this to avoid producing an unreliable decision.

For illustrative purposes, here we present the performance of the exemplary implementation shown in Figure 5.64. It uses inverse signal mode, 50 columns of pixels (which is slightly less than 20% of the image), 3-tap adaptive Wiener filter with variance $\sigma_w^2 = 3 \cdot 10^{-8}$ and $\beta = 1$, magnitude masking with scaling coefficient $\alpha_w = 0.326$, and correlation coefficient for matching. The normalized histograms (integrating to 1) of the correlation coefficients from running this implementation on 4,400 images acquired (at room temperature) with the 22 UPEK area scanners (see Section 6.1 in the appendix) for 10 images per finger for all 10 fingers of 2 persons are shown in Figure 5.66. Only a single image is used for scanner enrolment and only a single image for scanner verification. Every image is matched against all other images. Since all processing is completely symmetric for the enrolled image and the query image, only one of the matchings, AB or BA, is computed; the total number of computed matchings is about 10 million.

The two distributions of correlation coefficients (when the query image is acquired with the same scanner and when it is acquired with a different one) are very clearly separated, with their means about 0.75 apart. We chose the decision threshold (0.368) just below the smallest self correlation coefficient, and the empirical FRR is 0. The corresponding empirical FAR is $8.64 \cdot 10^{-7}$ (in the tails, the numerical precision of the empirical FAR and FRR is limited by the number of matchings).

We ran the same implementation (including with the same parameter values) on the 22 UPEK area scanners in the appendix) for 10 images per finger for 3 fingers

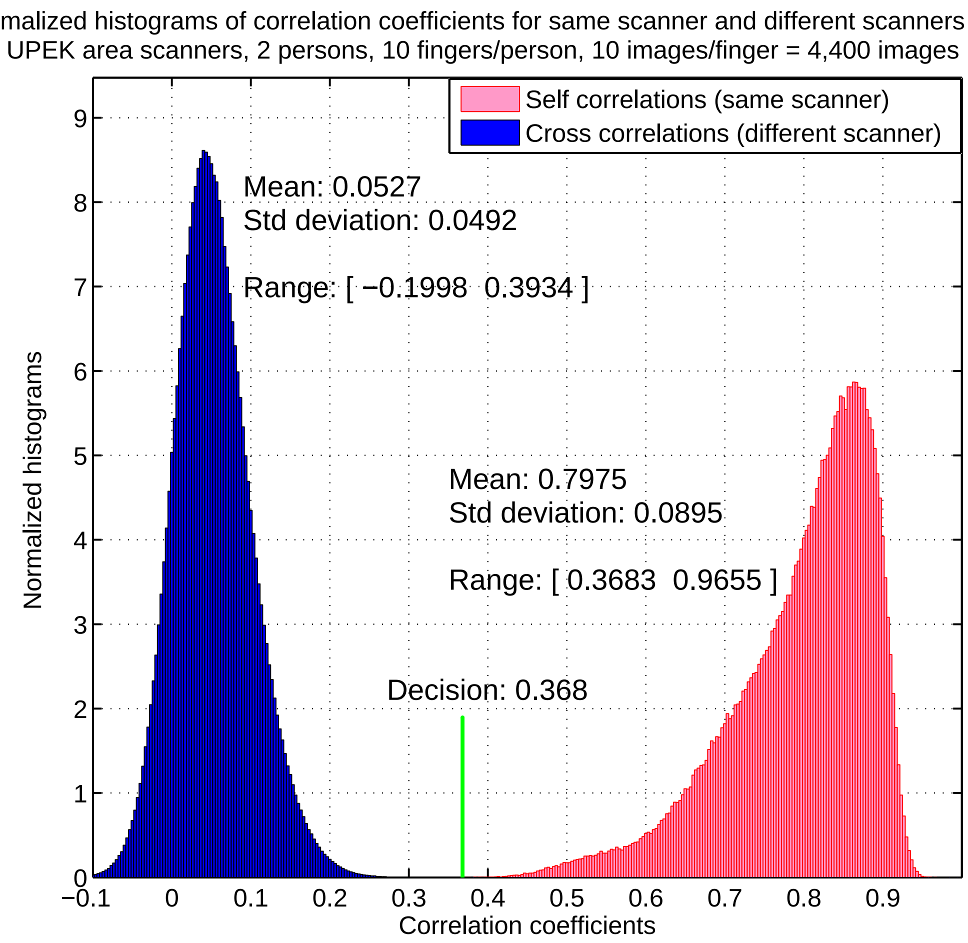


Figure 5.66: Performance of the advanced algorithms for the area scanners implemented with Wiener filtering

(right thumb, right index, and right little finger) of one person at 3 temperatures: at room temperature, at about 5 ^{o}C, and at about 40 ^{o}C (see Section 6.2 in the appendix); in total 1,980 images. The results are shown in Figure 5.67. When using the same decision threshold (0.368) as in Figure 5.66, no decision errors were registered, i.e., the empirical FAR = empirical FRR = 0. Somewhat surprising is that although the distribution of the self correlation coefficients in Figure 5.66 is substantially different than that of the self correlation coefficients in Figure 5.67 (with the thermal tests), both their means and standard deviations are almost identical.

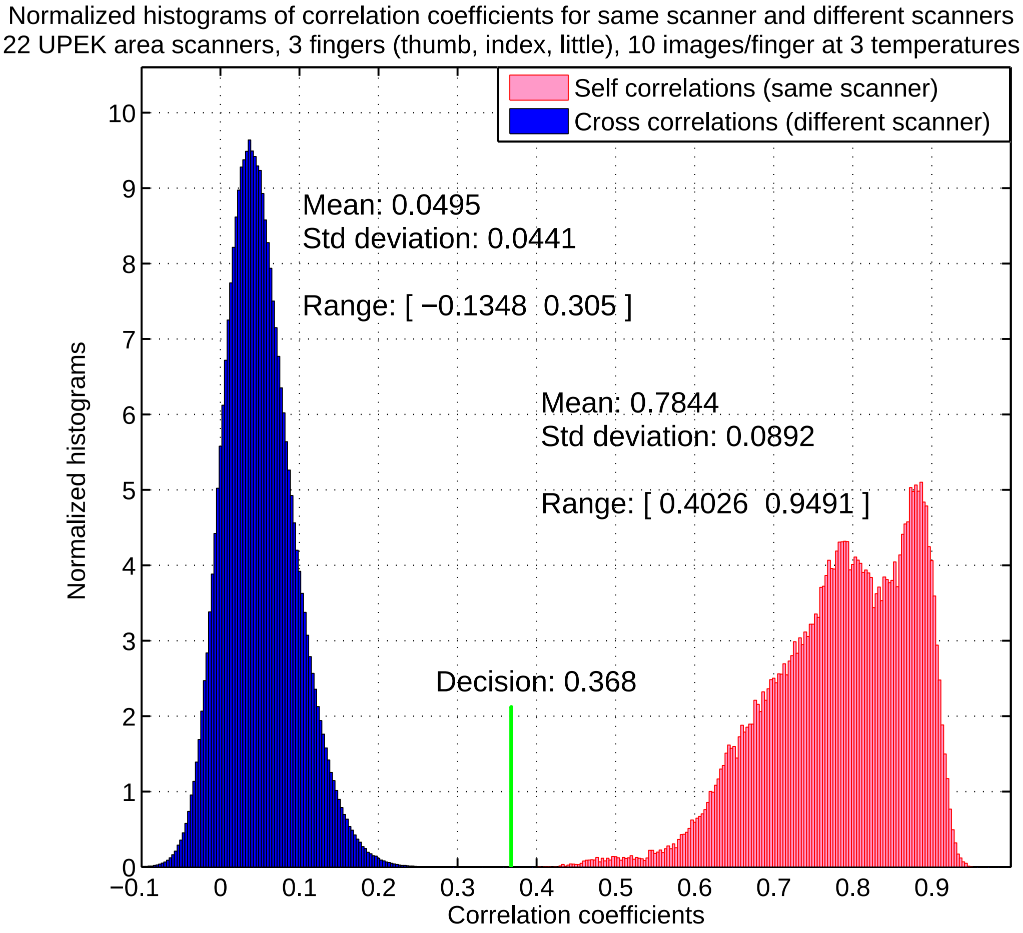


Figure 5.67: Performance of the advanced algorithms for the area scanners implemented with Wiener filtering under 3 different temperatures

5.5.2 Advanced Algorithms for the Swipe Scanners

Generally speaking, in the swipe scanners (also known as slide or sweep scanners), a line, being a row or a column, of sensor elements performs an instant scan of a tiny area of the fingertip skin and converts the readings into a line of pixels. As the fingertip is swiped over this line of sensor elements, a sequence of such lines of pixels is produced, which sequence is then assembled (and possibly further enhanced) to construct a two-dimensional fingerprint image.

The swipe scanners we used in this research are also from UPEK (details are provided in Section 6.1 in the appendix). Besides the fact that they, like the UPEK area scanners, also use capacitive sensing and besides the general technical information in scanners' datasheets, we were unable to find any publicly available information, including patents, with details about their specific acquisition or operation. Therefore, the only reasonable hypothesis was that they too use the same capacitive sensing cells as the UPEK area scanners, which led to the assumption that the values of the pixels they acquire follow Signal Model A. Although this assumption helped us develop the algorithms and it can, to a great extent, explain the algorithm operation, we have no other evidence about the applicability of this signal model to these fingerprint scanners. Furthermore, we also observed deviations from the results we expected, which also casts some doubt on that.

The first step, obviously, was to try to directly apply or extend the advanced algorithms of the area scanners for the swipe scanners. This, however, proved to be problematic because of three main reasons:

1. Although each UPEK swipe scanner contains several lines (rows) of sensing elements, our study on the topic concluded that only one of them scans the fingertip skin sequentially, row by row. The number of sensing elements per row depends on the type of swipe scanner; for the type of scanner we used, the sensing elements are only 144, producing images 144 pixels wide (and 384 pixels high). For comparison, the UPEK area scanners have nearly 3 orders of magnitude more sensing elements ($360 * 256 = 92,160$).

2. The software (libraries) acquire images only when there is a fingertip swiped over the scanner and therefore acquiring images with a predetermined object is impossible.

3. The software also combines the scanned lines and constructs a whole image from them. Sometimes the constructed images contain artifacts. And in addition to constructing an image, the software also enhances it. All this made acquiring the raw images we needed particularly difficult.

The swipe scanners, however, have two favorable properties over the area scanners:

- The pixels in the image never saturate ("clip"), unlike as the pixels in the UPEK area scanners do, and therefore all sensing elements in the row can be used. One possible explanation for this is that since the fingertip has to be swiped, pressing it hard enough as to saturate the sensing elements (which is easy with the UPEK area scanners) here is very difficult. This also led to another simplification (explained later).

- In a single image, each sensing element produces many (e.g., hundreds) pixel values, not only one pixel value as the area scanners do. In this way, the scanner pattern of each sensing element gets "incorporated" in many pixels of the image, thus facilitating the process of estimating it.

5.5.2.1 Averaging Along Columns

The last observation above naturally led to the idea that by averaging along columns, we should be able to "strengthen" the scanner pattern in an image. Combining this with the signal inversion essentially led to the solution.

From Signal Model A (see Expression 5.9) and neglecting the scanner noise $n(i, j, t)$ for now, for the pixel value at row i and column j, we have:

$$g(i, j) \approx \frac{s(i, j)}{1 + s(i, j) f(i, j)} \tag{5.85}$$

Since there is only one row of sensing elements, the scanner pattern along columns is the same, i.e., $s(i, j) = s(j)$ for all i. Next, inverting the signal gives:

$$h(i, j) = \frac{1}{g(i, j)} \approx \frac{1}{s(j)} + f(i, j) \tag{5.86}$$

As $s(j)$ is the same along each column j, averaging along columns will produce an average row $h_{avg}(j)$, which is still a function of an average fingerprint row $f_{avg}(j)$ (which is, however, random):

$$h_{avg}(j) \approx \frac{1}{s(j)} + f_{avg}(j) \tag{5.87}$$

An example for $h_{avg}(j)$ is shown in the upper plot in Figure 5.68. It is surprising how small local variations $h_{avg}(j)$ along the row (the column indices j) has. This is also difficult to believe given that the values of $h_{avg}(j)$ are averages along columns and that the pixel values $g(i,j)$ along nonadjacent columns can substantially differ. An example for the pixel values $g(i,j)$ of three such columns is shown in the lower plot of Figure 5.68.

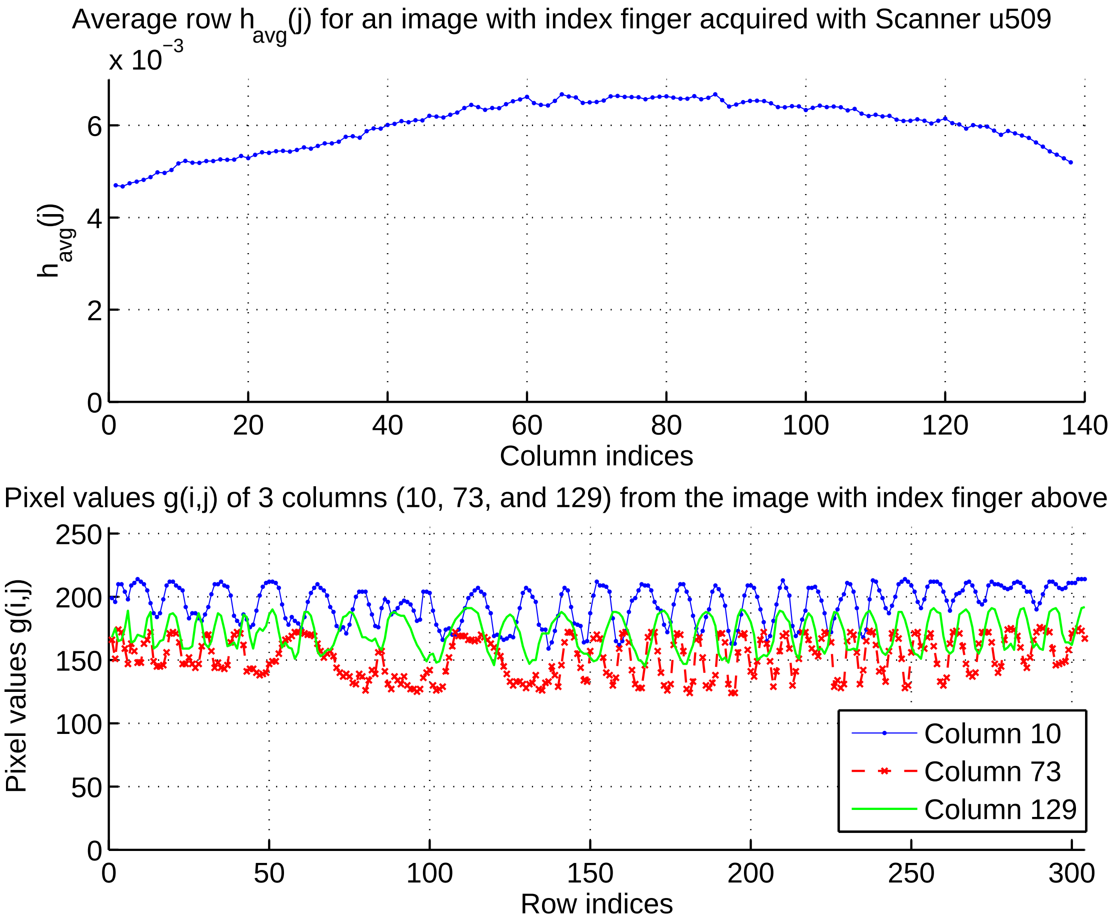


Figure 5.68: Swipe scanners: average row in inverse mode and 3 columns

However, by looking at the (inverses of) adjacent columns (an example for 3 adjacent columns is shown in Figure 5.69), we observed that although their pixel values (and their inverses as well) can somewhat differ, their averages are very close to each other. This can be explained with the high scanning spatial resolution of the scanners: they sample the fingertip skin at about 10 times faster rate than the typical frequency of the fingerprint pattern (i.e., the sequence of valleys and ridges). Consequently, the pixel values for adjacent columns cannot differ by much, and more importantly, the averages along adjacent columns should be close to each other, as we see from Figure 5.69.

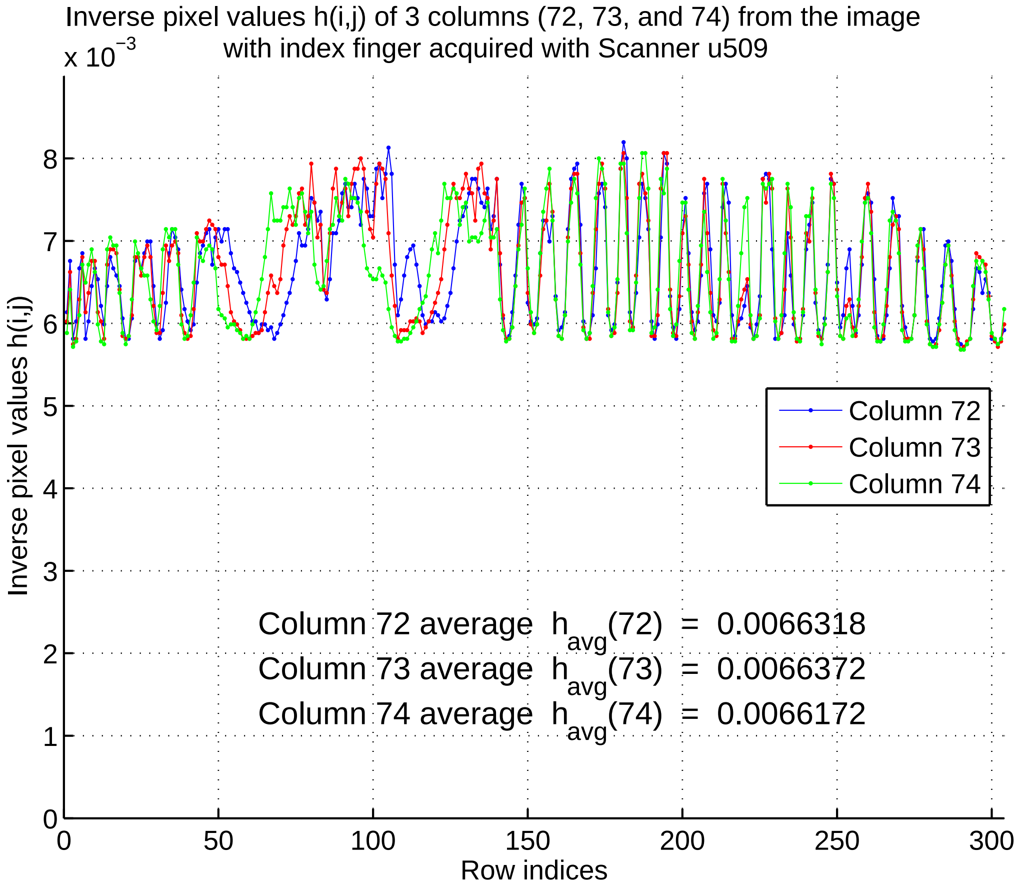


Figure 5.69: Swipe scanners: 3 columns in inverse mode and their averages

Next, as we know from the area scanners, the scanner pattern along rows (i.e., $s(j)$ in this case) has approximately Gaussian distribution, and consequently, its inverse $1/s(j)$ is also approximately Gaussian (see Section 6.5 in the appendix). Since $h_{avg}(j)$ in Expression 5.87 is slowly varying and the scanner pattern $s(j)$ and the (average) fingerprint pattern $f_{avg}(j)$ are independent, the problem of separating them from $h_{avg}(j)$ appears to be like separating a Gaussian noise $(1/s(j))$ from a (slowly) varying signal $(f_{avg}(j))$, which is exactly what we did in the advanced algorithms for the area scanners using a moving-average and adaptive Wiener filtering. And not surprisingly, both filters work here as well, which is the inverse mode of the algorithms we propose (described later).

Drawing from the experience with the area scanners, here we also tried these two filters in direct signal mode, i.e., without the inversion in Expression 5.86, and found that this also works; this is the direct mode of the algorithms we propose. The explanation why it works, however, proved much more difficult than in inverse mode. The main argument about that hinges on the observation that the pixel values $g(i,j)$ for the particular case of UPEK swipe scanners are approximately linearly dependent on the fingerprint pattern $f(i,j)$ in Expression 5.85. This (hypothetical) approximation is discussed in detail in Section 6.6 in the appendix. In summary:

$$g(i,j) \approx k(j)\,(f(i,j) - a) + b(j), \tag{5.88}$$

$$\text{where} \quad k(j) = -\frac{s^2(j)}{(1+s(j)a)^2} \quad \text{and} \quad b(j) = \frac{s(j)}{1+s(j)a}, \tag{5.89}$$

for a suitably chosen constant a (0.0025 in our case). By averaging $g(i,j)$ along columns, for the average row $g_{avg}(j)$, we receive the approximation:

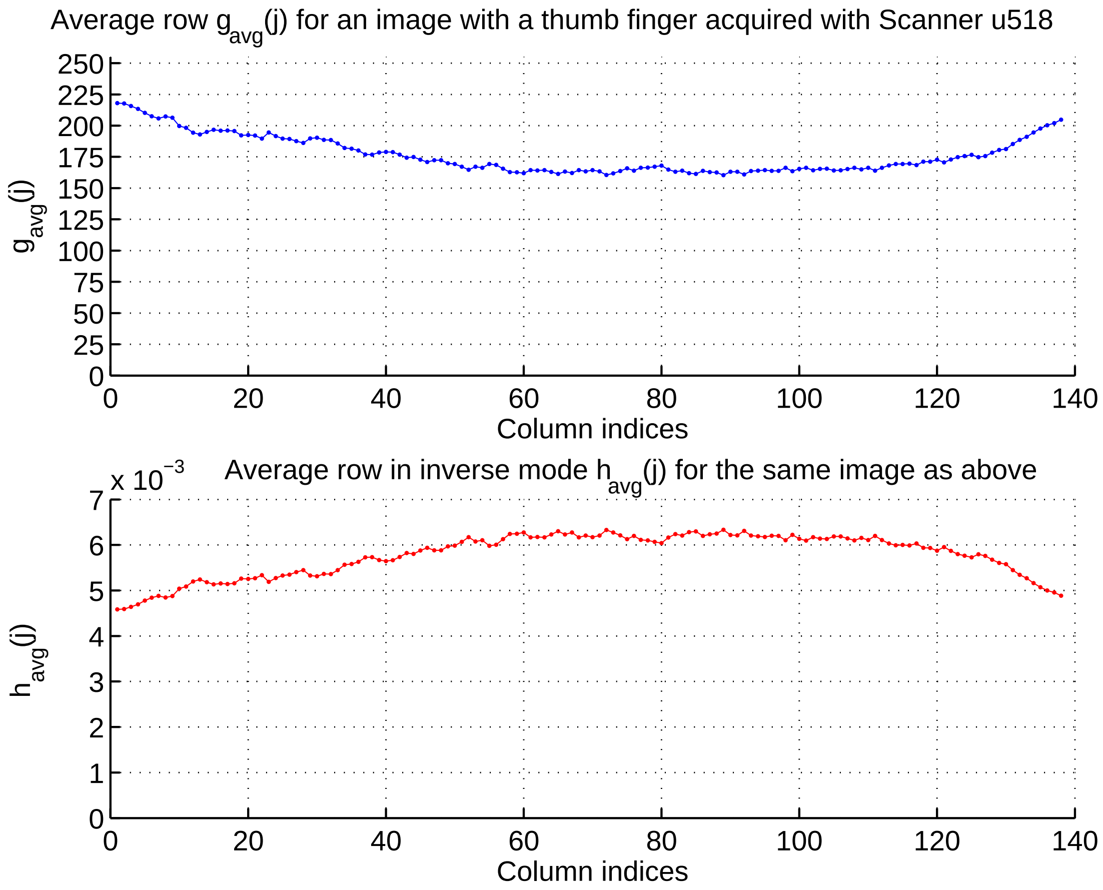
$$g_{avg}(j) \approx k(j)\,(f_{avg}(j) - a) + b(j). \tag{5.90}$$

because $k(j)$ and $b(j)$ do not depend on the row index i – they both are function only of the scanner pattern $s(j)$ (and the constant a).

Looking at Expression 5.87, we also observed that since $h_{avg}(j)$ does not change significantly for adjacent column indices j and that $1/s(j)$ does not change significantly either (because although it changes rapidly, the changes around its mean are small), then $f_{avg}(j)$ must also change very little for adjacent column indices j, i.e., $f_{avg}(j)$ is also a slowly varying function. Furthermore, as the number of rows is typically large (it is variable, but well over 100), the average $f_{avg}(j)$ is close to its mean, which we observed is about 0.0025, i.e., the constant a in Expression 5.89 (see Section 6.6 in the appendix for details). This implies that the term $(f_{avg}(j) - a)$ in Expression 5.90 is close to zero. So, our speculation is that $k(j)\,(f_{avg}(j) - a)$ is very small or at least slowly varying with j, which via the process of denoising, a filter can remove, producing the noise-like term $b(j)$ that is a function only of the scanner pattern $s(j)$. Therefore, processing $g_{avg}(j)$ with a moving-average or an adaptive Wiener filter will produce the scanner pattern (in some form), similarly to what filtering $h_{avg}(j)$ does.

Finally, we claim that (a) with the characteristics of our signals and (b) in the context of the denoising algorithms for swipe scanners we propose, the approximation $b(j) \approx const \cdot s(j)$, with *const* varying within only ±3%, is accurate. This essentially implies that $b(j)$ can be assumed as being the scanner pattern $s(j)$ it its direct form. Details about this approximation we also provide in Section 6.6 in the appendix.

Figure 5.70 shows an example average row in direct mode $g_{avg}(j)$ and in inverse mode $h_{avg}(j)$, computed from the same image, which has been acquired with a UPEK swipe scanner.


Average row $g_{avg}(j)$ for an image with a thumb finger acquired with Scanner u518

Average row in inverse mode $h_{avg}(j)$ for the same image as above

Figure 5.70: Swipe scanners: average row in direct and inverse modes

Next we describe the algorithms for swipe scanner authentication as disclosed in [Ivanov and Baras US'907]. Figure 5.71 shows a conceptual diagram of signal processing modules in which the signal **g**, the image, is processed to produce the signal d, the scanner verification decision, along with the interface signals among the modules. The signals between the subsequent modules represent only the main, not all, input and output signals of the modules. Similarly to the advanced algorithms for the area scanners, each signal processing module has different modes of operation (also called "implementations"). Next we discuss the processing of each module and their different modes of operation.

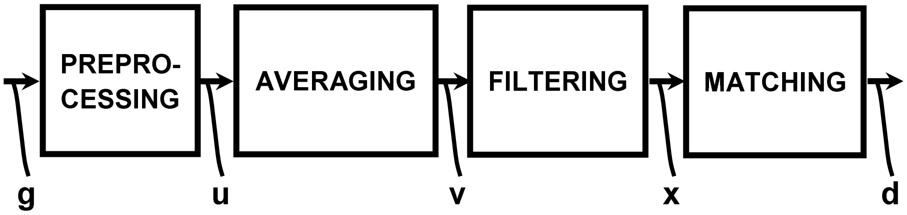


Figure 5.71: Conceptual diagram of operation of the modules for the swipe scanners

5.5.2.2 Preprocessing Module

The Preprocessing Module (see Figure 5.71) has the signal $g(i,j)$, i.e., the pixels of the image, as input and the signal **u**, a two-dimensional signal with the same size as **g**, as output. It has two modes of operation:

(a) Direct mode: $u(i,j) \;=\; g(i,j)$;

(b) Inverse mode: $u(i,j) \;=\; h(i,j) = \frac{1}{g(i,j)}$;

This mode implements the same signal inversion as discussed for area scanners (see Section 5.5).

5.5.2.3 Averaging Module

The Averaging Module (see Figure 5.71) computes the average values of the pixels along columns (or along rows, depending on the scanning direction of the line of sensor elements) from its input signal **u**, which represents the pixel values $u\,(i,j)$ of the image, and produces these average values as its output signal **v**. Thus, the input signal **u** is two dimensional, whereas the output signal **v** is one dimensional.

Let I be the total number of rows and J be the total number of columns in $\mathbf{g}$. Typically, the line of sensor elements in most scanners is perpendicular to the length of the finger, and therefore the finger is swept over the scanner in the direction of finger's length. In this case, the sequentially produced lines of pixels form rows in the two-dimensional image $\mathbf{g}$. Thus, the pixels in each column of $\mathbf{g}$ are produced by one and the same sensing element, i.e., for a given (and fixed) column j and for all row indices i from 1 through I, the pixels $g(i,j)$ are produced by the sensing element with index j in the line of J sensing elements. Alternatively, in scanners for which the orientation of the line of sensor elements is along the length of the finger, for a given (and fixed) row i and for all column indices j from 1 through J, the pixels $g(i,j)$ are produced by the sensing element with index i in the line of I sensing elements.

The Averaging Module computes the average of the pixel values produced by one and the same sensor element. Thus, for scanners in which the line of sensor elements is perpendicular to the length of the finger, the averaging is along columns and the output signal $\mathbf{v}$ is:

$$v\left(j\right) = \frac{1}{I}\sum_{i=1}^{I} g\left(i,j\right) \tag{5.91}$$

where j is from 1 to J. For scanners in which the orientation of the line of sensor elements is along the length of the finger, the averaging is along rows and the output signal $\mathbf{v}$ is:

$$v\left(i\right) = \frac{1}{J}\sum_{j=1}^{J} g\left(i,j\right) \tag{5.92}$$

where i is from 1 to I.

Some swipe scanners may employ more than one line of sensing elements, in which case the process of constructing a fingerprint image from the sequence of lines of pixels may involve sophisticated signal processing, which, unfortunately, is usually manufacturer proprietary.

5.5.2.4 Filtering Module

The Filtering Module (see Figure 5.71) filters the input signal $\mathbf{v}$, which is the output of the Averaging Module, to produce the output signal $\mathbf{x}$, which contains the scanner pattern. Similarly to the advanced algorithms for the area scanners, the Filtering Module here also essentially comprises two operations: (1) a smoothing operation $\mathcal{F}(.)$ that smooths $\mathbf{v}$ and (2) a subtraction operation that subtracts thus smoothed signal $\mathcal{F}(\mathbf{v})$ from $\mathbf{v}$, and produces the output signal $\mathbf{x}$:

$$\mathbf{x} = \mathbf{v} - \mathcal{F}(\mathbf{v}) \tag{5.93}$$

In this way, the smoothing also removes the (variable) mean of the scanner pattern and yields only the variable part of it.

Let N denote the number of elements of the input signal $\mathbf{v}$. In scanners for which the line of sensor elements is perpendicular to the length of the finger, $N = J$, whereas in scanners for which the orientation of the line of sensor elements is along the length of the finger, $N = I$.

From this point onwards, the processing is essentially the same as for the area scanners. The signal $\mathbf{v}$ here is similar to one vector $\mathbf{v_i}$ in the signal $\mathbf{v}$ in the advanced algorithms for the area scanners, or in other words, the signal $\mathbf{v}$ there can be thought as having only a single vector ($\mathbf{v_1}$), i.e., $c = 1$ in Expression 5.43 or Expression 5.44.

Similarly, the Filtering Module here can be implemented in two ways: with a low-pass filter and with an adaptive Wiener filter. Both implementations are essentially the same as the low-pass filter and the adaptive Wiener filter implementations of the advanced algorithms for the area scanners (see Section 5.5.1.3), except for some values of the parameters.

Because of the finite length of the input signal $\mathbf{v}$, the signal processing of the discontinuity at the beginning and at the end of $\mathbf{v}$ may lead to unwanted artifacts. The methods we propose here are the same as described in the advanced algorithms for the area scanners (see Section 5.5.1.3)): computation shortening, replica padding, and constant padding, although using other methods is also possible.

Incorporating such methods to avoid edge effect artifacts may seem unjustified, but actually it is quite important because the length N of $\mathbf{v}$ is relatively small (in the order of one to several hundreds) and such artifacts may affect the estimate of the scanner pattern of about 10 pixels, which is not negligible and may decrease the performance. Furthermore, because applying a fingertip tightly in the regions around the edges of the scanner platen area (and in the two ends of the line of sensor elements in this respect) is difficult, the pixels in these regions typically contain no fingerprint pattern. Hence, the estimate of the scanner pattern in these regions can be made very accurate if introduction of unwanted artifacts is avoided as specified above.

Another important aspect of the processing in this module is using a windowing function applied to the signal being processed. The windowing here is exactly the same as the windowing in the advanced algorithms for the area scanners and therefore will not be repeated.

Low-pass Filter Implementation of the Filtering Module

All discussions about the Low-pass Filter Implementation of the Filtering Module in the advanced algorithms for the area scanners apply here as well. The low-pass filter of preference here is also a (windowed) moving-average filter. Selecting M about 3 gives optimal results, but good overall performance is also achieved for M in the range from 2 to about 7.

For the pixels that are close to the beginning or the end of $\mathbf{v}$, the three techniques for computing the local mean $\mathbf{v}^{(\mathbf{lm})}$ proposed here are computation shortening, replica padding, and constant padding, which are also the same as for computing the local mean in the advanced algorithms for the area scanners. Using other techniques, however, is also possible.

Finally, the output signal $\mathbf{x}$ of the Filtering Module in this implementation is the difference between the input signal $\mathbf{v}$ and the local mean signal $\mathbf{v}^{(\mathbf{lm})}$:

$$x\left(k\right) = v\left(k\right) - v^{(lm)}\left(k\right), \tag{5.94}$$

where k is the current pixel index, an integer from 1 to N.

Figure 5.72 shows the input signals $\mathbf{v}$ and output signals $\mathbf{x}$ of this module for two images containing a thumb and a little finger and acquired with one and the same scanner, processed in direct mode with a moving-average filter, and the correlation coefficient between them. The lower subplot shows only the first half of the columns for better visibility.

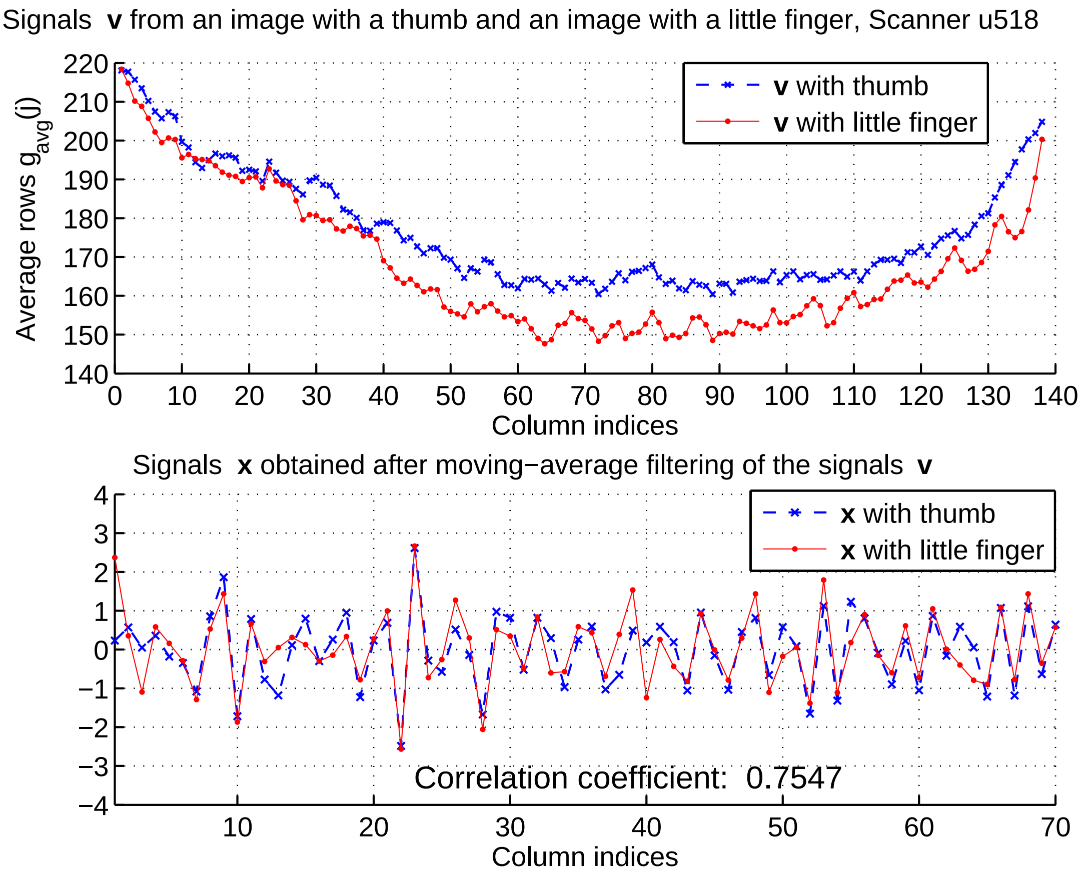


Figure 5.72: Swipe scanners: signals of the moving-average filtering in direct mode

Figure 5.73 shows the input signals **v** and output signals **x** of this module for the same images as in Figure 5.72, processed in inverse mode with a moving-average filter, and the correlation coefficient between them. The lower subplot shows only the first half of the columns for better visibility.

Adaptive Wiener Filter Implementation of the Filtering Module

All discussions about the Adaptive Wiener Filter Implementation of the Filtering Module in the advanced algorithms for the area scanners apply here as well, and the computation of the local mean $\mathbf{v}^{(\mathbf{lm})}$, local square $\mathbf{v}^{(\mathbf{ls})}$, local variance $\mathbf{v}^{(\mathbf{lv})}$,

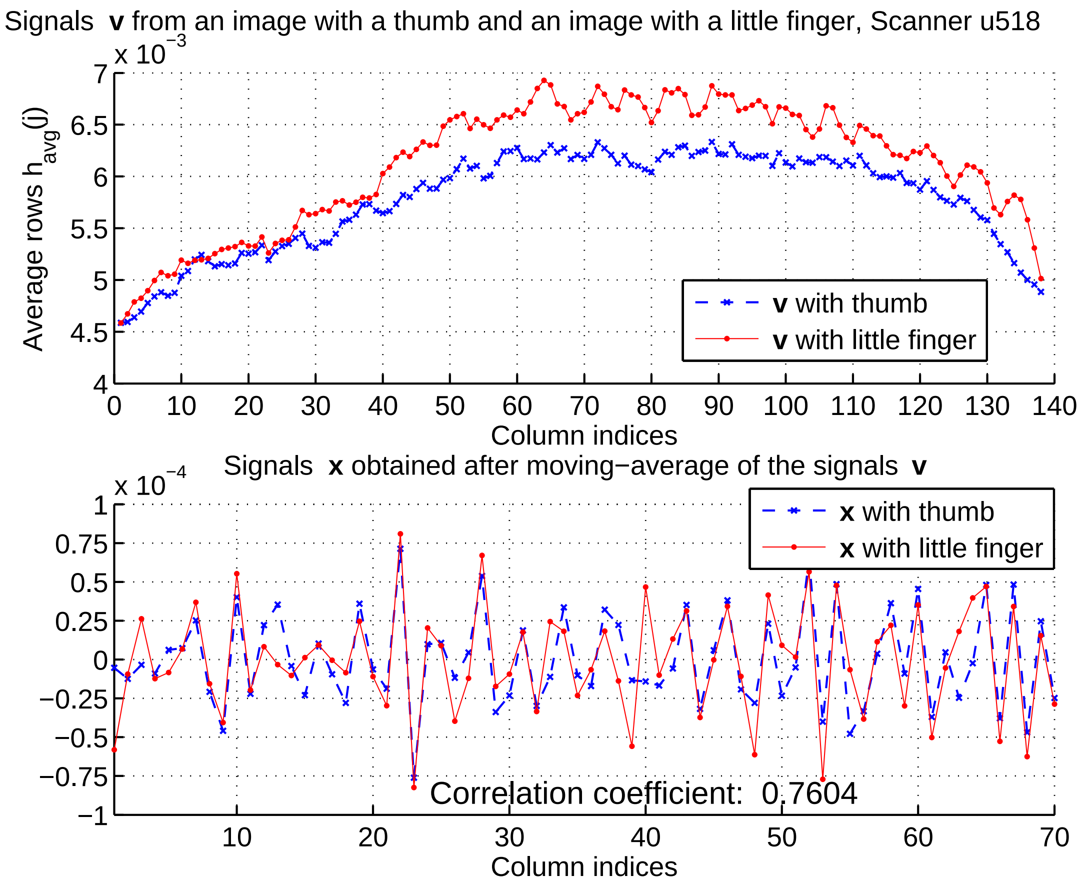


Figure 5.73: Swipe scanners: signals of the moving-average filtering in inverse mode

and scaling coefficient **d** vectors are exactly the same. Similarly, for computing both $\mathbf{v}^{(\mathbf{lm})}$ and $\mathbf{v}^{(\mathbf{ls})}$, selecting M about 3 gives optimal results, but good overall performance is also achieved for M in the range from 2 to about 7.

For the pixels that are close to the beginning or the end of **v**, the three techniques for computing the local mean and the local square vectors proposed here are: computation shortening, replica padding, and constant padding, which are also the same as for computing the local mean in the advanced algorithms for the area scanners. Using other techniques, however, is also possible.

Regarding the Wiener filter variance σ_w^2, here it is also a critically important parameter that determines the overall performance. The major difference with the area scanners is that here the effect of the scanner noise can be neglected as signal $\mathbf{v}$ is the average of the columns (or rows, depending on the line of sensing elements). Nevertheless, σ_w^2 is still a tradeoff parameter that needs to be determined after careful optimization and tests. When doing such optimization is not feasible, as a very approximate guideline, in direct signal mode, σ_w^2 can be set to the estimate for the scanner pattern variance σ_s^2. For Signal Model A, for the direct signal mode implementation of the Preprocessing Module, σ_w^2 can be set to about 30 and to about $3 * 10^{-8}$ for the inverse signal mode.

Computing the smoothed signal is also done in a similar way. For each pixel with index k, where k is from 1 to N, the smoothed signal vector $\mathbf{v}^{(\mathbf{s})}$ is computed by:

$$v^{(s)}(k) = v^{(lm)}(k) + d(k) \,.\, \left(v(k) - v^{(lm)}(k)\right). \tag{5.95}$$

Finally, the output signal $\mathbf{x}$ of the Filtering Module in this implementation is the difference between the input signal $\mathbf{v}$ and the smoothed signal $\mathbf{v}^{(\mathbf{s})}$, corrected with the Wiener mean μ_w:

$$x(k) = v(k) - v^{(s)}(k) + \mu_w$$

where k is the current pixel index, an integer from 1 to N. In the preferred implementation, the Wiener filter mean μ_w is set to 0, but other values of μ_w are also possible as μ_w can be used to compensate in case when fixed-valued offset is present so that the output signal $\mathbf{v}$ becomes zero mean.

Figure 5.74 shows the input signals $\mathbf{v}$ and output signals $\mathbf{x}$ of this module for the same images as in Figure 5.72, processed in direct mode with an adaptive Wiener filter, and the correlation coefficient between them. The lower subplot shows only the first half of the columns for better visibility.

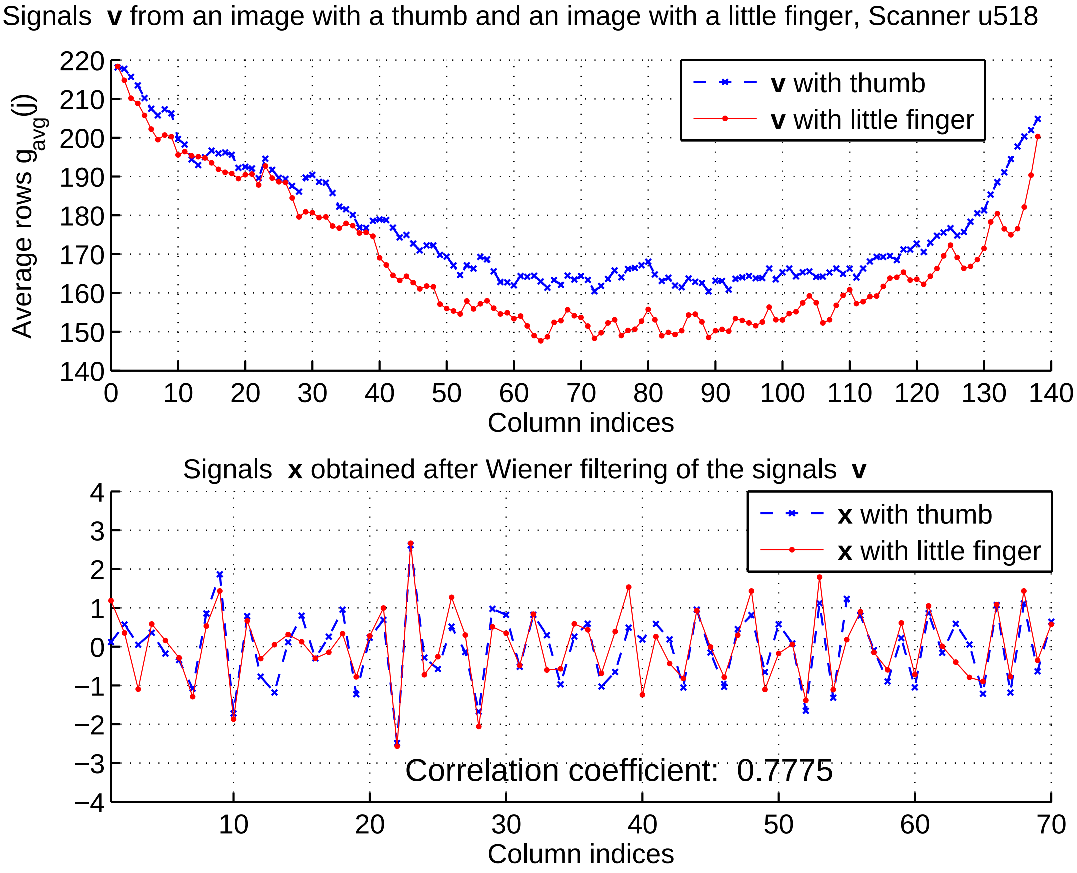


Figure 5.74: Swipe scanners: signals of the adaptive Wiener filtering in direct mode

Figure 5.75 shows the input signals **v** and output signals **x** of this module for the same images as in Figure 5.72, processed in inverse mode with an adaptive Wiener filter, and the correlation coefficient between them. The lower subplot shows only the first half of the columns for better visibility.

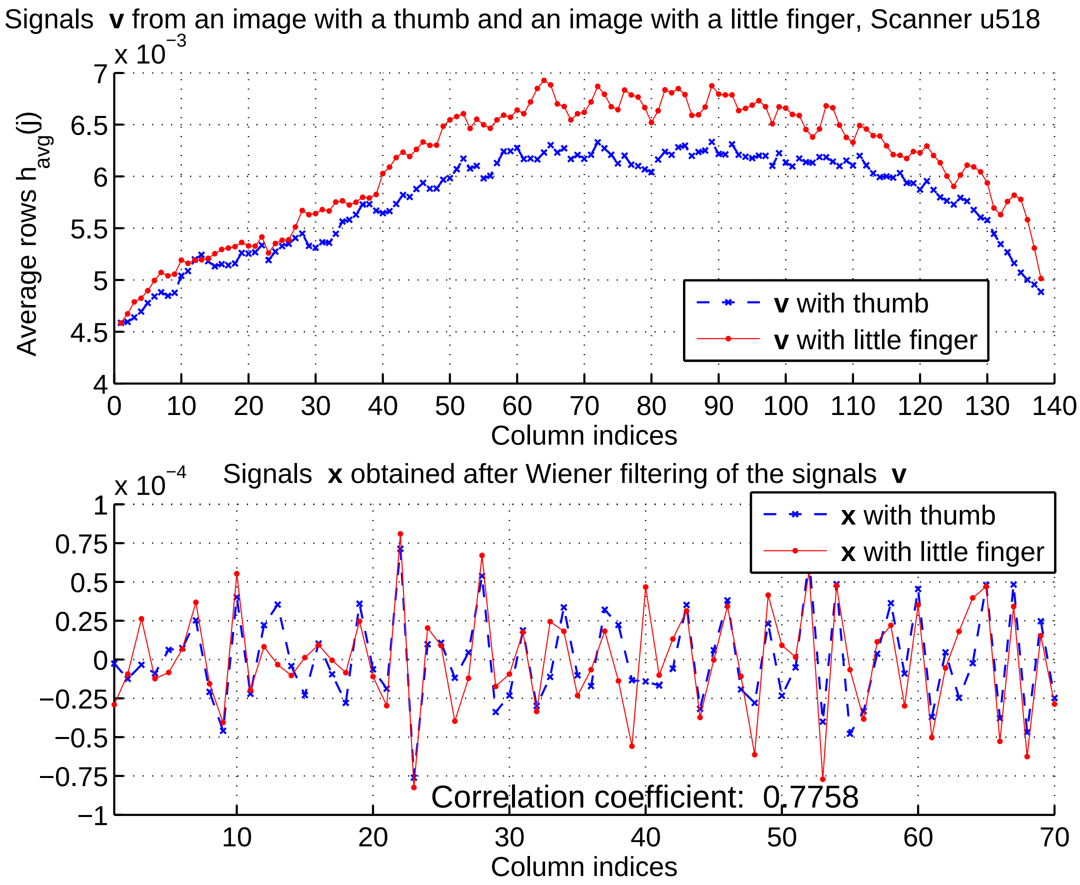


Figure 5.75: Swipe scanners: signals of the adaptive Wiener filtering in inverse mode

5.5.2.5 Matching Module

The Matching Module (see Figure 5.71) computes a similarity score between the scanner patterns extracted from two images and produces a decision as to whether they are sufficiently similar or not.

Let $\mathbf{x_e}$ denote the output signal of the Filtering Module when the input signal $\mathbf{g}$ is an image acquired during the scanner enrolment and $\mathbf{x_q}$ denote the output signal of the Filtering Module when the input signal $\mathbf{g}$ is an image acquired during the scanner verification. Using the signals $\mathbf{x_e}$ and $\mathbf{x_q}$ the Matching Module:

(i) quantifies the similarity between the two signals $\mathbf{x_e}$ and $\mathbf{x_q}$ in a score;

(ii) produces a decision via the output signal d as to whether the two images have been acquired with the same fingerprint scanner by comparing this score with a threshold. When the output signal d is set to 1, this indicates scanner match; when it is set to 0, this indicates scanner non-match; and when it is set to (-1), this indicates that a decision on matching/non-matching cannot be made and a new query image must be acquired.

Quantifying the similarity between the two signals $\mathbf{x_e}$ and $\mathbf{x_q}$ in a score can be done with the same three implementations as described in the advanced algorithms for the area scanners: normalized correlation, correlation coefficient, and relative mean-square error (see Section 5.5.1.5); however, other implementations are also possible. The only difference is that set $\mathbf{D}$ in the Matching Module specification of the advanced algorithms for the area scanners in Section 5.5.1.5 here comprises all indices k from 1 to N (and therefore $N_D = N$) because no masking is done. All other formulas and discussions are exactly the same.

5.5.2.6 Using Multiple Images

Similarly to the advanced algorithms for the area scanners, all implementations described above can use a single image for the scanner enrolment and a single image for the scanner authentication, and this is the preferred case. However, here it is also possible to use multiple images for the scanner enrolment and/or multiple images for the scanner verification. This may improve the overall accuracy of the scanner authentication.

The two implementations here are the same as those described in the advanced algorithms for the area scanners (see Section 5.5.1.6). The first method, averaging the similarity scores (Expression 5.82), is exactly the same, while the second one, computing an "average" enrolled scanner pattern is slightly different as there is no masking here, and therefore, $y_a(k) = 1$ for k from 1 to N. Hence, the computation of the "average" scanner pattern $\mathbf{x_a}$ in Expression 5.84 gets reduced to:

$$x_a(k) = \frac{1}{E} \sum_{r=1}^{E} x_r(k) \tag{5.96}$$

where k is an integer running from 1 to N. The rest of the computation of the similarity score is the same.

5.5.2.7 Image cropping

Some swipe scanners produce images that contain rows and/or columns of pixels with constant, "dummy," values around the area of the actual fingerprint pattern. Since the pixels with these constant values carry no information about the scanner pattern (and about the fingerprint pattern either), these pixels have to be detected and removed from the image before it is processed further, i.e., the image has to be cropped. These areas of constant values typically surround the useful area as rectangular pads on the top and bottom and on the left and the right of the fingerprint pattern.

One method for finding if a row falls within such a pad is by computing the absolute values of the differences between each two adjacent pixels along this particular row. If the maximum of these absolute values is 0, this indicates that all pixel values in the row are constant and thus this row of pixels has to be removed. Finding if a column falls within one of these pads can be done in the same way. This test is then applied to each row and each column in the image and the detected rows and columns are removed. The image cropping can be done before or after the Preprocessing Module.

5.5.2.8 Combinations of Module Modes for the Swipe Scanners

All modes (implementations) of the {Filtering Module, Matching Module} can be used in combination with any of the modes of the modules that precede this current module in the conceptual signal flow diagram shown in Figure 5.71. However, different combinations of modes may provide different overall performance. One well-performing combination is described next.

Figure 5.76 shows the flowchart of one exemplary implementation using a single enrolled image $\mathbf{g_e}$, acquired and processed during the scanner enrolment, and a single query image $\mathbf{g_q}$, acquired and processed during the scanner verification. Although $\mathbf{g_e}$ and $\mathbf{g_q}$ are processed at different times, the consecutive processing steps are identical, and therefore we discuss them simultaneously. $\mathbf{g_e}$ and $\mathbf{g_q}$ are first processed by the Preprocessing Module in its direct signal mode. The output signals $\mathbf{u_e}$ and $\mathbf{u_q}$ are processed by the Averaging Module. The resulting signals $\mathbf{v_e}$ and $\mathbf{v_q}$ are processed by the Filtering Module with a moving-average filter. Finally, the Matching Module computes the correlation coefficient and outputs the signal d, based on which a decision for scanner match or scanner nonmatch is made.

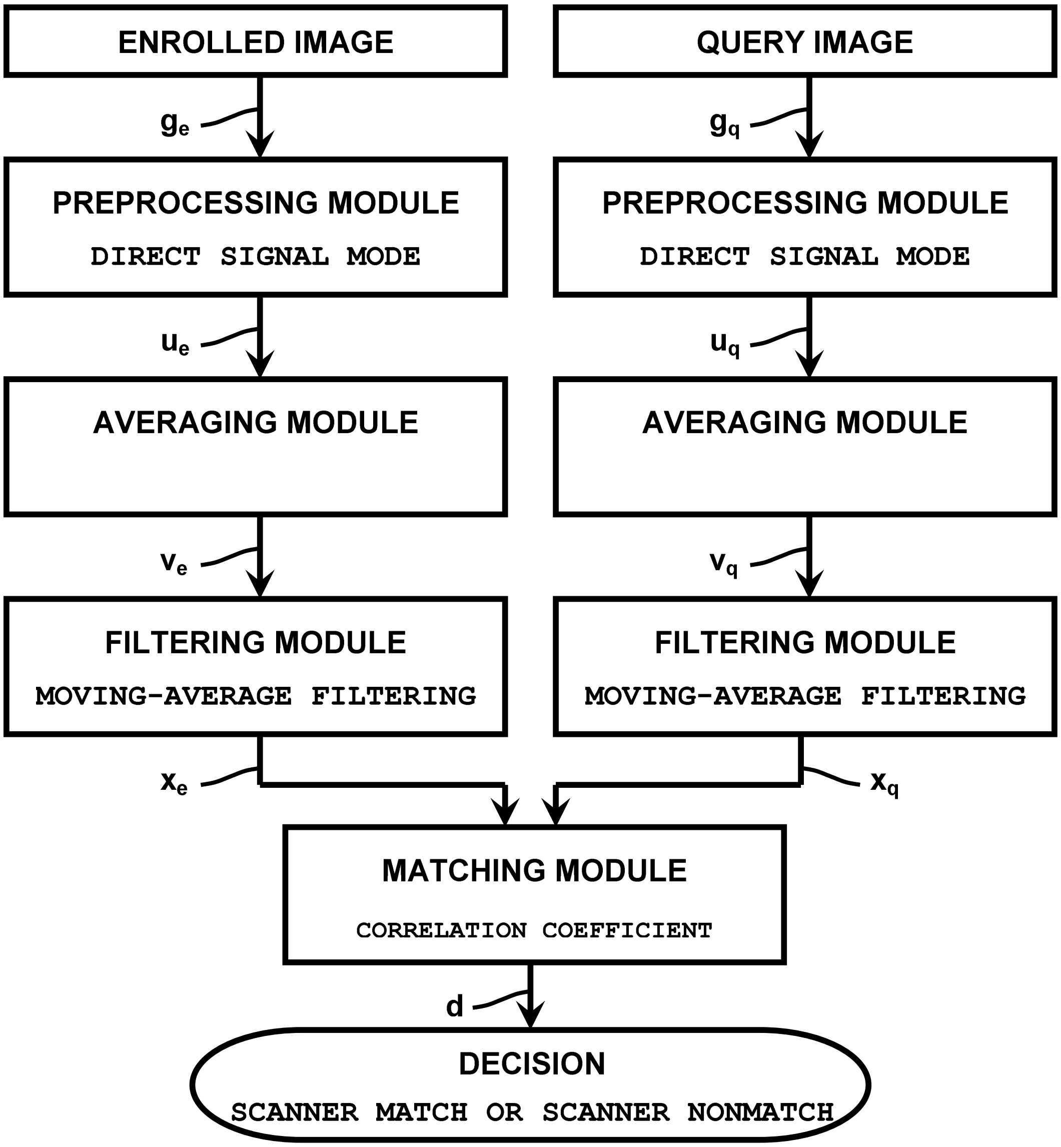


Figure 5.76: Advanced algorithms for swipe scanners: An exemplary implementation with moving-average filtering

5.5.2.9 Performance

Although the advanced algorithms for the swipe scanners offer fewer choices of modes and fewer parameters for optimization, a system designer can still trade performance for computational power and complexity.

For illustrative purposes, here we present the performance of the exemplary implementation shown in Figure 5.76. It uses direct signal mode, 3-tap moving-average filtering with replica-symmetric padding, and correlation coefficient for matching. The normalized histograms (integrating to 1) of the correlation coefficients from running this implementation on 5,400 images acquired (at room temperature) with the 27 UPEK swipe scanners (see Section 6.1 in the appendix) for 10 images per finger for all 10 fingers of 2 persons are shown in Figure 5.66. Only a single image is used for scanner enrolment and only a single image for scanner verification. Every image is matched against all other images. The total number of matchings is about 15 million (here, again, only one of the matchings, AB or BA, is computed since all processing is again completely symmetric for the enrolled image and the query image).

Here, however, in contrast with case with the area scanners, the two distributions of correlation coefficients visibly overlap. Although both the mean and the standard deviation of the self correlation coefficients here are very close to the mean and the standard deviation of the self correlation coefficients for the exemplary implementation of the advanced algorithms for the area scanners (see Figures 5.66 and 5.67), the mean and the standard deviation of the cross correlation coefficients here

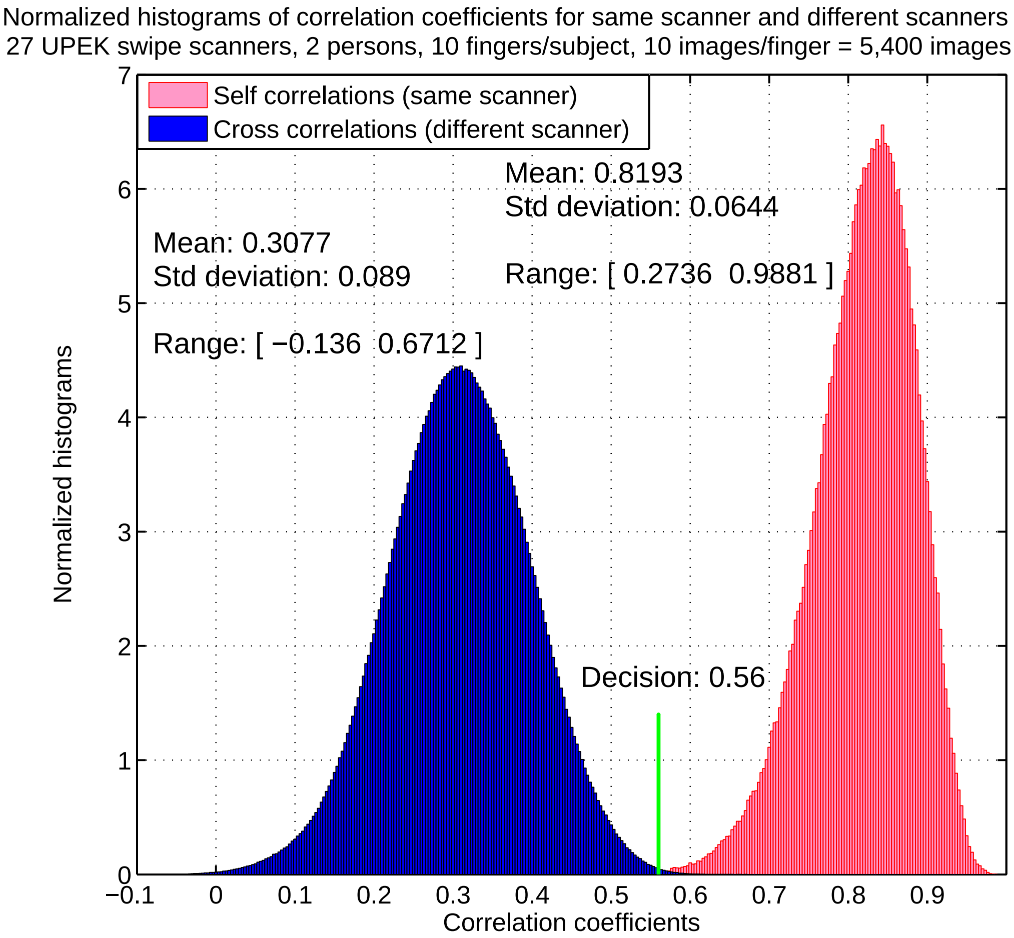


Figure 5.77: Performance of the advanced algorithms for the swipe scanners implemented with moving-average filtering

are considerably different: the mean is about 0.30, versus about 0.05 for the area scanners, and the standard deviation is about twice larger: about 0.09, versus about 0.045 for the area scanners. We think that the most probable reasons for this are:

1. The very limited number of pixels for which the scanner patter is extracted, upper bounded to about 140 pixels for the swipe scanners because only a single row of sensor elements produces all pixels in the image. Another possible reason that may compound the problem is the stronger correlation within the

scanner pattern $s(j)$; and

2. A possible suboptimal performance of the filters used.

With the decision threshold we chose (0.56), we achieve roughly an empirical EER of $1 \cdot 10^{-3}$: the empirical FAR is $9.25 \cdot 10^{-4}$ and the empirical FRR is $9.66 \cdot 10^{-4}$.

5.5.3 Physical Implementations of the Advanced Algorithms

Figure 5.78 illustrates a typical system that uses biometric information. The system may include one or more processors, e.g., general-purpose microprocessors, signal processors, or microcontrollers, along with the associated ROM and RAM memories of various types and also storage devices (not shown). The system may also have additional digital hardware, such as programmable arrays (FPGA, CPLD, PLA, etc.), ASICs, or any other type of hardware that can perform computations and process signals. The system may further include I/O interfaces and communication interfaces that connect it to various networks (wired and wireless). For increased security, the system may also have Trusted Computing functionality and for example, be equipped with a TPM that can provide complete control over the software that is running and that can be run in it. Finally, the fingerprint scanner may be connected to the system directly or via a network; it can also be part of the system.

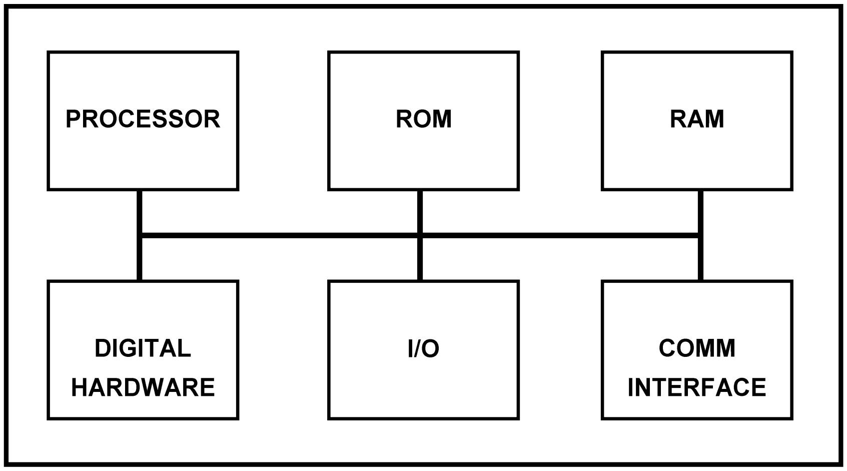


Figure 5.78: System block diagram

The advanced algorithms can be implemented entirely as a software program that runs on the processor(s), entirely in the digital hardware, or some modules can be implemented as software programs and some modules be implemented in the digital hardware. For example, the Selection Module and the Masking Module can be implemented in the digital hardware, while the Preprocessing Module, the Filtering Module, and the Matching Module can be implemented in software.

5.5.4 Features of the Advanced Algorithms

The results we achieved exceeded our expectations. Here we list some of features of the advanced algorithms.

- **Unmatched accuracy.** To best of our knowledge, our algorithms are the first and the only that provide error rates way well below 1% **and** for scanners of exactly the same type, manufacturer, and model.

It is important to note that we do not claim that the performance (as accuracy) of the exemplary implementations we showed earlier is the best one possible the advanced algorithms can deliver; rather, it is just an example for their potential. The advanced algorithms as we define them, both for area and for swipe scanners, should be considered as a set of tools for achieving the purpose of scanner authentication. Therefore, the modules and their modes to be implemented in a particular target application should be chosen and their parameters optimized once the specific application requirements and constraints are set (see below).

- **Computational efficiency.** We developed the advanced algorithms with the specific objective for computational efficiency, with simplicity being a central objective for the algorithm design. This led to straightforward and extremely computationally efficient algorithms.

The main reason for this efficiency is the one-dimensional signal processing of the advanced algorithms and their simplicity. In the Filtering Module, it is not necessary to compute conventional convolution as the moving-average filter scales the adjacent pixel values in a window with a constant, which in case of a fixed-point implementation can be speed-optimized by using a window with size that is a power of 2 (e.g., 2, 4, or 8) so that the division in the moving average is reduced to a shift-right (by 1, 2, 3, etc.) processor instruction or a shift-right hardware operation. In this respect, the adaptive Wiener filter needs more computations as the local variance requires raising to the power 2 and explicit division (whenever the scaling coefficient is nonzero). When using the variance masking mode (for the low-pass filter), the Masking Module similarly computes the local variance (to compare it with the two thresholds).

The second reason is the absence of any transforms from one domain into another (like the Fourier transform) – all processing is done directly and only in the base domain (which is space). These two reasons result in a linear dependence between the number of computations needed and the number of pixels used.

- **Scalability.** Each module has different modes of operation, allowing granularity with varying degrees of complexity depending on the computational and the time constraints. Furthermore, the algorithms for both the swipe and the area scanners (to much greater extent) provide higher accuracy when more pixels are used. For the area scanners, good accuracy can be achieved by using even only 4% of the pixels of an image (=10 columns out of 256).

- **Robustness.** Another advantage of the advanced algorithms is their robustness under a wide variety of conditions – environmental changes (temperature, moisture, dirt/grease, etc.), different fingertip pressure on the scanner platen, changes in the fingertip skin (scratches, wear, etc.). For example, the algorithms are specifically developed to work in a range of temperatures and in significant presence of water on the fingertip (not only the ordinary skin moisture). The problem arising from pressing the fingertip very strongly to the scanner platen is also handled particularly well.

In addition to working with images containing patterns of two completely different fingers, both as patterns and as types of fingers, the algorithms work properly even when not a fingertip, but another body part is applied to the scanner, e.g. a palm, because the algorithms inherently do not rely on the fingerprint pattern to have specific characteristics – as long as the object applied to the scanner platen does not adhere tightly and completely to the scanner platen, but leaves small air gaps, the algorithms work properly.

Finally, the parameters of the signal processing modules and their modes can vary in wide ranges.

- **Stability.** All modules and modes are unconditionally stable as there are no feedback loops in any form and at any level.

- **Fixed-point implementations.** Because of their simple design, the advanced algorithms can tolerate round-off effects (errors) due to finite-length effects in the parameter, coefficient, and signal quantization. The dynamic range of the input signal (the pixels of the image) is standardized and limited to 8 bits/pixel, which greatly facilitates the design of the scaling coefficients between the subsequent stages of the processing (both in a microprocessor and in a dedicated computational hardware). All processing revolves around computing moving-average sums, which cannot create overshoots in the intermediate signals, and scaling with bounded (even to a great extent predetermined) numbers. The computation of the correlation and its variant as well as the relative mean-square error, as to the fact that they involve multiplication and accumulation of two signals, can be arranged to operate with the current indices and thus the current sum to never exceed the dynamic range of the finite-precision arithmetic unit. The rest of the processing is indexing and comparisons.

The advanced algorithms do not use transforms from one domain into another domain, which are typically susceptible to numerical problems due to the finite word length. The signal inversion mode may, however, require care in implementing most of the modules in fixed-point systems because of possible scaling and roundoff errors. The inversion may also create other types of nonlinear effects, which we will study once we implement the algorithms in a real system.

Finally, our algorithms even have an edge over algorithms developed in the future that require floating point computations and use software libraries for this because they will probably require more time and consume additional energy.

- **Deployment in existing systems.** As the algorithms do not require changes in the fingerprint scanner, they can be implemented in systems that have already been manufactured and even sold to customers by upgrading their system software, firmware, and/or hardware (if using programmable hardware), which can be done even online. The methods for identifying devices by designing special hardware, in particular analog and/or digital circuits, typically incur material and manufacturing cost and are not applicable to systems (including fingerprint scanners) that have already been manufactured.

- **Improvements and extra functionality.** Since only a single image is needed for the scanner enrolment (and also for the scanner verification), the advanced algorithms for the area scanners can offer additional functionality. The decision whether to be adopted in practice or not and the exact implementations of this additional functionality, however, is at discretion of the system integrator because it is very application and system specific and generally involves a tradeoff between the reliability (accuracy) of the scanner authentication and the security of the scheme as a whole.

One of the possible ways to increase the accuracy is using more than one image (e.g., 3 or even 5) for scanner enrolment and/or scanner verification, which (for both area and swipe scanners) has been described in the corresponding sections.

Another way to increase the accuracy and reliability of the scanner authentication is to reduce it Scenario A as in Scenario A, the system can compute a very good estimate of the scanner pattern by using an image (or even several images) with air, both for scanner enrolment and for scanner verification. In either step (enrolment or verification), the system can acquire image(s) with air shortly before, shortly after, or both shortly before and shortly after acquiring the image(s) with a fingerprint. The user does not need to know about these additionally acquired images with air or do anything specific because these images are not related to his or her fingerprint – the system simply measures its components. Once images with both air and fingerprint are available, the system can (1) estimate the scanner pattern from the images with air and (2) perform Scenario B–type scanner authentication between this scanner pattern estimate and the image(s) with fingerprint, in this way verifying that the scanner has not been changed in the meantime. If this is confirmed, the system can then store this scanner pattern estimate as a reference (if this happens during the scanner enrolment) or use it to match it against already stored scanner pattern estimate (if this happens during the scanner verification). We will leave the work-out of the implementation details of this method to the system integrator and will only point out that regardless of the small time period between the acquisition of images with air and with a fingerprint, the security of this scheme is somewhat lower than that of Scenario C; it will also require additional images. Nevertheless, the gained accuracy and reliability of the scanner authentication may make it preferred in certain applications.

An extension of this idea can be a technique that ensures continuous trustworthiness of the hardware components. The system can acquire images with air periodically and/or at system boot time, then process them and perform a tacit scanner authentication. In order to further increase the accuracy of the scanner pattern estimate and the authentication in general, the system may acquire tens, even hundreds, of images with air, which generally may take only a couple of minutes when the system is otherwise idle.

Another extra functionality is the automatic scanner re-enrolment. After every successful scanner verification, the query scanner pattern can be stored as a new enrolled (reference) scanner pattern because it is already extracted and needs only to be stored. This can allow the system to adapt to potential long-term changes of the scanner pattern of the legitimate scanner.

The automatic scanner re-enrolment, however, has a downside that may result in a serious security flaw. Due to environmental changes or other factors, the scanner authentication may make a false accept decision, which in this case will automatically enrol another, nonauthentic scanner. Nevertheless, the scanner re-enrolment still has a merit as it can be done not automatically, but on purpose, explicitly and not every time the user authenticates. Because of the features of the advanced algorithms, this can be done very easily and the scanner pattern information on record can be updated without additional processing and without even being noticed by the user.

Regardless of the downsides of the scanner re-enrolment, its modification can still be used to improve the security. After (every) successful scanner verification, the scanner pattern of a new set of columns (or rows) can be computed and the scanner re-enrolled. These new columns or rows can be randomly selected; the algorithm can even change from columns to rows (or vice versa) or to combinations of both. This will improve the security as it will add a random element that an attacker will be impossible to predict when trying to construct images with embedded counterfeit scanner pattern. This method can be also combined with the other techniques described above.

In conclusion, to emphasize how practical and easy to integrate our technology is, we developed the software for select implementations of the advanced algorithms for both the area and the swipe scanners of UPEK into a fully operational (and with a graphical interface) demonstration prototype. It uses the UPEK development kit (see Figure 6.3) and the available 22 area and 27 swipe scanners (see Section 6.1 in the appendix).

Chapter 6

Appendix

6.1 Fingerprint Scanners Used

In this section we list and provide details about the scanners used during the different stages of research.

6.1.1 UPEK Scanners

We used three groups of scanners equipped with UPEK sensors, all of which capacitive. The first group, which we used only during the initial research, contains 4 area scanners as follows:

- Two scanners equipped with UPEK area TCS1 sensors; we refer to them as *Scanner 1* and *Scanner 2*. The TCS1 sensors (see the sensor on the left side of Figure 6.1) are the only capacitive (and generally solid-state) fingerprint sensors that are FIPS-201 certified.

Figure 6.1: UPEK area TCS1 sensor (left) and TCS2 sensor (right)

- Two scanners equipped with UPEK area TCS2 sensors (see the sensor on the right side of Figure 6.1); we refer to them as *Scanner 3* and *Scanner 4*.

Each sensor is connected to a third-party hardware module (containing a processor running firmware) and the hardware module is connected to an adapter, which in turn is connected to a computer via the RS 232 port (the configuration is shown in Figure 6.2). A computer software program is used to acquire and save the images as files. This set of hardware and software we purchased as an off-the-shelf product developed by MB FingerMetrica. Our analysis of the images acquired with it, however, showed that the post-processing done by these third-party components introduces additional artifacts in the raw images, which artifacts, however, did not hinder our study of the acquisition process and were not critically important for developing the initial algorithms; therefore, we did not correct or modify the affected images to compensate for these artifacts.

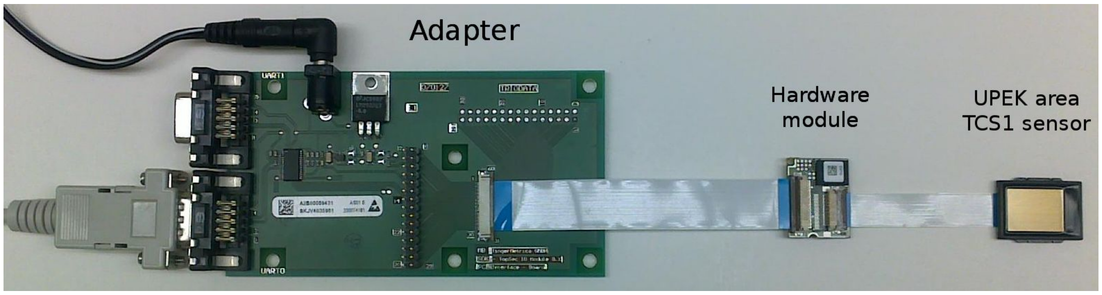


Figure 6.2: MB FingerMetrica kit with a UPEK area TCS1 sensor

The advanced algorithms we developed using the second and the third groups of scanners. Both groups consist of a development kit that connects to a computer via a USB interface (see Figure 6.3). The fingerprint sensor is mounted on a sensor module which, in turn, is connected to the development kit.

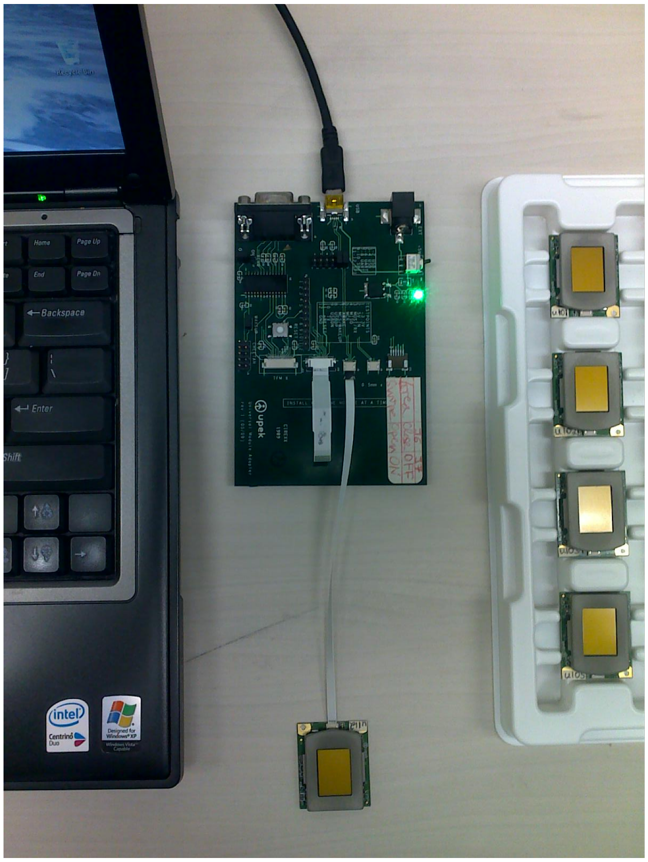


Figure 6.3: UPEK development kit with a UPEK area TCEFC1 sensor module

In the second group, each sensor module is TCEFC1 (see Figure 6.4, left) and contains a UPEK area TCS1 sensor (see Figure 6.1). Initially, we purchased 22 such TCEFC1 sensor modules, labeled as *Scanner u101* through *Scanner u122*, which is safe assume that were manufactured in one batch. The two TCEFC1 sensor modules that came included in the UPEK development kit (purchased separately), we labeled as *Scanner u151* and *Scanner u152*; they possibly do not come from the same batch as Scanners u101 through u122. Since during the study, three of the sensor modules (Scanners u104, u109, and u123) became damaged and unsuitable for further image acquisition and tests, later we purchased additionally three TCEFC1 sensor modules, labeled as *Scanner u131* through *Scanner u133*, which, again, possibly do not come from the same batch as Scanners u101 through u122 and Scanners u151 and u152. However, all these sensor modules are exactly of the same model and the sensors in them are exactly of the same model as well.

Figure 6.4: UPEK area TCEFC1 (left) and swipe TCESC4K (right) sensor modules

In the third group, each sensor module is TCESC4K (see Figure 6.4, right) and contains a UPEK swipe TCS4K sensor. We purchased 25 such TCESC4K sensor modules, labeled as *Scanner u501* through *Scanner u525*, which is safe to assume that were manufactured in one batch. The two TCESC4K sensor modules that came included in the UPEK development kit we labeled as *Scanner u551* and *Scanner u552*; they possibly are not from the same batch as Scanners u501 through u525. However, all these sensor modules are exactly of the same model and the sensors in them are exactly of the same model.

Regarding the software, besides the firmware, running in the sensor modules, to which we have no access, UPEK provides software libraries (via APIs) that allow acquiring images with their sensors. We used one of these libraries and developed a software tool that acquires raw images and saves them in BMP format. After developing the advanced algorithms, we extended the functionality of this software tool into a demonstration software that performs scanner authentication, both for the area and swipe scanners of UPEK, in real time.

AuthenTec acquired UPEK in September 2010, after which the are sensors with the UPEK technology are known as AuthenTec TouchChip® sensors. The UPEK swipe sensors were also integrated into the AuthenTec swipe sensors product line.

6.1.2 Veridicom Scanner

About a decade ago, the Veridicom fingerprint sensors were one of the first widely sold capacitive sensors and were integrated in many systems still used today. However, their sensitivity was quite limited, and since their technology apparently have not improved over the years, it lagged behind that of their competitors. Veridicom eventually sold the technology to Fujitsu, for which reason these sensors are also called Veridicom/Fujitsu. Fujitsu continued selling essentially the same Veridicom sensors until 2007, when Fujitsu discontinued their fingerprint products. For this reason, we could not buy scanners with recent Veridicom sensors. Instead, we used a USB reader (connected via the USB directly to the computer) equipped with a Veridicom sensor model FPS200 from around the year 2000, which sensor Fujitsu later sold as MBF200. This USB reader with the FPS200 sensor was used in our research, and in the thesis, it is referred to as *Scanner 5* (see Figure 6.5). The scanner was directly connected to the USB port of a computer, and the image acquisition was done by the native Veridicom software.

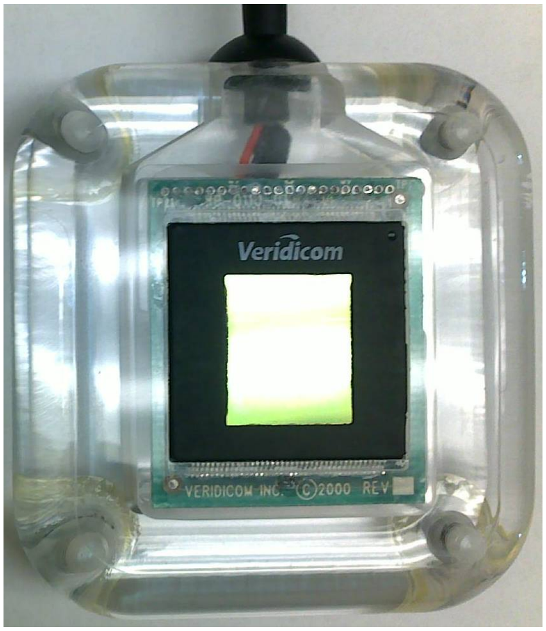


Figure 6.5: Veridicom reader with FPS200 sensor

6.2 Thermal Tests

Once we observed that the scanner pattern depends significantly on the temperature, we created a setup for thermal tests. We used a laboratory thermal plate that can both heat up and cool down. The fingerprint scanner to be tested is placed on top of this plate and heated up or cooled down. The temperature of the plate can be controlled precisely (within 1 ^{o}C) via a thermal controller, but because of the thermal resistance of the scanner hardware from the back of the scanner to the surface of its platen where the array sensor is, the temperature of the scanner platen is substantially different from that of the thermal plate (can reach over 10 ^{o}C). For this reason, we placed a separate temperature sensor (RTD sensor) on the scanner platen and connected it to a multimeter, allowing us to have an accurate estimate of the temperature of the fingerprint sensor array.

Nevertheless, however accurate the temperature measurement was, it was very difficult have a precise control over the scanner platen temperature. The first obvious reason for this is that once placing the fingerprint on the scanner platen for image acquisition, the scanner platen temperature changes uncontrollably, possibly with several degrees. The second problem is that maintaining precise temperature outside of a closed, dedicated temperature chamber (i.e., in open air and hence unavoidable air flows) is virtually impossible. It is clear, however, that a closed temperature chamber was not an option for us as we had to place a fingertip on top of the scanner. The third problem was that we could not achieve very low temperatures (around 0 ^{o}C as initially planned) because of the large temperature resistance of the sensors and the limited cooling capacity of the thermal plate. Thus, even with a plate temperature of -7 ^{o}C, the temperature on the scanner platen was about 5 ^{o}C, considerably far away from the freezing point of water.

A separate problem when cooling down to low temperatures, even well above 0 ^{o}C, is the air moisture: the water in the air condenses and covers the scanner platen, effectively making the fingertip very wet. When no fingertip is present, the scanner platen literally gets covered by a layer of water, and, consequently, acquiring images with air, as the ideal predetermined object, was practically impossible – all images acquired with “air” at low temperatures are heavily contaminated with water. The condensing water required constant cleaning of the scanner platen which, together with the difficulty of setting and maintaining the target temperature, took enormous amount of time and made the images acquired in such conditions very far from the desired ones. And finally, because the water in the air condensed also on the internals of the scanners, sometimes it created electric short circuits, most of which temporary, but some permanent, irreversably damaging a couple of scanners.

Despite all problems, we succeeded to acquire images, both with and without fingertips, at two temperatures of the scanner platen: at about 5 ^{o}C and at about 40 ^{o}C. Since the temperature on the scanner platen could not be controlled tightly, estimating the scanner pattern by taking repetitive images with air was impossible.

The images acquired outside the thermal tests were at room temperature: about 23 to 25 ^{o}C.

6.3 Initial Work

Query scanner ⟶	**Scnr 1**	**Scnr 2**	**Scnr 3**	**Scnr 4**	**Scnr 5**
Enrolled scanner and indices of the files used for enrolment ⇓					
Scanner 1 with parameters 55/0.20	***FRR, %***	FAR, %	FAR, %	FAR, %	FAR, %
1, 2, and 3	***0***	0	0	0	0
4, 5, and 6	***0***	0	0	0	0
10, 11, and 12	***0***	0	0	0	0
28, 29, and 30	***0***	0	0	0	0
Scanner 2 with parameters 55/0.20	FAR, %	***FRR, %***	FAR, %	FAR, %	FAR, %
1, 2, and 3	0	***0***	0	0	0
4, 5, and 6	0	***0***	0	0	0
10, 11, and 12	0	***0***	0	0	0
28, 29, and 30	0	***0***	0	0	0
Scanner 3 with parameters 30/0.20	FAR, %	FAR, %	***FRR, %***	FAR, %	FAR, %
1, 2, and 3	0	0	***0***	0	0
4, 5, and 6	0	0	***0***	0	0
10, 11, and 12	0	0	***0***	0	0
28, 29, and 30	0	0	***0***	0	0
Scanner 4 with parameters 30/0.20	FAR, %	FAR, %	FAR, %	***FRR, %***	FAR, %
1, 2, and 3	0	0	0	***1.3***	1.0
4, 5, and 6	0	0	0	***1.7***	0.3
10, 11, and 12	0	0	0	***2.0***	1.3
28, 29, and 30	0	0	0	***2.0***	1.3
Scanner 5 with parameters 18/0.25	FAR, %	FAR, %	FAR, %	FAR, %	***FRR, %***
1, 2, and 3	0	0	0	0	***0***
4, 5, and 6	0	0	0	0	***0***
10, 11, and 12	0	0	0	0	***0***
28, 29, and 30	0	0	0	0	***0***

Table 6.1: Scenario B results: FAR and FRR for all five capacitive scanners

Query scanner ⟶	**Scnr 1**	**Scnr 2**	**Scnr 3**	**Scnr 4**	**Scnr 5**
Indices of the files used for enrolment ⇓	FAR, %	***FRR, %***	FAR, %	FAR, %	FAR, %
Left little finger					
1, 2, and 3	0	***0***	0	0	0
4, 5, and 6	0	***0***	0	0	0
10, 11, and 12	0	***0***	0	0	0
28, 29, and 30	0	***0***	0	0	0
Left middle finger					
1, 2, and 3	0	***0.3***	0	0	0
4, 5, and 6	0	***0.7***	0	0	0
10, 11, and 12	0	***1.0***	0	0	0
28, 29, and 30	0	***1.7***	0	0	0
Left thumb finger					
1, 2, and 3	0	***0.7***	0	0	0
4, 5, and 6	0	***1.3***	0	0	0
10, 11, and 12	0	***0.3***	0	0	0
28, 29, and 30	0	***2.0***	0	0	0
Right index finger					
1, 2, and 3	0	***0.3***	0	0	0
4, 5, and 6	0	***1.0***	0	0	0
10, 11, and 12	0	***1.3***	0	0	0
28, 29, and 30	0	***2.3***	0	0	0
Right ring finger					
1, 2, and 3	0	***1.0***	0	0	0
4, 5, and 6	0	***0.3***	0	0	0
10, 11, and 12	0	***0.3***	0	0	0
28, 29, and 30	0	***0.7***	0	0	0

Table 6.2: Scenario C results: FAR and FRR for Scanner 2 (parameters 55/0.40)

Query scanner ⟶	**Scnr 1**	**Scnr 2**	**Scnr 3**	**Scnr 4**	**Scnr 5**
Indices of the files used for enrolment ⇓	FAR, %	FAR, %	***FRR, %***	FAR, %	FAR, %
Left little finger					
1, 2, and 3	0	0	***1.0***	0	0
4, 5, and 6	0	0	***0.7***	0	0
10, 11, and 12	0.7	0	***0***	0	0
28, 29, and 30	0	0	***3.0***	0.3	0
Left middle finger					
1, 2, and 3	0.3	0	***0***	0	0
4, 5, and 6	0.3	0	***0.3***	0	0
10, 11, and 12	0	0	***2.7***	0	0
28, 29, and 30	0.3	0	***1.0***	0	0
Left thumb finger					
1, 2, and 3	0.7	0	***0.7***	0	0
4, 5, and 6	6.0	1.67	***0***	1.3	1.7
10, 11, and 12	1.7	1.3	***0***	0.3	0.3
28, 29, and 30	1.3	0	***0***	0	0
Right index finger					
1, 2, and 3	0.7	0.7	***0.7***	0.3	0
4, 5, and 6	3.3	1.7	***0***	2.3	0.67
10, 11, and 12	0.3	0	***6.3***	0	0
28, 29, and 30	0	0	***6.3***	0	0
Right ring finger					
1, 2, and 3	2.0	0	***0***	0.3	0
4, 5, and 6	1.0	0	***1.0***	1.3	0
10, 11, and 12	3.3	0	***0.3***	0.3	0
28, 29, and 30	0.3	0.3	***0.3***	0	0

Table 6.3: Scenario C results: FAR and FRR for Scanner 3 (parameters 30/0.40)

Query scanner ⟶	**Scnr 1**	**Scnr 2**	**Scnr 3**	**Scnr 4**	**Scnr 5**
Indices of the files used for enrolment ⇓	FAR, %	FAR, %	FAR, %	***FRR, %***	FAR, %
Left little finger					
1, 2, and 3	0	0.3	0	***2.7***	0
4, 5, and 6	0	0	0	***0***	0
10, 11, and 12	0.3	0.7	0	***2.7***	0
28, 29, and 30	0.3	0	0	***0***	0
Left middle finger					
1, 2, and 3	0	0	0	***2.3***	0
4, 5, and 6	0	0.7	0	***2.0***	1.3
10, 11, and 12	0	0	0	***1.3***	1.3
28, 29, and 30	0	0.7	0	***0.3***	0
Left thumb finger					
1, 2, and 3	0	0.7	0	***2.3***	0
4, 5, and 6	0	0	0	***3.0***	0
10, 11, and 12	0	0	0	***1.7***	1.7
28, 29, and 30	0	0	0	***2.7***	0
Right index finger					
1, 2, and 3	0	0.3	0	***1.3***	0
4, 5, and 6	0	0	0	***3.7***	0
10, 11, and 12	0	0	0	***6.0***	0
28, 29, and 30	0.3	1.0	0	***1.0***	0.7
Right ring finger					
1, 2, and 3	0	0	0	***0.7***	0
4, 5, and 6	0	1.0	0	***1.3***	0
10, 11, and 12	0	0.3	0	***0***	0
28, 29, and 30	0	0.7	0	***0***	0

Table 6.4: Scenario C results: FAR and FRR for Scanner 4 (parameters 30/0.40)

Query scanner ⟶	**Scnr 1**	**Scnr 2**	**Scnr 3**	**Scnr 4**	**Scnr 5**
Indices of the files used for enrolment ⇓	FAR, %	FAR, %	FAR, %	FAR, %	***FRR, %***
Left little finger					
1, 2, and 3	0	0	0	0	***0.3***
4, 5, and 6	0	0	0	0	***1.3***
10, 11, and 12	0	0	0	0	***0***
28, 29, and 30	0	0	0.3	0	***0.3***
Left middle finger					
1, 2, and 3	0.3	0	1.3	0.7	***1.7***
4, 5, and 6	0	0	0.7	0.3	***0***
10, 11, and 12	0.3	0	0	0	***0***
28, 29, and 30	0	0	0	0	***0***
Left thumb finger					
1, 2, and 3	0.3	0	0.7	0.7	***0***
4, 5, and 6	0	0	0.3	0	***0.3***
10, 11, and 12	0.3	0	1.0	1.3	***0***
28, 29, and 30	0.7	0	0.3	0	***1.3***
Right index finger					
1, 2, and 3	0	0	0	0.7	***0***
4, 5, and 6	0.3	0	0	0.3	***1.7***
10, 11, and 12	0	0	0	0	***0***
28, 29, and 30	0	0	0	0	***0***
Right ring finger					
1, 2, and 3	0.3	0	0.3	0	***0***
4, 5, and 6	0	0	0	0	***0***
10, 11, and 12	0	0	0	0.7	***0***
28, 29, and 30	0	0	0	0	***0***

Table 6.5: Scenario C results: FAR and FRR for Scanner 5 (parameters 18/0.25)

6.4 Signal Characteristics

6.4.1 Scanner Noise: Temporal Characteristics

6.4.1.1 Scanner Noise: Temporal Characteristics: Sample Variances

Scnr	**Grp 1**	**Grp 2**	**Grp 3**	**Avg**	**Scnr**	**Grp 1**	**Grp 2**	**Grp 3**	**Avg**
u101	1.759	1.680	1.679	**1.706**	**u114**	2.039	1.977	1.948	**1.988**
u102	1.768	1.674	1.695	**1.712**	**u115**	1.873	1.860	1.850	**1.861**
u103	1.821	1.827	1.848	**1.832**	**u116**	2.075	2.032	1.998	**2.035**
u105	1.907	1.913	1.834	**1.884**	**u117**	1.636	1.679	1.681	**1.665**
u106	1.872	1.958	1.811	**1.880**	**u118**	1.791	1.697	1.679	**1.722**
u107	1.819	1.751	1.749	**1.773**	**u119**	1.432	1.378	1.366	**1.392**
u108	1.749	1.848	1.724	**1.774**	**u120**	1.686	1.651	1.663	**1.667**
u110	1.730	1.664	1.686	**1.693**	**u121**	1.629	1.741	1.739	**1.703**
u111	1.847	1.860	1.792	**1.833**	**u122**	1.790	1.857	1.780	**1.809**
u112	1.966	2.011	1.953	**1.977**	**u151**	1.857	1.840	1.831	**1.843**
u113	1.797	1.714	1.709	**1.740**	**u152**	1.843	1.734	1.768	**1.782**
v1	0.890	0.882	0.883	**0.885**					

Table 6.6: Sample variances of the temporal scanner noise for the three groups of images, each of 100 images, and the average of the three for each scanner: u101 through u152 (UPEK) and v1 (Veridicom)

6.4.1.2 Scanner Noise: Temporal Characteristics: Jarque-Bera Test

Scanner	**Significance 0.01 Groups: 1 2 3, %**			**Signif. 0.01 Maximum, %**	**Significance 0.05 Groups: 1 2 3, %**			**Signif. 0.05 Maximum, %**
u101	6.6	7.8	7.7	**7.8**	16.8	19.0	18.7	**19.0**
u102	8.4	10.5	10.1	**10.5**	18.7	21.9	21.4	**21.9**
u103	11.6	12.9	12.0	**12.9**	24.4	26.8	26.0	**26.8**
u105	11.1	9.8	9.7	**11.1**	23.9	22.4	21.7	**23.9**
u106	9.8	9.5	12.0	**12.0**	22.4	22.4	25.3	**25.3**
u107	9.6	12.1	11.8	**12.1**	23.0	25.9	25.3	**25.9**
u108	9.1	7.4	10.1	**10.1**	21.5	19.2	23.4	**23.4**
u110	7.5	8.8	8.2	**8.8**	19.3	21.3	20.5	**21.3**
u111	8.4	7.4	8.6	**8.6**	19.9	18.3	20.5	**20.5**
u112	9.3	8.7	9.9	**9.9**	21.4	20.5	22.0	**22.0**
u113	9.3	9.8	9.8	**9.8**	20.8	22.1	21.9	**22.1**
u114	7.9	7.8	8.5	**8.5**	18.2	18.6	20.2	**20.2**
u115	9.5	8.2	9.5	**9.5**	21.7	19.3	21.5	**21.7**
u116	8.9	10.0	10.3	**10.3**	21.5	22.9	23.7	**23.7**
u117	11.6	12.7	13.0	**13.0**	24.6	27.2	27.9	**27.9**
u118	7.8	9.3	10.2	**10.2**	19.2	22.1	23.0	**23.0**
u119	6.3	7.4	7.5	**7.5**	15.9	17.8	18.1	**18.1**
u120	9.4	9.9	11.2	**11.2**	22.0	22.7	25.0	**25.0**
u121	9.9	10.4	10.7	**10.7**	21.7	23.4	24.4	**24.4**
u122	8.8	8.1	10.4	**10.4**	20.7	19.7	23.4	**23.4**
u151	10.5	10.9	11.1	**11.1**	23.6	23.9	24.3	**24.3**
u152	9.7	12.3	12.4	**12.4**	22.1	26.5	26.2	**26.5**
v1	21.2	26.2	25.0	**26.2**	43.8	51.3	49.8	**51.3**

Table 6.7: Percentage of pixels (for the 3 groups of images and their maxima) whose Jarque-Bera test p-values for the values of each pixel across 100 images are below 0.01 and 0.05 significance levels. Pixel values deviating over 3σ from the sample mean are excluded ("3σ trimming").

6.4.2 Scanner Noise: Spatial Characteristics

6.4.2.1 Scanner Noise: Spatial Characteristics: Sample Variances

Scnr	**Grp 1**	**Grp 2**	**Grp 3**	**Avg**	**Scnr**	**Grp 1**	**Grp 2**	**Grp 3**	**Avg**
u101	1.670	1.649	1.649	**1.656**	**u114**	1.967	1.926	1.902	**1.932**
u102	1.662	1.652	1.648	**1.654**	**u115**	1.834	1.805	1.822	**1.820**
u103	1.792	1.803	1.794	**1.796**	**u116**	2.015	1.984	1.975	**1.991**
u105	1.871	1.842	1.809	**1.841**	**u117**	1.599	1.652	1.659	**1.637**
u106	1.818	1.785	1.788	**1.797**	**u118**	1.701	1.660	1.656	**1.672**
u107	1.749	1.731	1.710	**1.730**	**u119**	1.336	1.346	1.348	**1.343**
u108	1.696	1.693	1.700	**1.696**	**u120**	1.619	1.598	1.637	**1.618**
u110	1.665	1.634	1.641	**1.647**	**u121**	1.608	1.680	1.716	**1.668**
u111	1.801	1.772	1.769	**1.781**	**u122**	1.722	1.731	1.756	**1.737**
u112	1.935	1.911	1.896	**1.914**	**u151**	1.816	1.814	1.810	**1.813**
u113	1.765	1.668	1.658	**1.697**	**u152**	1.734	1.709	1.723	**1.722**
v1	0.868	0.870	0.868	**0.869**					

Table 6.8: Sample variances for the three groups, each of 100 images, and the average of the three for each scanner: u101 through u152 (UPEK) and v1 (Veridicom)

6.4.2.2 Scanner Noise: Spatial Characteristics: χ^2 Test

Scanner	**Significance 0.01 Groups: 1 2 3, %**			**Signif. 0.01 Maximum, %**	**Significance 0.05 Groups: 1 2 3, %**			**Signif. 0.05 Maximum, %**
u101	6.9	7.5	7.4	**7.5**	17.8	18.7	18.6	**18.7**
u102	7.8	8.4	7.5	**8.4**	19.2	20.8	19.3	**20.8**
u103	8.9	10.8	11.4	**11.4**	21.7	24.2	25.5	**25.5**
u105	9.4	9.4	8.1	**9.4**	22.0	22.3	19.9	**22.3**
u106	9.8	9.8	9.7	**9.8**	22.8	23.0	23.0	**23.0**
u107	11.2	10.7	11.4	**11.4**	25.1	24.9	25.5	**25.5**
u108	8.6	8.7	8.9	**8.9**	20.7	21.1	21.2	**21.2**
u110	8.3	8.6	8.6	**8.6**	20.6	21.2	21.0	**21.2**
u111	9.4	8.8	9.3	**9.4**	21.4	21.4	22.0	**22.0**
u112	9.7	10.0	8.9	**10.0**	22.9	23.6	21.9	**23.6**
u113	8.7	10.7	10.1	**10.7**	20.4	24.7	23.1	**24.7**
u114	8.2	8.3	9.4	**9.4**	19.6	20.7	22.6	**22.6**
u115	9.2	8.1	8.7	**9.2**	21.8	19.6	21.2	**21.8**
u116	10.9	11.6	11.5	**11.6**	24.7	25.9	26.1	**26.1**
u117	11.1	13.8	14.4	**14.4**	24.2	29.2	29.6	**29.6**
u118	9.7	10.6	10.7	**10.7**	22.3	24.2	24.5	**24.5**
u119	6.4	6.7	7.0	**7.0**	16.9	17.5	17.5	**17.5**
u120	10.5	10.7	11.6	**11.6**	23.8	24.1	25.9	**25.9**
u121	9.2	11.4	11.5	**11.5**	21.3	25.5	25.6	**25.6**
u122	8.2	9.1	9.2	**9.2**	20.6	21.6	21.9	**21.9**
u151	11.5	11.7	11.5	**11.7**	25.6	25.8	25.8	**25.8**
u152	11.1	12.4	13.1	**13.1**	25.1	27.4	28.3	**28.3**
v1	14.8	16.2	15.9	**16.2**	27.1	28.9	28.8	**28.9**

Table 6.9: Average percentage of columns (for the 3 groups of images and their maxima) whose χ^2 test p-values for the scanner noise estimate are below 0.01 and 0.05 significance levels. Pixel values deviating over 3σ from the sample mean are excluded ("3σ trimming").

6.4.2.3 Scanner Noise: Spatial Characteristics: Jarque-Bera Test

	Significance 0.01			**Signif. 0.01**	**Significance 0.05**			**Signif. 0.05**
Scanner	**Groups: 1 2 3, %**			**Maximum, %**	**Groups: 1 2 3, %**			**Maximum, %**
u101	15.4	16.9	16.5	**16.9**	32.8	34.6	33.8	**34.6**
u102	21.5	23.6	23.5	**23.6**	40.0	42.3	42.0	**42.3**
u103	27.4	31.9	33.0	**33.0**	46.5	51.2	52.8	**52.8**
u105	26.6	25.5	21.2	**26.6**	45.3	45.1	39.0	**45.3**
u106	24.9	27.9	28.2	**28.2**	44.8	47.8	47.7	**47.8**
u107	26.7	31.2	32.0	**32.0**	47.2	51.1	51.5	**51.5**
u108	20.3	21.5	21.6	**21.6**	39.4	40.9	41.2	**41.2**
u110	18.0	20.3	19.6	**20.3**	36.5	39.5	38.8	**39.5**
u111	21.3	19.3	19.8	**21.3**	39.6	37.8	39.2	**39.6**
u112	23.1	24.6	25.4	**25.4**	42.4	44.8	45.0	**45.0**
u113	22.0	26.6	26.8	**26.8**	40.1	46.1	46.5	**46.5**
u114	18.5	18.4	20.8	**20.8**	35.4	36.2	39.5	**39.5**
u115	23.4	19.6	22.2	**23.4**	43.0	37.5	41.7	**41.7**
u116	23.5	26.2	27.3	**27.3**	43.1	46.8	47.7	**47.7**
u117	30.7	34.5	35.9	**35.9**	50.3	54.9	56.3	**56.3**
u118	21.8	25.1	25.7	**25.7**	41.2	45.5	45.9	**45.9**
u119	14.5	15.4	15.2	**15.4**	32.0	32.8	32.5	**32.8**
u120	25.5	26.5	28.8	**28.8**	45.7	46.6	49.5	**49.5**
u121	23.9	28.9	28.6	**28.9**	43.2	49.7	49.4	**49.7**
u122	20.5	22.5	23.6	**23.6**	39.5	41.9	43.6	**43.6**
u151	29.1	28.9	29.0	**29.1**	49.2	48.8	49.4	**49.4**
u152	31.0	35.8	36.2	**36.2**	51.0	56.0	56.4	**56.4**
v1	50.5	51.4	51.5	**51.5**	63.6	64.8	64.8	**64.8**

Table 6.10: Average percentage of columns (for the 3 groups and their maxima) whose Jarque-Bera test p-values for the scanner noise estimate for the values of the each pixel across 100 images are below 0.01 and 0.05 significance levels. Pixel values deviating over 3σ from the sample mean are excluded ("3σ trimming").

6.4.3 Scanner Pattern Characteristics

6.4.3.1 Scanner Pattern Characteristics: Sample Variances

Scnr	Grp 1	Grp 2	Grp 3	Avg	Scnr	Grp 1	Grp 2	Grp 3	Avg
u101	16.360	16.381	16.411	**16.384**	**u114**	15.628	15.692	15.652	**15.657**
u102	15.315	15.365	15.357	**15.346**	**u115**	17.601	17.558	17.634	**17.598**
u103	19.269	19.352	19.331	**19.317**	**u116**	13.157	13.215	13.177	**13.183**
u105	18.472	18.530	18.425	**18.475**	**u117**	12.726	12.696	12.729	**12.717**
u106	14.150	14.120	14.150	**14.140**	**u118**	14.107	14.116	14.111	**14.111**
u107	15.431	15.414	15.403	**15.416**	**u119**	12.420	12.482	12.475	**12.459**
u108	16.815	16.796	16.796	**16.802**	**u120**	15.066	15.096	15.071	**15.078**
u110	14.651	14.771	14.797	**14.740**	**u121**	15.059	15.134	15.121	**15.105**
u111	17.194	17.264	17.266	**17.241**	**u122**	16.844	16.978	16.932	**16.918**
u112	14.394	14.423	14.390	**14.403**	**u151**	15.837	15.902	15.883	**15.874**
u113	14.165	14.161	14.130	**14.152**	**u152**	14.227	14.207	14.156	**14.197**
v1	8.501	8.672	8.763	**8.645**					

Table 6.11: Sample variances $\hat{\sigma}_s^2$ of the variable part, computed with an 11-tap moving-average filter, of the scanner pattern, estimated from each group of 100 images, and the average of the three for each scanner: u101 through u152 (UPEK) and v1 (Veridicom)

6.4.3.2 Scanner Pattern Characteristics: Jarque-Bera Test

Scanner	**Significance 0.01 Groups: 1 2 3, %**	**Signif. 0.01 Maximum, %**	**Significance 0.05 Groups: 1 2 3, %**	**Signif. 0.05 Maximum, %**
u101	0.0 0.0 0.0	**0.0**	0.8 0.8 1.6	**1.6**
u102	0.0 0.0 0.0	**0.0**	0.4 0.4 0.4	**0.4**
u103	0.0 0.0 0.0	**0.0**	1.6 1.2 1.2	**1.6**
u105	0.0 0.0 0.0	**0.0**	2.3 3.5 2.3	**3.5**
u106	0.0 0.0 0.0	**0.0**	1.6 1.2 1.2	**1.6**
u107	0.0 0.0 0.0	**0.0**	1.2 1.6 1.2	**1.6**
u108	0.0 0.0 0.0	**0.0**	1.6 1.2 1.2	**1.6**
u110	0.0 0.0 0.0	**0.0**	0.4 0.4 0.0	**0.4**
u111	0.0 0.0 0.0	**0.0**	0.8 0.8 0.8	**0.8**
u112	0.0 0.0 0.0	**0.0**	0.0 0.0 0.0	**0.0**
u113	0.0 0.0 0.0	**0.0**	1.6 2.0 2.3	**2.3**
u114	0.0 0.0 0.0	**0.0**	0.8 1.2 0.8	**1.2**
u115	0.0 0.0 0.0	**0.0**	1.6 2.0 1.6	**2.0**
u116	0.4 0.4 0.4	**0.4**	1.2 0.8 1.2	**1.2**
u117	0.0 0.0 0.0	**0.0**	1.2 1.2 1.2	**1.2**
u118	0.0 0.0 0.0	**0.0**	1.2 1.6 0.8	**1.6**
u119	0.0 0.0 0.0	**0.0**	0.8 0.8 1.6	**1.6**
u120	0.0 0.0 0.0	**0.0**	2.7 2.3 3.5	**3.5**
u121	0.0 0.0 0.0	**0.0**	0.4 0.4 0.4	**0.4**
u122	0.0 0.0 0.0	**0.0**	0.8 0.4 1.2	**1.2**
u151	0.0 0.0 0.0	**0.0**	0.4 0.4 0.4	**0.4**
u152	0.0 0.0 0.0	**0.0**	0.4 0.8 0.4	**0.8**
v1	0.4 0.4 0.4	**0.4**	1.6 1.6 1.6	**1.6**

Table 6.12: Percentage of columns (for the 3 groups and their maxima) whose Jarque-Bera test p-values for the variable part, computed with an 11-tap moving-average filter, of the scanner pattern, estimated from each group of 100 images, are below 0.01 and 0.05 significance levels. Values deviating over 3σ from the sample mean are excluded ("3σ trimming").

6.4.3.3 Scanner Pattern Characteristics: Anderson-Darling Test

Scanner	**Significance 0.01 Groups: 1 2 3, %**			**Signif. 0.01 Maximum, %**	**Significance 0.05 Groups: 1 2 3, %**			**Signif. 0.05 Maximum, %**
u101	0.8	0.4	0.4	**0.8**	4.3	3.5	3.1	**4.3**
u102	1.2	1.6	0.8	**1.6**	3.9	3.9	3.9	**3.9**
u103	1.2	1.2	1.2	**1.2**	6.6	5.5	5.1	**6.6**
u105	1.6	1.6	1.2	**1.6**	5.1	4.7	4.3	**5.1**
u106	0.8	0.4	0.0	**0.8**	4.3	5.1	5.5	**5.5**
u107	1.6	2.3	2.0	**2.3**	4.3	5.1	5.9	**5.9**
u108	0.4	0.8	1.6	**1.6**	5.9	4.3	4.7	**5.9**
u110	1.6	0.8	0.8	**1.6**	6.6	6.2	5.1	**6.6**
u111	0.4	0.4	0.4	**0.4**	2.7	3.5	3.9	**3.9**
u112	1.6	1.2	0.8	**1.6**	5.9	6.2	5.9	**6.2**
u113	1.6	2.3	2.3	**2.3**	7.4	7.4	7.0	**7.4**
u114	1.6	1.6	2.0	**2.0**	5.9	5.1	5.9	**5.9**
u115	0.8	0.4	0.4	**0.8**	6.2	5.5	5.9	**6.2**
u116	0.8	0.8	0.8	**0.8**	3.9	2.7	3.1	**3.9**
u117	1.6	2.0	1.6	**2.0**	5.9	5.9	6.2	**6.2**
u118	0.4	0.4	0.8	**0.8**	4.7	4.3	5.1	**5.1**
u119	0.4	0.8	0.8	**0.8**	4.7	4.3	5.1	**5.1**
u120	2.0	1.6	1.6	**2.0**	8.2	7.0	6.6	**8.2**
u121	0.4	0.8	0.8	**0.8**	5.5	5.5	3.9	**5.5**
u122	1.2	0.4	0.4	**1.2**	4.3	3.9	4.7	**4.7**
u151	0.4	0.4	0.4	**0.4**	6.6	5.9	6.2	**6.6**
u152	1.2	1.2	1.2	**1.2**	6.2	5.9	5.5	**6.2**
v1	1.6	2.3	2.3	**2.3**	5.1	5.1	5.5	**5.5**

Table 6.13: Percentage of columns (for the 3 groups and their maxima) whose Anderson-Darling test p-values for the variable part, computed with an 11-tap moving-average filter, of the scanner pattern, estimated from each group of 100 images, are below 0.01 and 0.05 significance levels. Values deviating over 3σ from the sample mean are excluded ("3σ trimming").

6.4.4 Scanner Pattern and Noise Characteristics

6.4.4.1 Scanner Pattern and Noise Characteristics: Sample Variances

Scnr	**Grp 1**	**Grp 2**	**Grp 3**	**Avg**	**Scnr**	**Grp 1**	**Grp 2**	**Grp 3**	**Avg**
u101	17.878	17.886	17.914	**17.893**	**u114**	17.415	17.443	17.383	**17.413**
u102	16.825	16.868	16.855	**16.850**	**u115**	19.265	19.199	19.294	**19.253**
u103	20.899	20.996	20.962	**20.952**	**u116**	14.987	15.023	14.972	**14.994**
u105	20.171	20.207	20.069	**20.149**	**u117**	14.176	14.196	14.237	**14.203**
u106	15.798	15.738	15.774	**15.770**	**u118**	15.652	15.628	15.620	**15.633**
u107	17.024	16.989	16.956	**16.990**	**u119**	13.632	13.704	13.697	**13.678**
u108	18.354	18.335	18.338	**18.342**	**u120**	16.534	16.553	16.556	**16.548**
u110	16.166	16.258	16.290	**16.238**	**u121**	16.520	16.666	16.657	**16.615**
u111	18.834	18.881	18.874	**18.863**	**u122**	18.406	18.550	18.526	**18.494**
u112	16.153	16.164	16.114	**16.143**	**u151**	17.486	17.554	17.533	**17.525**
u113	15.765	15.678	15.636	**15.693**	**u152**	15.806	15.765	15.722	**15.764**
v1	9.289	9.464	9.553	**9.436**					

Table 6.14: Sample variances of the variable part, computed with an 11-tap moving-average filter, of the pixel values $g^{(po)}$ in each image in the three groups of 100 images, and the average of the three for each scanner: u101 through u152 (UPEK) and v1 (Veridicom)

6.4.4.2 Scanner Pattern and Noise Characteristics: Jarque-Bera Test

Scanner	**Significance 0.01 Groups: 1 2 3, %**	**Signif. 0.01 Maximum, %**	**Significance 0.05 Groups: 1 2 3, %**	**Signif. 0.05 Maximum, %**
u101	0.0 0.0 0.0	**0.0**	1.1 1.1 1.2	**1.2**
u102	0.0 0.0 0.0	**0.0**	1.0 1.0 1.1	**1.1**
u103	0.0 0.0 0.0	**0.0**	1.6 1.4 1.4	**1.6**
u105	0.0 0.1 0.1	**0.1**	1.7 1.9 2.0	**2.0**
u106	0.0 0.0 0.0	**0.0**	1.5 1.6 1.6	**1.6**
u107	0.0 0.0 0.0	**0.0**	1.5 1.5 1.4	**1.5**
u108	0.0 0.0 0.0	**0.0**	1.2 1.2 1.2	**1.2**
u110	0.0 0.0 0.0	**0.0**	1.0 0.8 0.9	**1.0**
u111	0.0 0.0 0.0	**0.0**	0.7 0.6 0.7	**0.7**
u112	0.0 0.0 0.0	**0.0**	1.0 1.0 1.0	**1.0**
u113	0.0 0.0 0.0	**0.0**	1.6 1.5 1.6	**1.6**
u114	0.0 0.0 0.0	**0.0**	0.9 1.1 0.9	**1.1**
u115	0.0 0.0 0.0	**0.0**	1.3 1.3 1.3	**1.3**
u116	0.4 0.4 0.4	**0.4**	1.3 1.3 1.3	**1.3**
u117	0.0 0.0 0.0	**0.0**	1.1 1.1 1.1	**1.1**
u118	0.0 0.0 0.0	**0.0**	0.9 0.8 0.9	**0.9**
u119	0.0 0.0 0.0	**0.0**	1.1 1.2 1.1	**1.2**
u120	0.0 0.1 0.1	**0.1**	1.6 1.6 1.5	**1.6**
u121	0.0 0.0 0.1	**0.1**	0.8 0.9 0.9	**0.9**
u122	0.0 0.0 0.0	**0.0**	0.9 0.9 0.9	**0.9**
u151	0.1 0.0 0.0	**0.1**	1.2 1.3 1.2	**1.3**
u152	0.0 0.0 0.0	**0.0**	0.7 0.8 0.8	**0.8**
v1	0.4 0.5 0.4	**0.5**	2.5 2.4 2.4	**2.5**

Table 6.15: Average percentage of columns (for the 3 groups and their maxima) whose Jarque-Bera test p-values for the variable part, computed with an 11-tap moving-average filter, of the pixel values $g^{(po)}$ are below 0.01 and 0.05 significance levels. Deviations beyond 3σ from the sample mean are excluded ("3σ trimming").

6.4.4.3 Scanner Pattern and Noise Charac's: Anderson-Darling Test

Scanner	**Significance 0.01 Groups: 1 2 3, %**	**Signif. 0.01 Maximum, %**	**Significance 0.05 Groups: 1 2 3, %**	**Signif. 0.05 Maximum, %**
u101	1.0 0.8 0.9	**1.0**	4.9 4.7 5.0	**5.0**
u102	0.8 0.7 0.8	**0.8**	4.4 4.6 4.5	**4.6**
u103	1.1 1.2 1.3	**1.3**	6.0 5.7 5.8	**6.0**
u105	1.1 1.2 1.2	**1.2**	4.8 4.9 5.0	**5.0**
u106	1.2 1.2 1.2	**1.2**	5.7 5.9 5.8	**5.9**
u107	1.1 1.4 1.3	**1.4**	5.7 6.2 6.2	**6.2**
u108	1.2 1.2 1.4	**1.4**	5.5 5.7 6.0	**6.0**
u110	1.1 0.9 1.1	**1.1**	5.9 5.3 5.4	**5.9**
u111	0.6 0.6 0.6	**0.6**	3.7 3.7 4.0	**4.0**
u112	1.0 1.1 1.3	**1.3**	5.4 5.5 5.5	**5.5**
u113	1.5 1.3 1.4	**1.5**	6.5 6.4 6.6	**6.6**
u114	1.4 1.3 1.3	**1.4**	5.5 5.7 5.7	**5.7**
u115	1.3 1.2 1.2	**1.3**	5.9 5.7 5.7	**5.9**
u116	1.2 1.3 1.3	**1.3**	5.0 5.1 5.1	**5.1**
u117	1.4 1.4 1.4	**1.4**	5.9 5.8 5.7	**5.9**
u118	0.7 0.7 0.7	**0.7**	4.5 4.5 4.6	**4.6**
u119	1.0 1.0 1.0	**1.0**	5.0 5.0 4.9	**5.0**
u120	1.7 1.7 1.7	**1.7**	6.9 6.8 6.8	**6.9**
u121	0.9 0.8 0.8	**0.9**	5.0 4.9 4.8	**5.0**
u122	0.9 0.9 0.9	**0.9**	5.2 4.9 5.1	**5.2**
u151	1.0 1.0 1.0	**1.0**	5.1 5.0 5.0	**5.1**
u152	1.0 1.1 1.0	**1.1**	5.1 4.8 4.9	**5.1**
v1	1.8 1.9 1.8	**1.9**	6.1 5.9 6.0	**6.1**

Table 6.16: Average percentage of columns (for the 3 groups and their maxima) whose Anderson-Darling p-values for the variable part, computed with an 11-tap moving-average filter, of the pixel values $g^{(po)}$ are below 0.01 and 0.05 significance levels. Deviations beyond 3σ from the sample mean are excluded ("3σ trimming").

6.5 Gaussian Approximation of the Inverse of a Gaussian Random Variable

6.5.1 Derivation of the Approximation

Let X be a Gaussian random variable with PDF $f_X(x) \sim N\left(\mu, \sigma^2\right)$ and $Y = \frac{1}{X}$. In our case, $\mu \gg \sigma$, e.g., μ of about 100 and σ about 5, which is the worst case as the mean is the smallest and the standard deviation of the scanner pattern, possibly together with the spatial scanner noise, is the largest. Then:

$$P(X > x) = Q\left(\frac{x-\mu}{\sigma}\right) \qquad \text{since Gaussian}$$
$$P(X > 0) = Q\left(\frac{0-100}{5}\right) = Q(-20) = 1 - Q(20) \approx 1 - 3 * 10^{-89} \approx 1.$$

Therefore, we can very safely assume that $P(X > 0) = 1$ for all practical purposes (this is in addition to the fact that the pixel values in the images the scanners produce are positive or at least non-negative).

$$P(Y \le y) = P\left(\frac{1}{X} \le y\right) = P\left(X \ge \frac{1}{y}\right) \text{ since } X \text{ and } y \text{ can only be positive} \Rightarrow$$
$$P(Y \le y) = Q\left(\frac{\frac{1}{y}-\mu}{\sigma}\right) \text{ since } X \text{ is } N(\mu, \sigma^2), \text{and}$$
$$f_Y(y) = \left[P(Y \le y)\right]_y^{'} = Q\left(\frac{\frac{1}{y}-\mu}{\sigma}\right)_y^{'}$$

Let U be $N(0,1)$ and $f_U(u)$ is its PDF. Then :

$$\left[Q(u)\right]_u^{'} = \left[1 - P(U \le u)\right]_u^{'} = -\left[P(U \le u)\right]_u^{'} = -f_U(u) = -\frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}} \Rightarrow$$

$$f_Y(y) = Q\left(\frac{\frac{1}{y}-\mu}{\sigma}\right)'_y = -\frac{1}{\sqrt{2\pi}}exp\left(-\frac{1}{2}\left(\frac{\frac{1}{y}-\mu}{\sigma}\right)^2\right)\left(\frac{\frac{1}{y}-\mu}{\sigma}\right)'_y$$
$$= -\frac{1}{\sqrt{2\pi}}exp\left(-\frac{\left(\frac{1}{y}-\mu\right)^2}{2\sigma^2}\right)\left(-\frac{1}{y^2\sigma}\right) \Rightarrow$$

$$f_Y(y) = -\frac{1}{\sqrt{2\pi}\sigma y^2}exp\left(-\frac{\left(\frac{1}{y}-\mu\right)^2}{2\sigma^2}\right) \tag{6.1}$$

The PDF of Y in Expression 6.1 is not simple and easy to work with, but at least it is in closed form. Since $\mu \gg \sigma$ and $\mu \sim 200$, we now introduce two approximations:

- **Approximation #1:** $\frac{1}{y^2} \approx \frac{1}{\mu^2}$ because variations around μ will be negligibly small. Moreover, in the regions far from μ, $f_Y(y)$ will be dominated by the exponent, which exponent will be close to 0 anyway;
- **Approximation #2:** $exp\left(-\frac{\left(\frac{1}{y}-\mu\right)^2}{2\sigma^2}\right) \approx exp\left(-\frac{\left(y-\frac{1}{\mu}\right)^2}{2\left(\frac{\sigma}{\mu^2}\right)^2}\right)$, again around μ.

These two approximations seem to be very imprecise; when applied, however, the overall error is acceptably small for our purposes. Next we detail the steps in deriving Approximation #2. Using the Taylor series of the function $\left(\frac{1}{y}-\mu\right)$ around $\frac{1}{\mu}$, we receive:

$$\left(\frac{1}{y}-\mu\right) \approx \left(\frac{1}{y}-\mu\right)\Big|_{y=\frac{1}{\mu}} + \frac{\left(y-\frac{1}{\mu}\right)}{1!}\left(-\frac{1}{y^2}\right)\Big|_{y=\frac{1}{\mu}} + \frac{\left(y-\frac{1}{\mu}\right)^2}{2!}\left(\frac{2}{y^3}\right)\Big|_{y=\frac{1}{\mu}} +$$
$$+ \frac{\left(y-\frac{1}{\mu}\right)^n}{n!}\left(-\frac{(-1)^n.n}{y^{n+1}}\right)\Big|_{y=\frac{1}{\mu}} + \cdots$$

We observed that by using only the first several (2 or 3) terms, the Taylor series changes in magnitude considerably, both in positive and in negative directions. Since

$\left(\frac{1}{y}-\mu\right)$ participates squared in $f_Y(y)$ (see Expression 6.1), using the Taylor series with only these first several terms will result in large positive values for $\left(\frac{1}{y}-\mu\right)^2$. However, since $\left(\frac{1}{y}-\mu\right)^2$ is in the exponent of $f_Y(y)$ with a negative sign, the large positive values effectively make the whole exponent close to 0. Now, by taking only the first term, we have:

$$\left(\frac{1}{y}-\mu\right) \approx \frac{\left(y-\frac{1}{\mu}\right)}{1!}\left(-\frac{1}{y^2}\right)\Bigg|_{y=\frac{1}{\mu}} = -\left(y-\frac{1}{\mu}\right)\mu^2 \Rightarrow$$

$$exp\left(-\frac{\left(\frac{1}{y}-\mu\right)^2}{2\sigma^2}\right) \approx exp\left(-\frac{\left(\left(y-\frac{1}{\mu}\right)\mu^2\right)^2}{2\sigma^2}\right) = exp\left(-\frac{\left(y-\frac{1}{\mu}\right)^2}{2\left(\frac{\sigma}{\mu^2}\right)^2}\right)$$

By combining Approximations #1 and #2, we receive:

$$f_Y(y) \approx -\frac{1}{\sqrt{2\pi}\left(\frac{\sigma}{\mu^2}\right)} exp\left(-\frac{\left(y-\frac{1}{\mu}\right)^2}{2\left(\frac{\sigma}{\mu^2}\right)^2}\right) \sim N\left(\frac{1}{\mu},\left(\frac{\sigma}{\mu^2}\right)^2\right) \qquad (6.2)$$

An alternative view on this is to look at the function $y = 1/x$ at given x_0 (which represents the mean μ_X). The larger x_0 is, the more linear the function $y = 1/x$ becomes, and small perturbations Δx around x_0 will give almost linearly dependent on them deviations Δy, i.e., $\Delta y \approx k\Delta x$. Therefore, a Gaussian Δx will be almost linearly transformed into Δy, which is again Gaussian by theorem, thus corroborating the approximation just derived.

In sum, when $X \sim N(\mu, \sigma^2)$ and $\mu \gg \sigma$ (also $\mu > 0$), then $1/X$ is approximately Gaussian $N\left(\frac{1}{\mu},\left(\frac{\sigma}{\mu^2}\right)^2\right)$.

6.5.2 Numerical Evaluation of the Approximation

The comparison between the exact PDF of the inverse Y of a Gaussian random variable X with $N(200, 5^2)$ and the approximate PDF is shown in Figure 6.6. The mean of the inverse Y is $\mu_Y = 1/\mu_X = 5 * 10^{-3}$, and the standard deviation of the approximating Gaussian is $\sigma_Y = \sigma_X/\mu_X^2 = 1.25 * 10^{-4}$. Note that the PDF becomes very large (over 3000) but diminishes very quickly just $\pm 4 * 10^{-4}$ (which is $\pm 3\sigma_Y$) away from the mean μ_Y.

The second plot shows the relative error, which is within ±10% in the range $\pm 2\sigma_Y$ of Y around its mean. The tails beyond $\pm 2\sigma_Y$ have probability $2 * Q(2) \approx 4.55 * 10^{-2}$. This means that roughly 5% of the pixels (on average) lie outside the 10% relative error window, which we believe is tolerable.

The third plot on the same figure shows the reduced to $|y - \mu|$ curves for both the exact PDF and the Gaussian approximation. The reduction is intended to present the PDFs on a linear scale (by taking logarithm) in order to allow easier comparison. Thus, when $g(y)$ is a Gaussian PDF with $N(\mu, \sigma^2)$, then $|y - \mu| = \sqrt{-2\sigma^2.log\left(\sqrt{2\pi}.\sigma.g(y)\right)}$. The plot shows that the two PDFs are indeed very close to each other.

Tables 6.17 through 6.20 show the p-values of 4 hypothesis tests (χ^2, Jarque-Bera, Lilliefors, and Anderson-Darling) for the inverses of 4 types of toy processes, with 20 signals (realizations) for each type. Each signal (realization) is 360 pixels (samples) long in order to match the length of one column of pixels of the UPEK scanners. All 4 types of processes are Gaussians with parameters being all pairs of

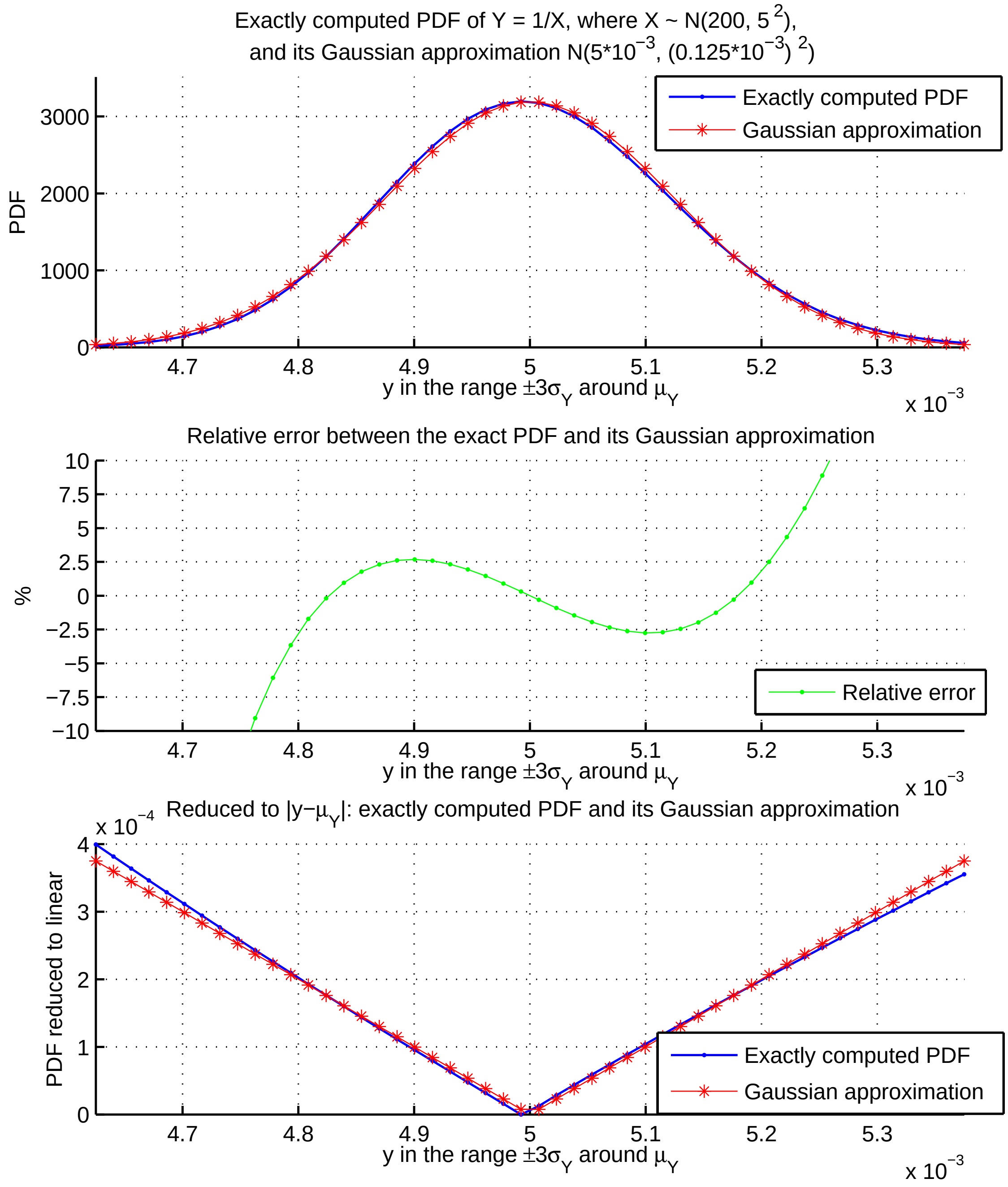


Figure 6.6: Inverse of a Gaussian RV: the exact PDF and the approximate PDF

two extreme means (200 and 100) and two extreme standard deviations (5 and 3, although the smaller the standard deviation is, the more accurate the approximation is): $N\left(200, 5^2\right)$, $N\left(100, 5^2\right)$, $N\left(200, 3^2\right)$, and $N\left(100, 3^2\right)$, all generated with the

MATLAB function `randn.m`. The original (i.e., before the inversion) of each signal shown in the tables below has p-values above 0.05 for all hypothesis tests, i.e., all original signals can be assumed Gaussian. (Note: the largest p-value returned by the Jarque-Bera and Lilliefors tests is 0.5).

For the original process $N\left(200, 5^2\right)$ (see Table 6.17), only three of the inverse signals have p-values of some, but not all, of the hypothesis tests below 0.05 significance level: signals #02, #11 and #18. The only problematic of the three is probably signal #02, which has two p-values below 0.01. Signals #12 and #15 have p-values of a single test slightly below 0.05.

When the mean of the original process is much smaller, i.e., for $N\left(100, 5^2\right)$ (see Table 6.18), the signals whose p-values are smaller than the 0.05 significance level threshold are many more: signals #06, #13, and #16 have one test below the threshold, signals #03, #07, and #15 have two tests below the threshold, and signals #11 and #17 have the Gaussian hypothesis rejected for all four tests. Still, the overwhelming majority of tests have very large p-values, well above 0.05.

As for the original process $N\left(200, 3^2\right)$ (see Table 6.19), only one (signal #03) of the inverse signals has p-values of three of the hypothesis tests below 0.05 significance level, although these p-values are quite large (close to 0.04). From the 4 types of toy processes, the inverse signals of this type appear to best conform to the Gaussian distribution. This is to be expected as its mean is the largest (200) and the standard deviation the smallest (3), which makes the approximation more accurate than for the other 3 types of processes. This is very intuitive when looking from the alternative point of view on the approximation in the preceding section.

Signal #	χ^2	**Jarque-Bera**	**Lilliefors**	**Anderson-Darling**
01	0.408	0.500	0.500	0.944
02	0.066	0.105	0.008	0.007
03	0.368	0.500	0.470	0.525
04	0.631	0.086	0.172	0.089
05	0.730	0.500	0.500	0.896
06	0.230	0.500	0.500	0.524
07	0.262	0.500	0.298	0.392
08	0.107	0.061	0.173	0.035
09	0.725	0.131	0.400	0.519
10	0.775	0.500	0.312	0.749
11	0.049	0.038	0.046	0.094
12	0.515	0.052	0.047	0.142
13	0.775	0.500	0.500	0.924
14	0.151	0.064	0.228	0.195
15	0.573	0.043	0.139	0.311
16	0.614	0.172	0.063	0.297
17	0.612	0.500	0.500	0.784
18	0.057	0.046	0.099	0.045
19	0.256	0.374	0.418	0.504
20	0.571	0.405	0.500	0.726

Table 6.17: P-values of the hypothesis tests for the inverse of a toy process $N\left(200, 5^2\right)$

Finally, the results for the original process $N\left(100, 3^2\right)$ (shown in Table 6.20) is similar to those for $N\left(200, 5^2\right)$: only three of the inverse signals have p-values of some, but not all, of the hypothesis tests below 0.05 significance level: signals #07 (for one test), #12 and #19 (for two tests). This is also expected because, although the mean here is half the mean of $N\left(200, 5^2\right)$, the standard deviation is also nearly half of the standard deviation of $N\left(200, 5^2\right)$, and therefore, the approximation is

Signal #	χ^2	**Jarque-Bera**	**Lilliefors**	**Anderson-Darling**
01	0.325	0.146	0.500	0.328
02	0.087	0.094	0.244	0.268
03	0.137	0.001	0.125	0.028
04	0.684	0.067	0.500	0.385
05	0.108	0.071	0.052	0.060
06	0.122	0.006	0.196	0.128
07	0.268	0.033	0.088	0.027
08	0.102	0.083	0.149	0.165
09	0.529	0.500	0.500	0.903
10	0.497	0.036	0.469	0.137
11	0.008	0.007	0.010	0.001
12	0.063	0.068	0.500	0.142
13	0.648	0.043	0.337	0.403
14	0.308	0.500	0.278	0.440
15	0.055	0.021	0.064	0.009
16	0.588	0.041	0.453	0.362
17	0.011	0.022	0.047	0.008
18	0.580	0.098	0.500	0.381
19	0.631	0.439	0.437	0.529
20	0.142	0.366	0.281	0.269

Table 6.18: P-values of the hypothesis tests for the inverse of a toy process $N\left(100, 5^2\right)$

comparably accurate.

Signal #	χ^2	**Jarque-Bera**	**Lilliefors**	**Anderson-Darling**
01	0.799	0.500	0.500	0.565
02	0.931	0.489	0.420	0.748
03	0.048	0.500	0.039	0.035
04	0.184	0.360	0.448	0.432
05	0.765	0.500	0.500	0.887
06	0.125	0.405	0.189	0.274
07	0.817	0.217	0.500	0.837
08	0.250	0.500	0.254	0.205
09	0.894	0.500	0.500	0.916
10	0.720	0.500	0.500	0.534
11	0.100	0.090	0.299	0.116
12	0.073	0.500	0.346	0.523
13	0.475	0.388	0.500	0.602
14	0.209	0.500	0.167	0.201
15	0.528	0.500	0.500	0.533
16	0.499	0.500	0.500	0.804
17	0.691	0.500	0.436	0.345
18	0.544	0.210	0.478	0.410
19	0.846	0.500	0.500	0.867
20	0.116	0.500	0.125	0.143

Table 6.19: P-values of the hypothesis tests for the inverse of a toy process $N\left(200, 3^2\right)$

Signal #	χ^2	**Jarque-Bera**	**Lilliefors**	**Anderson-Darling**
01	0.990	0.500	0.500	0.883
02	0.050	0.447	0.394	0.412
03	0.510	0.303	0.079	0.247
04	0.839	0.500	0.500	0.968
05	0.711	0.255	0.307	0.515
06	0.758	0.500	0.500	0.658
07	0.201	0.069	0.076	0.022
08	0.888	0.341	0.500	0.594
09	0.241	0.500	0.500	0.214
10	0.058	0.331	0.500	0.219
11	0.603	0.358	0.500	0.348
12	0.061	0.006	0.180	0.015
13	0.871	0.420	0.500	0.902
14	0.052	0.500	0.500	0.502
15	0.225	0.449	0.076	0.147
16	0.714	0.500	0.500	0.969
17	0.583	0.500	0.321	0.200
18	0.080	0.210	0.096	0.066
19	0.177	0.038	0.030	0.088
20	0.896	0.494	0.500	0.761

Table 6.20: P-values of the hypothesis tests for the inverse of a toy process $N\left(100, 3^2\right)$

6.6 Linear Approximations for Signal Model A

6.6.1 Linear Approximation for $g(i,j)$

In this section, we derive a linear approximation for $g(i,j)$ in function of $f(i,j)$. By neglecting the scanner noise $n(i,j,t)$ in Signal Model A (see Expression 5.9), for the pixel value at row i and column j, we have:

$$g(i,j) \approx \frac{s(i,j)}{1+s(i,j)f(i,j)}. \tag{6.3}$$

To simplify the notation, we work only with a single pixel (i,j) and therefore omit the indices. We define the function $l(f)$ and use the standard tangent-line linear approximation at a specifically chosen point a:

$$l(f) \triangleq \frac{s}{1+s.f} \approx k(f-a)+b$$
$$\text{where} \quad k = l'(a) = \left(\frac{s}{1+s.f}\right)'\Bigg|_{f=a} = -\frac{s^2}{(1+s.a)^2} \quad \text{and} \quad b = \frac{s}{1+s.a}.$$

The accuracy of this approximation largely depends on the selection of the point a. As we discussed in Section 5.5.2, for the UPEK swipe scanners, the pixels in an image never saturate. We also observed that the pixel values in an image rarely fall below 100 or even 120. This means that the range of the fingerprint pattern f is rather small. One possible explanation for this is that since the fingertip has to be swiped, pressing it hard enough to produce sufficiently large f (much less to saturate the sensing elements) is difficult. Using Signal Model A (Expression 5.9) with a typical scanner pattern value of 200, we computed that roughly:

$$f \approx \frac{1}{g} - \frac{1}{s} = \frac{1}{1/100} - \frac{1}{1/200} = 0.005.$$

For this reason, we assume that f for the UPEK swipe scanners varies in the range from 0 (no fingerprint) to about 0.005. Selecting a in the middle, i.e., $a = 0.0025$ gives the best overall accuracy of the approximation.

Figure 6.7 shows $l(f)$ and its linear approximation at $a = 0.0025$ for 3 values of s and the corresponding relative errors. Even in the worst case (when $s = 220$), the relative error is at most 12.5%, which happens only near the ends of the range of f.

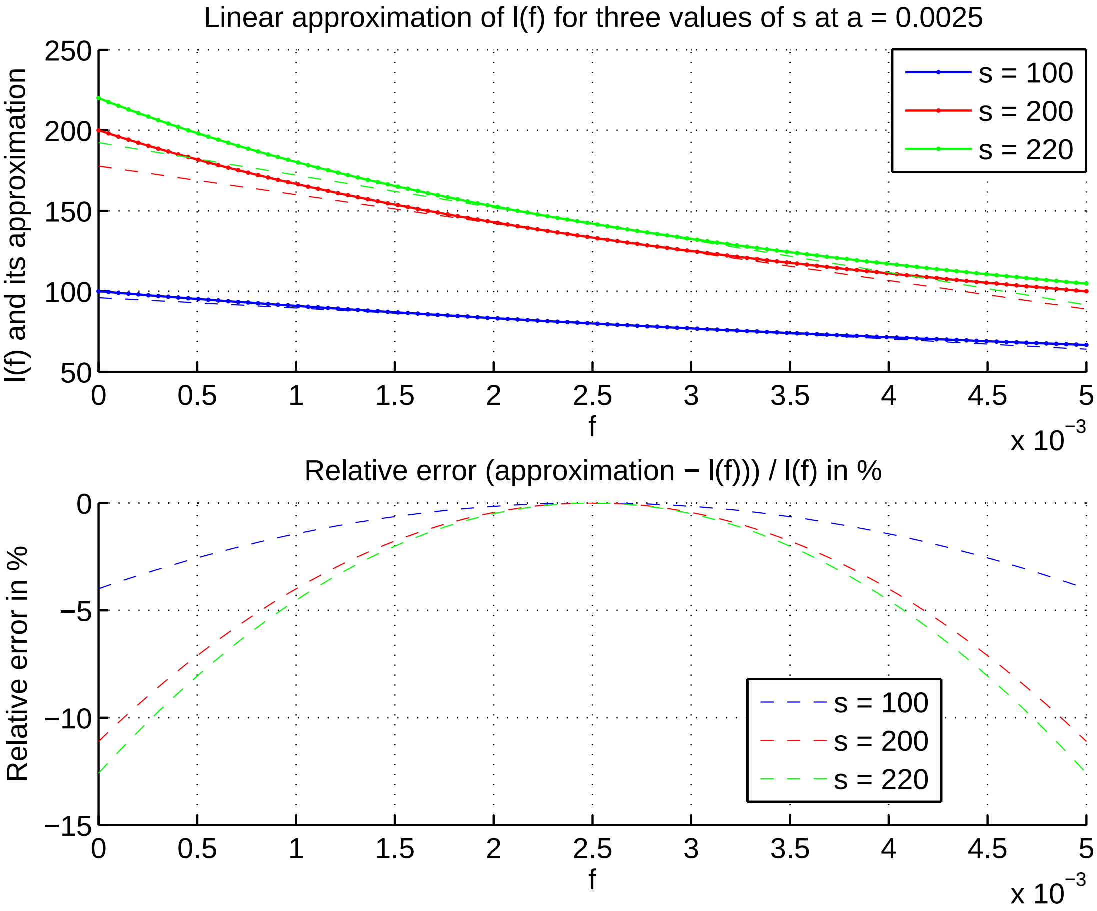


Figure 6.7: Linear approximation for Signal Model A and its accuracy

By reinstating the pixel indices, we obtain:

$$g(i,j) \approx k(i,j)\,(f(i,j)-a)+b(i,j), \tag{6.4}$$

$$\text{where}\quad k(i,j) = -\frac{s^2(i,j)}{(1+s(i,j)a)^2} \quad\text{and}\quad b(i,j)=\frac{s(i,j)}{1+s(i,j)a}. \tag{6.5}$$

6.6.2 Linear Approximation for $s(j)/(1+s(j).a)$

Here we discuss the case specific for the UPEK swipe scanners and for which $s(i,j)=s(j)$. We claim that in certain cases and when properly applied:

$$b(j)=\frac{s(j)}{1+s(j)a} \approx const \cdot s(j). \tag{6.6}$$

Table 6.21 below shows the accuracy analysis of the approximation:

$$\frac{1}{1+s(j)a} \approx \frac{1}{1+\mu_s a} \tag{6.7}$$

for the ranges of parameter values we have in the UPEK swipe scanners. The value of a is taken from the preceding discussion about the approximation for $g(i,j)$, and the scanner pattern standard deviation σ_s is assumed in its very worst case for the approximation. Since 5 is well above the typical standard deviation of the scanner pattern, it can also represent the standard deviation of the sum of the scanner pattern and the scanner noise (see Sections 6.4.3.1 and 6.4.4.1 in this appendix). The choice of the range $\pm 3\sigma$ around the mean μ_s of the scanner pattern does not need explanation. In sum, for the values we have and the ranges of the parameters, $\frac{1}{1+s(j)a} \approx \frac{1}{1+\mu_s a}$ within ±3%, which we believe is sufficiently accurate.

Before claiming the approximation in Expression 6.6, however, we have to stress that this approximation is good only in our specific processing of $b(j)$ for the

Factors and parameters	**Min**	**Typical**	**Max**
$\sigma_s = 5, \quad a = 0.0025 = 1/400$			
μ_s	**100**	**200**	**220**
$1 + \mu_s a$	1.2500	1.5000	1.5500
$1 + (\mu_s - 3\sigma_s)a$	1.2125	1.4625	1.5125
$1 + (\mu_s + 3\sigma_s)a$	1.2875	1.5375	1.5875
$1/\left(1 + \mu_s a\right)$	0.8000	0.6667	0.6452
$1/\left(1 + (\mu_s - 3\sigma_s)a\right)$	0.8247	0.6838	0.6612
% of difference w.r.t. $1/\left(1 + \mu_s a\right)$	**+3.1%**	**+2.6%**	**+2.5%**
$1/\left(1 + (\mu_s + 3\sigma_s)a\right)$	0.7767	0.6504	0.6299
% of difference w.r.t. $1/\left(1 + \mu_s a\right)$	**-2.9%**	**-2.4%**	**-2.4%**

Table 6.21: Linear approximation for $s(j)/(1 + s(j).a)$: accuracy analysis

purpose of extracting the variable part of the scanner pattern $s(j)$, not in general. The reason for this is that the scanner pattern mean μ_s is variable, i.e., it depends on the scanner pattern at index j, and therefore it is $\mu_s(j)$. The scanner pattern $s(j)$ can be represented as its mean $\mu_s(j)$ plus its variable part $s_v(j)$. Therefore, by using Expression 6.7, Expression 6.6 becomes:

$$b(j) = \frac{s(j)}{1 + s(j)a} \approx \frac{1}{1 + \mu_s(j)a} s(j) = \tag{6.8}$$

$$= \frac{1}{1 + \mu_s(j)a} (\mu_s(j) + s_v(j)) = \tag{6.9}$$

$$= \frac{\mu_s(j)}{1 + \mu_s(j)a} + \frac{1}{1 + \mu_s(j)a} s_v(j)). \tag{6.10}$$

We also know that the mean $\mu_s(j)$ is a slowly varying process. Therefore, when $b(j)$ is processed with a denoising filter, the term $\mu_s(j)/(1 + \mu_s(j)a)$ above is removed and the only term produced at the output is the processed scaled variable part $s_v(j)$ of the scanner pattern. Its scaling coefficient, however, still depends on the index j. Nevertheless, we believe that this is not a problem for the advanced algorithms for the swipe scanners as both use filters with very short span (a couple of pixels at most), within which span the mean $\mu_s(j)$ is essentially the same and therefore the scaling factor $1/(1 + \mu_s(j)a)$ is constant for the indices j over which the denoising filters operate at the time of producing the estimate for the variable part $s_v(j)$ of the scanner pattern.

In addition to the previous caveat, it is also important to note that although this approximation in Expression 6.6 seems to be universally applicable to all signals of Signal Model A, actually it may not be. In the UPEK area scanners, the fingerprint pattern $f(i, j)$ can and often do span the whole range $(0, 1]$, especially for thumb fingers, and thus $f(i, j)$ is not bounded to about 0.005 as in the UPEK swipe scanners, potentially making this whole approximation framework fall apart.

In summary, we showed that (a) with the characteristics of our signals and (b) in the context of the denoising algorithms for swipe scanners we propose, the approximation $b(j) \approx \mathit{const} \cdot s(j)$, with *const* varying within only $\pm 3\%$, is accurate. This essentially implies that $b(j)$ can be assumed as being the scanner pattern $s(j)$ it its direct form. Although we did not do a comparative analysis, this approximation may as well be more accurate than the Gaussian approximation for the inverse of the scanner pattern (see Section 6.5 in this appendix).

Bibliography

[Anderson and Darling 1952] Theodore Anderson and Donald Darling, "Asymptotic Theory of Certain 'Goodness-of-Fit' Criteria Based on Stochastic Processes," *Annals of Mathematical Statistics*, vol. 23, no. 2, pp. 193–212, 1952.

[Anderson and Darling 1954] Theodore Anderson and Donald Darling, "A Test of Goodness-of-Fit," *Journal of the American Statistical Association*, vol. 49, pp. 765–769, 1954.

[Bartlow et al. 2009] Nick Bartlow, Nathan Kalka, Bojan Cukic, and Arun Ross, "Identifying Sensors from Fingerprint Images," in *Proceedings of the 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, June 2009.

[Blythe and Fridrich 2004] Paul Blythe and Jessica Fridrich, "Secure Digital Cameras,"in *Proceedings of the Digital Forensic Research Workshop*, Baltimore, MD, Aug. 2004.

[Cappelli et al. 2008] Raffaele Cappelli, Matteo Ferrara, and Davide Maltoni, "On the Operational Quality of Fingerprint Scanners," *IEEE Transactions on Information Forensics and Security*, vol. 3, issue 2, pp. 192–202, June 2008.

[Celiktutan et al. 2008] Oya Celiktutan, Bulent Sankur, and Ismail Avcibas, "Blind Identification of Source Cell-phone Model," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 3, pp. 553–566, Sept. 2008.

[Chen et al. 2008] Mo Chen, Jessica Fridrich, Miroslav Goljan, and Jan Lukas, "Determining Image Origin and Integrity Using Sensor Noise," *IEEE Transactions on Information Forensics and Security*, vol. 3, issue 1, pp. 74–90, March 2008.

[Cox et al. 2006] Ingemar Cox, Matthew Miller, Jeffrey Bloom, Jessica Fridrich, and Ton Kalker, *Digital Watermarking and Steganography*, Second Ed., Elsevier, 2008.

[D'Agostino and Stephens 1986] Ralph D'Agostino and Michael Stephens, *Goodness-of-fit techniques*, Marcel Dekker, New York, 1986.

[Derakhshani et al. 2003] R. Derakhshani, S. Schuckers, L. Hornak, and L. O'Gorman, "Determination of Vitality from a Non-invasive Biomedical Measurement for Use in Fingerprint Scanners," *Pattern Recognition Journal*, vol. 36, no. 2, 2003.

[Drennan 2002] Patrick Drennan, "Device Mismatch in BiCMOS Technologies," in *Proceedings of the 2002 Bipolar/BiCMOS Circuits and Technology Meeting*, pp. 104–111, Sept. 2002.

[Drennan and McAndrew 2003] Patrick Drennan, Colin McAndrew, "Understanding MOSFET Mismatch for Analog Design," *IEEE Journal of Solid-State Circuits*, vol. 38, issue 3, pp. 450–456, March 2003.

[FBI PIV 2006] FBI CJIS, "Personal Identity Verification (PIV) Image quality specifications for single finger capture devices," July 2006. Available online at http://fips201ep.cio.gov/documents/FBI_PIVspec_071006.pdf

[El Gamal and Eltoukhy 2005] Abbas El Gamal, Helmy Eltoukhy, "CMOS Image Sensors," in *IEEE Circuits and Devices Magazine*, vol. 21, issue 3, pp. 6–20, May-June 2005.

[El Gamal et al. 1998] Abbas El Gamal, Boyd Fowlera, Hao Minb , Xinqiao Liua, "Modeling and Estimation of FPN Components in CMOS Image Sensors", in *Proceedings of SPIE*, vol. 3301, pp. 168–177, 1998.

[Filler et al. 2008] Tomas Filler, Jessica Fridrich, Miroslav Goljan, "Using Sensor Pattern Noise for Camera Model Identification," in *Proceedings of ICIP'08*, San Diego, CA, Oct. 2008.

[FotoNation Patent US'218] Eran Steinberg, "Method and Apparatus for In-camera Image Marking and Authentication," US Patent No. 5,862,218, Jan. 19, 1999.

[Fry et al. 1970] Peter Fry, Peter Noble, and Robert Rycroft, "Fixed-pattern Noise in Photomatrices," in *IEEE Journal of Solid-State Circuits*, vol. 5, issue 5, pp. 250–254, Oct. 1970.

[Gassend et al. 2004] Blaise Gassend, Daihyun Lim, Dwaine Clarke, Marten van Dijk, Srinivas Devadas, "Identification and Authentication of Integrated Circuits," in *Concurrency and Computation: Practice and Experience*, vol. 16, issue 11, pp. 1077-1098, Sept. 2004.

[Geller et al. 1999] Boris Geller, Joseph Almog, Pierre Margot, and Eliot Springer, "A Chronological Review of Fingerprint Forgery," *Journal of Forensic Science*, vol. 44, no. 5, pp. 963-968, Sept. 1999.

[Geradts et al. 2001] Z. J. Geradts, J. Bijhold, M. Kieft, K. Kurusawa, K. Kuroki and N. Saitoh, "Methods for Identification of Images Acquired with Digital Cameras," in *Proceedings of SPIE*, vol. 4232, 2001.

[Goljan et al. 2009] Miroslav Goljan, Jessica Fridrich, and Tomas Filler, "Camera Identification – Large Scale Test," in *Proceedings of SPIE, Electronic Imaging, Security and Forensics of Multimedia Contents XI*, San Jose, CA, Jan. 2009.

[Gou et al. 2007] Hongmei Gou, Ashwin Swaminathan, and Min Wu, "Robust Scanner Identification based on Noise Features," in *IS&T SPIE Conference on Security, Steganography and Watermarking of Multimedia Contents IX*, San Jose, CA, Jan. 2007.

[Gou et al. 2009] Hongmei Gou, Ashwin Swaminathan, and Min Wu, "Intrinsic Sensor Noise Features for Forensic Analysis on Scanners and Scanned Images", in *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 3, pp. 476–491, Sept. 2009.

[Harris Interoperability] Ryan Harris, "Enhancing Fingerprint Sensor Image Interoperability." Available online at http://homes.cerias.purdue.edu/ bhargav/cs526/studentProjects/A16.ppt.

[Heise Online 2008] Heise Online, "Chaos Computer Club publishes fingerprints of German Home Secretary", March 31, 2008. Available online at http://www.h-online.com/newsticker/news/item/CCC-publishes-fingerprints-of-German-Home-Secretary-734713.html

[Holst 1996] Gerald Holst, *CCD Arrays, Cameras, and Displays*, SPIE Optical Engineering Press, 1996.

[INCITS M1/06–0424] INCITS M1/06-0424: Study Report on Biometrics in E-Authentication. Available online at
http://www.incits.org/tc_home/m1htm/2006docs/m1060424.pdf.

[ISO/IEC SD 11] ISO/IEC JTC 1/SC 37: Standing Document 11 (SD 11): Part 1 Overview Standards Harmonization Document.

[Ivanov and Baras US'952] Vladimir Ivanov and John Baras, "Method and Apparatus for Authenticating Biometric Scanners," US Patent Application No.12/838,952, July 19, 2010.

[Ivanov and Baras US'907] Vladimir Ivanov and John Baras, "Methods for Authenticating Biometric Scanners," US Patent Application No.61/489,907 (provisional).

[Ivanov and Baras 2011] Vladimir Ivanov and John Baras, "Authentication of Fingerprint Scanners," in *Proceedings of the 36th International Conference on*

Acoustics, Speech and Signal Processing (ICASSP 2011), pp. 1912–1915, Prague, Czech Republic, May 2011.

[Ivanov et al. 2009] Vladimir Ivanov, Paul Yu, and John Baras, "Securing the Communication of Medical Information Using Local Biometric Authentication and Commercial Wireless Links," in *Proceedings of the 14th International Symposium for Health Information Management Research (ISHIMR 2009)*, pp. 119-127, Kalmar, Sweden, Oct 14-16, 2009.

[Ivanov et al. 2010] Vladimir Ivanov, Paul Yu, and John Baras, "Securing the Communication of Medical Information Using Local Biometric Authentication and Commercial Wireless Links," Health Informatics Journal, vol. 16, no. 3, pp. 211-223, Sept. 2010.

[Jain et al. 2006] Anil Jain, A. Ross, S. Pankanti, "Biometrics: A Tool for Information Security," *IEEE Transactions on Information Forensics and Security*, vol. 1, issue 2, pp. 125–143, June 2006.

[Jain et al. 2008] Anil Jain, Patrick Flynn, Arun Ross (editors), *Handbook of Biometrics*, Springer Science+Business Media, 2008.

[Jarque and Bera 1987] Carlos Jarque and Anil Bera, "A Test for Normality of Observations and Regression Residuals," *International Statistical Review*, vol. 55, issue 2, pp. 163–172, 1987.

[Kharrazi et al. 2004] Mehdi Kharrazi, Husrev Sencar, Nasir Memon, "Blind source camera identification," *International Conference on Image Processing (ICIP) 2004*, vol. 1, pp. 709–712, Oct. 2004.

[Khanna et al. 2007] Nitin Khanna, Aravind Mikkilineni, George T. C. Chiu, Jan Allebach, and Edward Delp, "Scanner Identification Using Sensor Pattern Noise," in *Proceedings of SPIE Security, Steganography, and Watermarking of Multimedia Contents IX* San Jose, CA, Jan. 2007, p. 65051K (2007).

[Kinget 2005] Peter Kinget, "Design Mismatch and Tradeoffs in the Design of Analog Circuits," in *IEEE Journal of Solid-State Circuits*, vol. 40, issue 6, pp. 1212–1224, June 2005.

[Kinget 2007] Peter Kinget, "Device Mismatch: An Analog Design Perspective," in *IEEE International Symposium on Circuits and Systems (ISCAS) 2007*, pp. 1245–1248, May 2007.

[Kurosawa et al. 1999] Kenji Kurosawa, Kenro Kuroki, Naoki Saitoh, "CCD Fingerprint Method – Identification of a Video Camera from Videotaped Images," in *Proceedings of the International Conference on Image Processing 1999 (ICIP 99)*, vol. 3, pp. 537-540, Oct. 1999.

[Lee et al. 2004] Jae W. Lee, Daihyun Lim, Blaise Gassend, G. Edward Suh, Marten van Dijk, Srinivas Devadas, "A Technique to Build a Secret Key in Integrated Circuits for Identification and Authentication Applications," in *Digest of Technical Papers, 2004 Symposium on VLSI Circuits*, pp. 176–179, June 2004.

[Lilliefors 1967] Hubert Lilliefors, "On the Kolmogorov-Smirnov Test for Normality with Mean and Variance Unknown," *Journal of the American Statistical Association*, vol. 62, pp. 399402, June 1967.

[Lim 1989] Jae Lim, *Two-dimensional Image and Signal processing*, Prentice Hall PTR, 1989.

[Loftstrom et al. 2000] Keith Lofstrom, W. Robert Daasch, and Donald Taylor, "IC Identification Circuit Using Device Mismatch," in *Digest of Technical Papers of 2000 IEEE International Solid-State Circuits Conference*, pp. 372–373, Feb. 2000.

[Lukas et al. 2005] Jan Lukas, Jessica Fridrich, Miroslav Goljan, "Determining Digital Image Origin Using Sensor Imperfections," in *Proceedings of SPIE Electronic Imaging*, San Jose, CA, pp. 249-260, Jan. 2005.

[Lukas et al. 2006] Jan Lukas, Jessica Fridrich, Miroslav Goljan, "Digital Camera Identification from Sensor Pattern Noise," in *IEEE Transactions on Information Forensics and Security*, vol. 1, issue 2, pp. 205–214, June 2006.

[Maeda et al. 2003] S. Maeda, H. Kuriyama, T. Ipposhi, S. Maegawa, Y. Inoue, M. Inuishi, N. Kotani, T. Nishimura, "An Artificial Fingerprint Device (AFD): a Study of Identification Number Applications Utilizing Characteristics Variation of Polycrystalline Silicon TFTs," in *IEEE Transactions on Electron Devices*, vol. 50, issue 6, pp. 1451–1458, June 2003.

[Maltoni et al. 2003] Davide Maltoni, Dario Maio, Anil K. Jain, and Salil Prabhakar, *Handbook of Fingerprint Recognition*, Springer Verlag, 2003.

[Mihcak et al. 1999] M. Kivanc Mihcak, Igor Kozintsev, and Kannan Ramchandran, " Spatially Adaptive Statistical Modeling of Wavelet Image Coefficients and its Application to Denoising," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 1999)*, Phoenix, Arizona, vol. 6, pp. 3253-3256, March 1999.

[MITRE PIV 2006] MITRE Corporation, "Test Procedures for Verifying Image Quality Requirements for Personal Identity Verification (PIV) Single Finger Capture Devices," December 2006. Available online at http://www.mitre.org/work/tech_papers/tech_papers_07/06_1384.

[NIST NBIS] NIST Biometric Image Software (NBIS). Available online at http://fingerprint.nist.gov/NFIS.

[NIST SP 800–32] NIST Special Publication 800–32: Introduction to Public Key Technology and the Federal PKI Infrastructure.

[NIST SP 800–63] NIST Special Publication 800–63: Electronic Authentication Guideline: Recommendations of the National Institute of Standards and Technology.

[PR Clearninghouse 2010] Privacy Rights Clearinghouse, "Chronology of Data Breaches," August 26, 2010. Available online at http://www.privacyrights.org/500-million-records-breached

[Ratha and Bolle 2004] Nalini Ratha, Ruud Bolle (editors), *Automatic Fingerprint Recognition Systems*, Springer–Verlag New York, 2004.

[Robertson 2009] Jordan Robertson, "Weak Security Enables Credit Card Hacks", The Associated Press, June 14, 2009. Available online at http://www.msnbc.msn.com/id/31331078/ns/technology_and_science-security/t/weak-security-opens-door-credit-card-hacks

[Ross and Jain 2004] Arun Ross and Anil Jain, "Biometric Sensor Interoperability: A Case Study In Fingerprints," in *Proc. of International ECCV Workshop on Biometric Authentication (BioAW)*, Prague, Czech Republic, LNCS vol. 3087, pp. 134–145, Springer Publishers, May 2004.

[Ross and Nadgir 2008] Arun Ross and Rohan Nadgir, "A Thin-Plate Spline Calibration Model For Fingerprint Sensor Interoperability," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 5, Aug. 2008.

[Sencar and Memon 2007] Husrev Sencar, Nasir Memon, "Overview of State-of-the-Art in Digital Image Forensics," Book chapter, Available online at http://isis.poly.edu/ forensics/pubs/sencar_memon_chapter.pdf

[Swaminathan et al. 2007] Ashwin Swaminathan, Min Wu, K. J. Ray Liu, "Non-intrusive Component Forensics of Visual Sensors Using Output Images," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 1, March 2007.

[TCG] Trusted Computing Group. Available online at https://www.trustedcomputinggroup.org.

[Trujillo-Ortiz et al. 2007] A. Trujillo-Ortiz, R. Hernandez-Walls, K. Barba-Rojoi, and A. Castro-Perez, "AnDartest: Anderson-Darling Test for Assessing Normality of a Sample Data," A MATLAB file, 2007, Available online at http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=14807

[UPEK Patent US'381] Alan Kramer," Enhanced Fingerprint Detection," US Patent No. 6,512,381, Jan. 28, 2003.

[Van der Putte and Keuning 2000] Ton van der Putte and Jeroen Keuning, "Biometrical Fingerprint Recognition: Don't Get Your Fingers Burned," IFIP TC8/WG8.8 Fourth Working Conference on Smart Card Research and Advanced Applications, pp. 289-303, 2000.

[Veridicom Patent US'620] Alexander Dickinson, Ross McPherson, Sunetra Mendis, Paul Ross, "Capacitive fingerprint sensor with adjustable gain," US Patent No. 6,049,620, Apr. 11, 2000.

[Wayman et al. 2005] James Wayman, Anil Jain, Davide Maltoni, Dario Maio (editors), *Biometric Systems: Technology, Design and Performance Evaluation*, Springer–Verlag London, 2005.

[Wiehe et al. 2004] Anders Wiehe, Torkjel Sondrol, Ole Kasper Olsen, Frederik Skanderud, "Attacking Fingerprint Sensors," Available online at http://www.olekasper.no/articles/attacking_fingerprint_sensors.pdf.

[Yau et al. 2004] Wei Yun Yau, Tai Pang Chen, and Peter Morguet, "Benchmarking of Fingerprint Sensors," Springer Verlag 2004.

[Yu et al. 2008] Paul Yu, John Baras, and Brian Sadler, "Physical-Layer Authentication," *IEEE Transactions on Information Forensics and Security*, vol. 3, issue 1, pp. 38–51, March 2008.