FUSION OF FINGERPRINT AND IRIS RECOGNITION FOR EMBEDDED MULTIMODAL BIOMETRIC SYSTEM

by

Tariq Mahmood Khan

MACQUARIE
University
SYDNEY·AUSTRALIA

Dissertation submitted in fulfilment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

Department of Engineering
Faculty of Science
Macquarie University
Sydney, Australia

November 2016

ABSTRACT

A multimodal biometric system is considered to be more reliable for person identification. It uses multiple biometric credentials/traits to identify a person rather than a single biometric trait. It uses multiple sensors to acquire biometric traits. This system allows capturing either samples of multiple biometric traits or multiple samples of a single biometric trait. This system improves the accuracy and dependability by providing an optimal False Acceptance Rate (FAR) and False Rejection Rate (FRR).

Hardware implementation of a multimodal biometric system, in resources-constrained embedded systems, poses great challenges. Although there has been a substantial amount of work on combining different biometrics for a variety of purposes, not much work has focused on the hardware implementation of the multimodal biometric system. The aim of this dissertation is to build a reliable multimodal biometric system that takes into account multiple constraints: low cost, real-time processing, hygienic, straightforward, user-friendly, limited memory, etc. To achieve this, we present a hardware architecture of a multimodal biometric system that massively exploits the inherent parallelism.

The proposed system is based on multiple biometric fusions that use two biometric traits, fingerprint and iris. In fingerprint feature extraction, several challenges are addressed that directly affect the minutiae extraction process like fingerprint normalisation, scar removal, orientation estimation, fingerprint enhancement, binarization and thinning and feature extraction. In iris recognition,

each individual block involved in feature extraction is optimised independently, including pupil segmentation, iris segmentation, normalisation and iris feature enhancement. After completing the software design, its hardware equivalent is implemented in VHDL. In both biometric identifiers, each sub-block operates in sequence. For example, in fingerprint identification, first normalisation is performed followed by image enhancement then binarization and thinning and finally feature extraction. This allows the hardware implementation to form a temporal parallelism. The temporal parallelism allows the design to be implemented component by component. Separate processors are used for each component to form a pipelined architecture for both biometrics. Finally, the extracted features are fused with matching-level fusion. To the best of the author's knowledge, no other FPGA-based design that uses these two traits exists to date.

STATEMENT OF CANDIDATE

I certify that the work in this thesis ***"Fusion of Fingerprint and Iris Recognition for Embedded Multimodal Biometric System"*** has not previously been submitted for a degree nor has it been submitted as part of the requirements for a degree to any other university or institution other than Macquarie University.

I also certify that the thesis is an original piece of research and it has been written by me.

In addition, I certify that all information sources and literature used are indicated in the thesis.

. .

Tariq Mahmood Khan

ACKNOWLEDGMENTS

This PhD dissertation would not have been possible without the support and contribution of many extraordinary individuals. Foremost, I would like to thank my supervisor, Dr Yinan Kong, for his mentorship and supervision. He provided me with many helpful suggestions, important advice and constant encouragement during this work.

I would also like to acknowledge the friendly co-supervision provided by Professor Donald G. Bailey and Dr Muhammad Asmit Ullah Khan for my research projects. They helped me at all the critical times I faced during my studies. Their encouragement was vitally important, laying the foundation to learn more. I am grateful to Dr Keith Imrie for his help in proof reading my research papers/thesis. I am also thankful to the admirable staff in the Department of Engineering for their wonderful support.

I wish to acknowledge Macquarie University for awarding me International Macquarie University Research Excellence Scholarship (iMQRES), and providing financial support to attend national and international conferences during my PhD.

I am blessed to have an amazing family; my parents, brothers, and my sister. Their support has been unconditional for all these years. In particular, I am deeply indebted to my parents and my sister who have cherished with me every

great moment and supported me whenever I needed it. Without their everlasting love and esteemed guidance, I would not have accomplished anything in my life. Also, thanks to Taiba for her support and encouragement.

Finally, to Almighty God, for bestowing countless blessings on me and giving me the strength to realise my life-long dream. Thank you so much!

To My Parents

Contents

List of Publications

This thesis is based on the following original publications, which are referred to in the text by Roman numbers. Original publications are reproduced with permission from their copyright holders.

Discussed in Thesis

I M. A. U. Khan, Tariq M. Khan, D.G. Bailey and Y. Kong,“A spatial domain scar removal strategy for fingerprint image enhancement,” *Pattern Recognition*, vol. 60, pp. 258-274, 2016.

II Tariq M. Khan, M. A. U. Khan, Y. Kong and O. Kittaneh, “Stopping criterion for linear anisotropic image diffusion: a fingerprint image enhancement case,” *EURASIP Journal on Image and Video Processing*, (2016) 2016: 6. doi:10.1186/s13640-016-0105-x.

III Tariq M. Khan, M. A. U. Khan, D.G. Bailey and Y. Kong,“A fast and accurate iris segmentation method using an LoG filter and its zero-crossings,” *IEEE Transactions on Information Forensics and Security*, 2016, *in review.*

IV Tariq M. Khan, D.G. Bailey, M. A. U. Khan and Y. Kong,“Efficient Hardware Implementation for Local Feature Enhancement of Fingerprint Images,”*Journal of Real-*

Time Image Processing, accepted in July 2016 (*DOI 10.1007/s11554-016-0625-8*).

V Tariq M. Khan, D.G. Bailey, M. A. U. Khan and Y. Kong,"Efficient Hardware Implementation For Fingerprint Image Enhancement Using Anisotropic Gaussian Filter,"*IEEE Transactions on Image Processing*, accepted in July 2016 (*minor revision submitted*).

VI Tariq M. Khan, Y. Kong and M. A. U. Khan ,"Hardware implementation of fast pupil segmentation using region properties," *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 9534, pp. 95340F-95340F-10, 2015.

VII Tariq M. Khan, M. A. U. Khan, D.G. Bailey and Y. Kong,"Real-time implementation of fast iris segmentation and normalization on FPGA," *IEEE Transactions on Computers*, 2016, *in review.*

VIII Tariq M. Khan, Y. Kong and M. A. U. Khan,"Hardware implementation of Multi-modal biometric using fingerprint and iris," *Machine Vision and Applications*, 2016, *in review.*

Other Publications

These publication are related to this thesis but are not discussed in the thesis.

1. M. A. U. Khan, Tariq M. Khan, O. Kittaneh and Y. Kong, "Stopping criterion for linear anisotropic image diffusion," *Optik-International Journal for Light and Electron Optics* , vol. 127, no. 1, pp. 156-160, 2016.

2. Tariq M. Khan, D. G. Bailey, M. A. U. Khan, Y. Kong, "Real-time edge detection and range finding using FPGAs", *Optik-International Journal for Light and Electron Optics* vol. 126 (17), pp. 1545-1550, 2015.

3. Tariq M. Khan, M. A. U. Khan, Y. kong, “Fingerprint Image Enhancement using Multi-Scale DFB based Diffusion Filters and Modified Hong Filters”, *Optik-International Journal for Light and Electron Optics*, vol 125 (16), pp. 4206-4214, 2014

4. M. A. U. Khan, Tariq M. Khan et al, “Coherence enhancement diffusion using robust orientation estimation”, *Second International Conference on Signal, Image Processing and Pattern Recognition*, pp. 203-214, 2014

5. Tariq M. Khan, Muhammad A. U Khan, Y. Kong, “Boosting CED Using Robust Orientation Estimation”, *The International Journal of Multimedia & Its Applications (IJMA)*, vol, 2(2), pp. 23-34, 2014

6. Tariq M. Khan, Muhammad A. U Khan, “Fingerprint Image Enhancement Using Data Driven Directional Filter Bank”, *Optik-International Journal for Light and Electron Optics*, vol 124(23), pp 6063-6068, 2013

7. Muhammad Talal Ibrahim, Tariq M. Khan, Shahid A. Khan, M. Aurangzeb Khan, Ling Guan, “Iris Localization Using Local Histogram And Other Image Statistics”, *Optics and Lasers in Engineering*, vol 50(5), pp 645-654, 2012

8. Mohammad A. U. Khan, Tariq M. Khan, Rabia Bahadur Khan, Attiqa Kiyani, M. A. Khan, “Noise Characterization in Web Cameras Using Independent Component Analysis”, *International Journal of Computers Communications & Control*, vol.7 (2), pp. 302-311, 2012

9. Tariq M. Khan, M. Aurangzeb Khan, Shahzad A. Malik, Shahid A. Khan, Tariq Bashir, Amir H. Dar, “Automatic Localization of Pupil Using Eccentricity and iris using gradient Based Method”, *Optics and Lasers in Engineering Elsevier*, vol 49 (2), pp 177-187, 2011

10. Muhammad Talal Ibrahim, Tariq M. Khan, M. Aurangzeb Khan and Ling Guan, “A Novel and Efficient Feedback Method for Pupil and Iris Localization”, *Lecture Notes in Computer Science*, vol 6754/2011, pp 79-88, 2011

11. Muhammad Talal Ibrahim, Tariq M. Khan, M. Aurangzeb Khan and Ling Guan, “Automatic Segmentation of Pupil Using Local Histogram and Standard Deviation”, *VCIP Proceedings of the SPIE*, vol 7744, pp. 77442S-77442S-8, 2010.

12. M. A. U. Khan, Tariq M. Khan, W. Al-Halabi, Y. Kong. “Boosting Coherence Enhanced Diffusion with Embedded Orientation Diffusions”, *Opto-Electronics Review*, 2016, *in review.*

List of Contributors

Supervisor Dr. Yinan Kong, Department of Engineering, Macquarie University, Australia

Adjunct Supervisor Dr. Mohammad Asmat Ullah Khan, Biometric and Sensor Lab, Effat University, Jeddah, Saudi Arabia

Adjunct Supervisor Professor Donal G. Bailey, School of Engineering and Advanced Technology, Massey University, Palmerston North, New Zealand

In all the publications discussed in the thesis, I have conducted the major investigations, designs, measurements, data processing, and drafting. Only in Article I, my adjunct Dr. M. A. U . Khan and I had equal contributions. Dr. Yinan Kong, who is my principal supervisor, provided me suggestion, advice and invaluable guidance at every stage of this research. For software design, my adjunct supervisor Dr. M.A.U Khan provided me invaluable guidance. He reviewed, proof-read and corrected all the software based manuscript. My other adjunct supervisor Professor Donal G. Bailey provided support regarding the hardware implementation of fingerprint and iris recognition system. He reviewed, corrected and proof-read all the hardware articles. Table 1 gives a detailed list of contributors.

LIST OF CONTRIBUTORS

DIVISION OF LABOUR IN CO-AUTHORED ARTICLES

TK-Tariq Khan; YK- Yinan Kong; MAK- Mohammad A.U. Khan; DB-Donald Bailey; OK-Omar Kittaneh

	I	II	III	IV	V	VI	VII	VIII
Conception & design	MAK, TK	MAK, TK, OK	MAK, TK	TK, DB	TK, DB	TK	TK	TK, YK
Planning & implementation	MAK, TK	MAK, TK	TK, YK	TK, DB, MAK	TK, DB, MAK	TK, YK	TK, YK	TK
Data collection	MAK, TK	TK	TK	TK	TK	TK	TK	TK
Analysis & interpretation	MAK, TK	MAK, TK	TK	TK	TK	TK	TK	TK
Writing the article	MAK, TK, DB, YK	MAK, TK, YK	TK, MAK, DB, YK	TK, MAK, DB, YK	TK, DB	TK, MAK, YK	TK, DB, MAK, YK	TK, YK MAK
Overall responsibility	YK	YK	YK	YK, DB	YK, DB	YK	YK	YK

Table 1: *Author's Contributions*

List of Figures

List of Tables

Chapter 1

Introduction

In our vastly interconnected society, the demand for verifying the identity of people is increasing day-by-day. The level of heightened concerns about security demands a reliable user authentication. Cataclysmic events like the disappearance of Malaysian airline flight MH370 or events like the 9/11 twin-tower attack have raised concern on the issue of accurate identification in all types of border control environments. Furthermore, the wars and unfortunate rise of violence from the beginning of the 21st century have increased the flight of refugees across borders. The increased criminal activities and the tragedy of human trafficking are pushing governments to evaluate stricter border control security protocols.

To deal with these critical issues, advanced countries have introduced e-passport programs for citizen and traveller identification. These e-passports are based on biometric identification systems. Many countries have a mandatory biometric enrolment of immigrants in immigration processing or border crossing environments. Most of the deployed biometric systems are unimodal and rely on a single biometric credential for authentication. This limits the identification accuracy because of environment variations, signal distortion, lack of individuality, and non-universality (in some cases, due to different pathological conditions; an individual might not be able to provide a particular biometric trait).

A multimodal biometric system is considered to be more reliable for person identification. It uses multiple biometric credentials to identify a person rather than a single biometric trait. It uses multiple sensors to acquire biometric traits. This allows capturing either samples of multiple biometric traits or multiple samples of a single biometric trait. This improves the system accuracy and dependability by providing an optimal False Acceptance Rate (FAR) and False Rejection Rate (FRR).

Most of the existing multimodal biometric systems are computer based. The authentication is performed in an insecure environment that uses a central server for template storage. This can cause a critical information leakage issue. Another disadvantage of a multimodal system is that it requires a large amount of processing as compared to a unimodal biometric system. This makes multimodal systems hard to implement for real-time application, although, in multimodal biometrics, most of the operations are independent. These can not be performed at the same time because of the serial nature of most programming languages, especially the ones used in computers. The implementation of a multimodal biometric system on hardware can address these critical problems.

1.1 Challenges and Objectives

Real-time image processing systems are hard to design, to perform complex operations for a certain task on an image a large set of data is required. A good alternative is the use of hardware design, by prototyping on an FPGA, which offers a compromise between the hardware-based speed of integrated circuit design and the flexibility of general-purpose processors. One of the main advantages of using an FPGA for real-time image processing applications is its ability to exploit spatial and temporal parallelism. FPGA implementations have the potential to be parallel using a mixture of these two forms. For example, in an FPGA, it is easy to partition the image and distribute the partitioned sections to multiple pipelines which could process data concurrently. This is a subjective task that totally depends on the processing mode and hardware constraints of the system.

In the literature, a substantial amount of work is reported on combining different biometrics for a variety of purposes, but not much work focused on real-time multimodal biometric system. One reason is that a real-time embedded system in a resource-constrained environment posses great challenges, as it possesses limited computational resources and limited memory space. On the other hand, most of the existing multimodal biometric systems are computationally very expensive and are not suitable for real-time implementation. Converting the software design to hardware is a very difficult task. For conversion, it must support spatial/temporal parallelism and fixed-point representation. To implement a real-time system, these algorithms need to be optimised for memory usage, fixed-point representation and computational capacity.

In this dissertation, the objective is to implement a multimodal biometric system on an FPGA that is best suited for real-time systems. For this purpose, two biometric traits (fingerprint and iris) are chosen. Most of the problems associated with a software-based system are addressed in this dissertation. For example, in fingerprint identification, scar lines hinder the matching performance of a fingerprint identification system. To compensate for major scar lines, a scar removal strategy is proposed and thoroughly studied. For fingerprint image enhancement, the entropy change for an anisotropic diffusion is investigated. A unique peak is found, associated with blurring of the dominant structure, that provides a reasonable stopping rule for the anisotropic diffusion process. A novel iris segmentation method is proposed that is based on the Laplacian of Gaussian function and its zero crossings. For a software-based system, rigorous studies are conducted using both numerical techniques and experiments. For hardware implementation, the complex blocks/components of both biometrics, fingerprint and iris, are implemented efficiently. A novel and efficient hardware implementation for fingerprint image segmentation and normalisation is proposed. One major hurdle associated with fingerprint filtering techniques is the expensive nature of their hardware implementations. To circumvent this, a modified anisotropic Gaussian filter is efficiently adopted in hardware. In the proposed structure, for a middle-range reconfigurable FPGA, both parallel computation-intensive and real-time goals were achieved. Moreover, this dissertation presents a real-time implementation of fast iris segmentation and normalisation on an FPGA. Finally, a real-time FPGA-based multimodal biometric system is designed that combines the matching scores of both fingerprint and iris.

1.2 Main Contributions

In this dissertation, a novel hardware architecture has been presented for multimodal biometric authentication systems. Our design is based on two biometric traits, fingerprint and iris. Each biometric trait is first optimised at the software level, by addressing some of the issues that directly affect the FAR and FRR. Then the hardware architectures for both biometric traits are presented, followed by a final multimodal hardware architecture. The proposed fingerprint normalisation and image-enhancement structure is the fastest, most cost-effective and most efficient one as compared to most existing FPGA-based structures. Moreover, the following are some of the key scientific contributions:

- To deal with scars and false conglutinated ridges at the same time, a better strategy is suggested in the form of first using a scar removal strategy to fill big linear scar cuts, spread across the surface of the image. Then a follow-up smoothing performed with a Curved Gaussian filter (a variant of a Curved Gabor Filter). This significantly improves the Equal Error Rate (EER) to 3.8 for a scared database.

- A reasonable stopping rule for the anisotropic diffusion process is presented that uses the entropy change for an anisotropic diffusion of a fingerprint image. This significantly reduces the overall computational complexity of the anisotropic diffusion process.

- A fast and novel method for pupil segmentation is developed based on a shape detector and an intensity-based threshold. The use of a LoG filter followed by region growing gives an accurate estimate of the pupil centre and radius. It achieved almost 99% accuracy on most of the iris databases with on average 0.8 seconds computational time. Our proposed method is computationally less expensive in achieving this state-of-the-art iris segmentation accuracy than the similar existing methods.

- The proposed hardware structure for fingerprint image normalisation manages to efficiently speed up the image-processing time by a factor of 18 and improve some of the resource utilisation of the FPGA over the best existing method.

- A correction factor is introduced that effectively segments the foreground segmentation and compensates for the division in the background regions. The proposed structure not only enhances the EER but also reduces the stress of subsequent processes like image enhancement and thresholding.

- A new anisotropic diffusion method is presented, modifying the traditional Gabor filter and decomposing it into an isotropic and an anisotropic Gaussian filter. By this decomposition, it manages to efficiently speed up the image-processing time by a factor of 9 and improve the resource utilisation of the FPGA over the best existing FPGA structure.

- The author managed to develop a single-pass fingerprint-minutiae-extraction structure for an FPGA without using any external memory (SRAM/DRAM). As far as the author know, no other structure is based on a single-pass process.

- A real-time implementation of iris segmentation is proposed that fully utilises the parallel and pipelined architecture of an FPGA.

1.3 Dissertation Outline

This dissertation follows the non-traditional "Thesis-by-Publication" format which has been approved by the *Macquarie University Higher Degree Research Office*. It consists of a general introduction, background, and a list of the PhD candidate's major scientific publications. The thesis materials are the original texts and graphics of author's publications, published or under review, that have been reformatted to improve readability. The outline of the thesis is summarised in Fig. 1.1. There are two main components of the proposed design: software and hardware. In the software design, some of the issues are addressed that affect the recognition rate of these two biometric. Then, in hardware design, both parallel computation-intensive and real-time goals were achieved.

Chapter 2 gives a brief overview of unimodal and multimodal biometric systems. A comprehensive review of the literature available on the approaches and methodologies and the hardware designs of multimodal system is included in this chapter, followed by the potential applications and a summary of the proposed design.

In Chapter 3, a scar-removal strategy for fingerprint image enhancement is presented. A fingerprint recognition system can create problems if the sensor is unable to capture a good fingerprint image due to uneven light effects or if scars or cuts are present on the finger tip. This directly affects the performance of the system. To overcome these, an

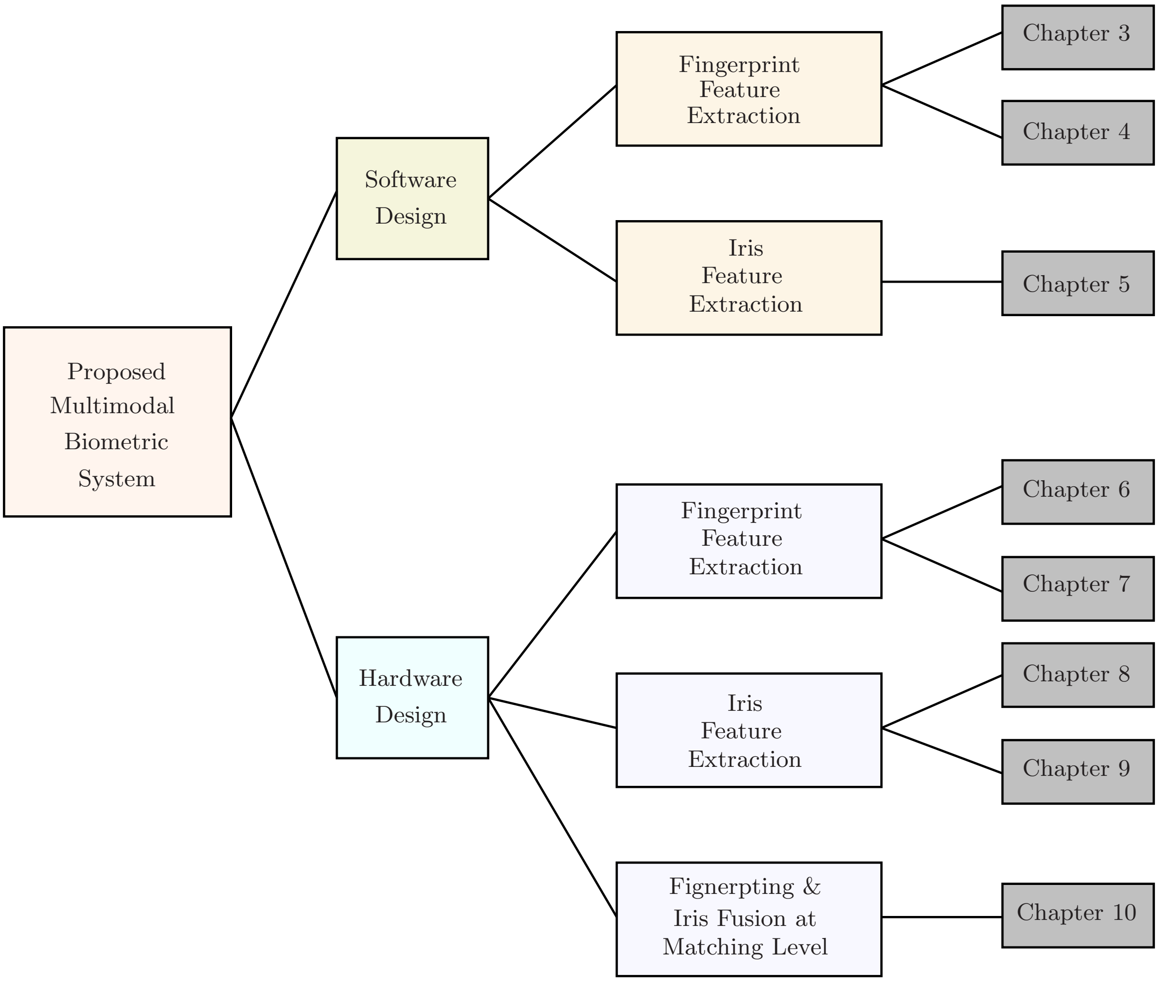


Figure 1.1: *Thesis outline*

explicit filling process is proposed that is a mix of Fourier and spatial-domain strategies. In the proposed method, a Fourier-domain directional field is used to trace an appropriate candidate for the scar pixels to be replaced with. There are four major contributions: an efficient technique for background homogenisation that tackles uneven light effects, the process of locating scars, orientation field estimation by oriented diffusion, and the scar filling strategy. This process can act as a front end to the subsequent Gabor and anisotropic diffusion filtering.

Fundamentally, the anisotropic diffusion process is an iterated one, that starts with a poor-quality image, and converges to a completely blurred mean-value image, with no significant structure left. Though the process starts by doing a desirable job of cleaning noise and filling gaps, called under-smoothing, it quickly passes into an over-smoothing phase where it starts destroying the important structure. Chapter 4 addresses this concern of finding the boundary between the under-smoothing and over-smoothing regions. To find the best stopping criterion, a spatial entropy change is used as a measure that provides important clues to describe that boundary. This not only enhances the performance of the diffusion process but also minimise the computational cost.

Accurately segmentation and normalisation of the iris image plays a vital role towards the accuracy of the iris recognition system and is a computationally intensive task. In Chapter 5, a hybrid approach is presented to achieve iris localization based on a Laplacian of Gaussian (LoG) filter, region growing, and zero crossings of the LoG filter. The use of LoG as blob detector along with its zero crossings makes the iris segmentation process fast, robust and computationally less expensive, and best suited for real-time implementation.

In Chapter 6, a modified fingerprint local normalisation is proposed that enhances the contrast of the foreground ridge/valley area uniformly. Low-pass Gaussian filtering is employed to estimate the local mean and variance. Generally, using a Gaussian filter gives the best results because the smooth transition can minimise artefacts (it has good stop-band performance in the frequency domain). To circumvent the amplification of background noise, a multiplication correction factor is introduced that is a monotonically increasing function of local variance values. The hardware structure for fingerprint image normalisation manages to efficiently speed up the image-processing time by a factor of 3 to 4 and improve some of the resource utilisation of the FPGA over the best existing hardware structure.

One major hurdle associated with fingerprint filtering techniques is the expensive nature of their hardware implementations. To circumvent this, in Chapter 7, a modified anisotropic Gaussian filter is efficiently adopted in hardware. In the proposed structure, we address several challenging problems for real-time fingerprint image enhancement. As far as the author knows, the proposed structure is the fastest, most cost-effective and most efficient one as compared to most existing FPGA-based structures.

Chapter 8 introduces a novel approach for automatic pupil segmentation. The proposed algorithm uses a local histogram-based threshold, area and eccentricity that looks for the region that has the highest probability of having the pupil. In order to implement it on an FPGA, a parallel technique is used in which the properties of regions are calculated simultaneously using an efficient connected-component method. This formulation makes this technique much faster and more efficient than the existing one. Later, a real-time iris segmentation technique is described, that results in a faster implementation of iris segmentation and normalisation on an FPGA.

One major hurdle associated with iris segmentation techniques is the use of iterative processes that lead to expensive hardware implementations. To circumvent this, Chapter 9 introduces a threshold of the signed image obtained from the background subtracted image along with morphological operators to localise the pupil. This makes the iris segmentation process improved based on the previous approach presented in Chapter 8.

The outer boundary is located by first normalising a selected image region that contains the iris, and then using a first-order gradient operator. This makes the implementation of the parallel and pipelined architecture of an FPGA easier and more efficient as compared to other existing algorithms.

In Chapter 10, these two traits are efficiently combined in a matching score-level fusion and the hardware architecture is presented. Concluding remarks are drawn in Chapter 11 and a non-exhaustive list of future research directions is included.

Chapter 2

Background and Related Work

This chapter provides background knowledge regarding multimodal biometric approaches and methodologies, hardware implementation and potential applications. An introduction to the unimodal biometric system is provided followed by its limitations. Then, the importance of a multimodal biometric system along with its working process is discussed. The limitations of existing software-based multimodal systems and the difficulties of implementing in hardware addressed in this dissertation are discussed. Finally, the chapter concludes with a summary of designs proposed in this dissertation.

2.1 Biometric

Biometric technology deals with measurable, albeit distinctive, characteristics that are used to label and to a large extent describe individuals [1]. These characteristics are often categorised as behavioural versus physiological. Physiological ones deal with the shape/propert of the body. Examples include DNA, face recognition, palm print, retina, hand geometry, iris recognition and fingerprint. Behavioural ones recognise people's behavioural patterns such as voice print or typing rhythm. Biometric-based systems are becoming the foundation of highly secured personal identification and verification solutions. A biometric can not be easily transferred between individuals, therefore it is able to provide personal data privacy and confidential financial transactions. If the verification procedures are made more user-friendly then the scalability for integrating biometrics can be extended to a variety of processes. Biometric-based authentication applications include network, workplace, single sign-on, entry access, data safeguarding, application login, web security, transaction security and remote access to resources. A biometric authentication is considerably more accurate and convenient than the conventional methods (e.g. Usage of personal identification number or passwords). The use of biometric nullifies the need to remember or carry any password or PIN. The inexpensiveness and rising popularity of such methods make the technology more acceptable.

2.1.1 Limitations

In recent years, biometric-based identification systems have been used in a number of real-world applications (e.g. defence establishments, airports, banks, amusement parks, etc.), as they offer reasonably good performance [2]. To date, even the most advanced biometric systems face numerous problems. A variety of factors are associated with these problems, including the algorithms or data used and the system design [3]. The following are the factors that limit the use of a biometric system:

1. Lack of universality: Although all biometric features are universal, every person does not possess the same physical features [4]. Due to the complexity and wide variety of the human body, it is possible that a person might not contain all the biometric features. In such cases, a biometric system will fail to recognise that person;

2. Trouble with data sensors: The input data from the sensor are often affected by noise, including several that include either user physical and physiological conditions (cut fingers, cold, etc) or the environmental conditions (power, insufficient light, etc.). The use of improperly maintained or defective sensors produces noisy biometric data that can result in poor recognition [5] ;
3. Distinctiveness ability: Most of the biometric features do not have the same distinctiveness degree (for example, fingerprint-based biometric systems are more selective than hand-geometry-based ones) [6];
4. The limitation of the discrimination of biometric systems due to a low inter-class and high intra-class variability. Large intra-class variation-based biometric datasets result in a lower recognition performance [7];
5. At a certain level, the recognition performances of the systems have an upper limit;
6. Unacceptable error rates for the unimodal biometric systems;
7. In biometrics, a fraud is possible through the cloning of a biometric trait, voluntarily or involuntarily. For example, in fingerprinting gummy fingers can be used to deceive a biometric system [8] ;

2.2 The requirements of multimodal biometric system

Multimodal biometric systems are a recent approach that was developed to overcome the problems of the unimodal biometric systems [9]. These systems demonstrate significant improvements over unimodal biometric systems, in higher accuracy and higher resistance to spoofing. A multimodal biometric system aims to reduce one or more of the following:

- Susceptibility to artefacts or mimics
- Failure- to-enrol rate (FTE)
- False acceptance rate (FAR)
- False rejectance rate (FRR)

There are several more requirements, such as

- Use of multiple traits/characteristics makes the multimodal system more reliable [10].
- A multimodal biometric system increases the secrecy and security of user data [11].
- The system still can provide security by employing another identifier even if any of the identifiers fails to work for any known or unknown reasons [12].
- In a multimodal biometric system, fusion strategies are conducted to combine decisions from each subsystem and then come up with a conclusion [12]. This makes a multimodal system more accurate.
- In multimodal systems liveliness detection techniques can be applied to provide knowledge about the liveliness of the sample being entered [13]. This makes them capable of detecting and handling spoofing [14].

2.3 Operation of Multimodal Biometric System

Like a unimodal biometric system, a multimodal biometric system operates in the following two modes:

- Identification

- Verification

Identification is the process that determines the identity of an unknown person. In identification mode, the input sample data is compared with subjects stored in the database. *Verification* is a process that accepts or rejects an identity claimed by a person. In verification, the input sample data is compared to the data of the claimed identity, either to accept or reject the claim. Verification is also known as authentication. The focus of this dissertation is on authentication. A multimodal biometric system has similar modules to a conventional unimodal system, like:

- Feature extraction module
- Decision-making module
- Capturing module
- Comparison module

A multimodal biometric system uses a fusion technique to integrate the information from two different authentication systems. Fig. 2.1 shows the block diagram of a multimodal biometric system. Fusion can be done:

- During feature extraction;
- During decision making;
- During the comparison of live samples with stored biometric templates.

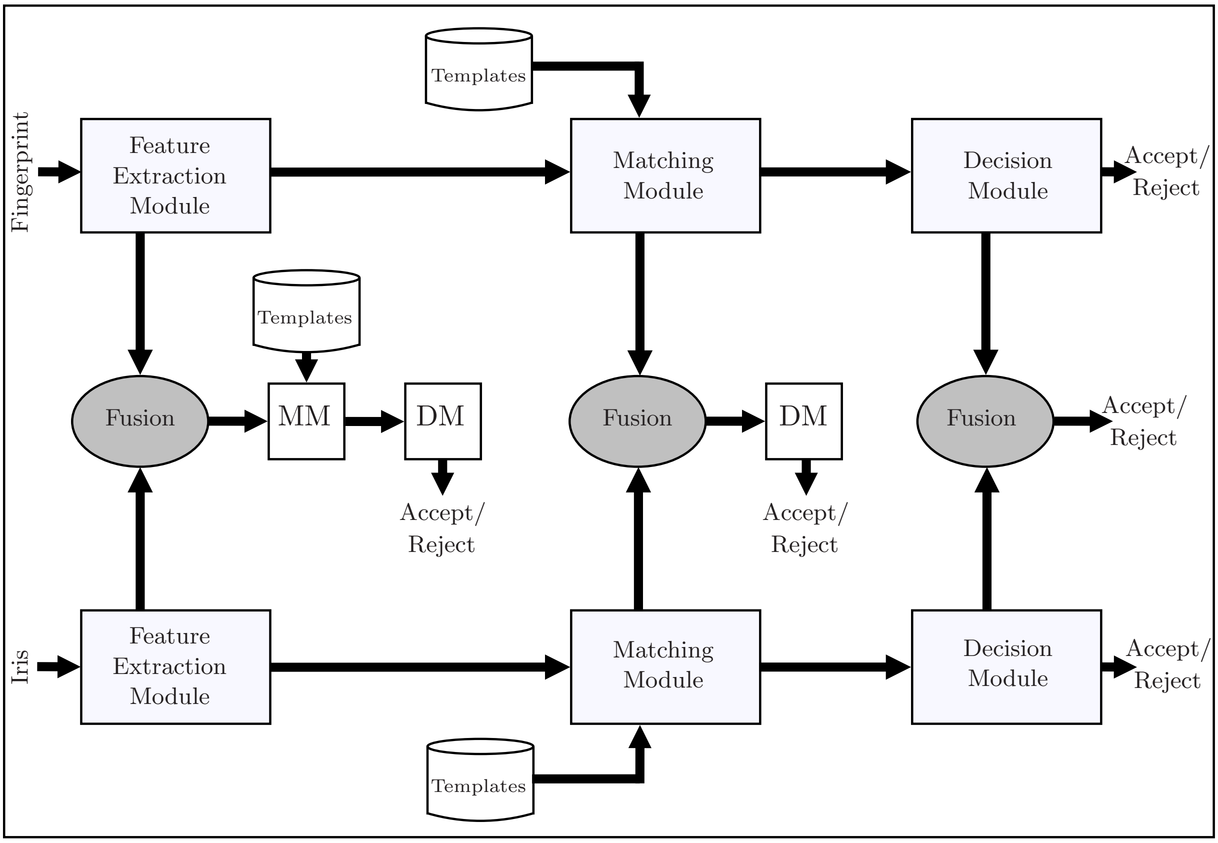


Figure 2.1: *Block diagram of a multimodal biometric system. The Fig shows three level of fusion: at feature extraction module (FEM), at matching module (MM), or at decision module (DM).*

2.4 Biometric Fusion

In multi-modal biometrics different methodologies are used to combine features of multiple biometrics. Normally, these are divided into two broader categories: Forms of fusion and Fusion levels.

2.4.1 Forms of Fusion

Fusion in the context of biometrics can take the following forms:

- Single biometric, multiple representations.
- Single biometric, multiple matchers.
- Multiple biometric fusion.
- Hybrid system.

Single Biometric, Multiple Representations

This type of fusion uses a single biometric indicator with multiple representations. Each representation possesses its own classifier. The similarity scores that are reported by these classifiers are then merged. For example, the single biometric indicator fingerprint represented by multiple prints. Each fingerprint has its own classifier that generates a matching score.

A fingerprint classification system proposed by Cappelli et al. [15] combines a KL transform-based classifier with a structural classifier. The generated scores are integrated by the two classifiers. A double-sigmoid function is used to map the scores into a common domain and then a weighted average is taken in the new domain. Multiple classifiers for fingerprint indexing are also used by Jain et al. [10]. Their technique uses a set of 10 neural network classifiers and a k-nearest neighbour classifier to classify fingerprints.

[16,17] suggested general strategies for combining multiple classifiers. All the approaches presented in [16] attempt to reduce or re-rank a given set of classes. These are relevant to the identification problem in which a large number of classes (identities) are present. In this type of approach, fusion takes place at the matching stage. To avoid performance degradation, the classifiers have to be carefully selected [18].

Single Biometric, Multiple Matchers

A single-biometric multiple-matcher fusion technique incorporates multiple matching strategies in the matching module of a biometric system and combines the scores generated by these strategies [19,20]. A logistic function used by Jain et al. [21] maps the matching scores obtained from two different fingerprint matching algorithms into a single score. Such an integration strategy improved the overall performance of a fingerprint verification system. This type of fusion can also take place at the matching stage [22]. In this case there should be multiple matchers that operate on the same biometric data representation.

Multiple Biometric Fusion

Although single-biometric multiple representations and single-biometric multiple matchers improved the accuracy of biometric systems over single biometrics systems, they still have some disadvantages, as they use only a single indicator of biometric. To further improve the accuracy and security of a multi-modal biometric system, multi-biometric fusion is introduced. Multi-biometric fusion refers to the fusion of multiple biometric traits. This kind of system seeks to improve reliability (accuracy) and the speed of a biometric system, by integrating matching scores obtained from multiple biometric sources [23]. In the literature, to combine these various sources a variety of fusion schemes have been described. These include sum and product rules [24], Support Vector Machine (SVM)

[25] , majority voting [26], K-NN classifiers [27], Bayesian methods, decision trees [28] . In this dissertation, we work on multi-biometric fusion rather than other fusion schemes.

At the matching score level, the normalisation of the scores obtained from the different domain experts is an important aspect that has to be addressed in the fusion process [29]. The scores obtained from multiple domains are mapped into a common domain before combining them. This is a two-step process. In the first step, a robust statistical technique is used to estimate the distributions of scores for each domain. In the second step, these distributions are then translated or scaled into a common domain [30].

Hybrid Systems

For robust authentication, in hybrid systems, more than one of the scenarios discussed above are used [31, 32]. For example, a hybrid biometric system may use three face-matching algorithms and two iris-matching algorithms in one iris-and-face based multi-modal biometric system [33].

2.4.2 Fusion Levels in Biometrics

Fusion is a broad term that is not limited to biometrics only. In the field of statistical pattern classification, biometric fusion is considered as a special case of combining multiple classifiers. Fusion methods are used in diverse fields like analysis of medical test results, analysis of satellite imagery, and internet search engines. Biometric fusion is not a new idea. In large-scale fingerprint systems, for years, various aspects of fusion have been an integral part of the successful implementation of biometric systems. Normally, in biometrics there are three levels of fusion: one is decision level, the second one is score level, and the third is feature level. Mostly in multiple biometric fusion either decision-level fusion or feature-level fusion is used. The system designed in this dissertation used decision level fusion.

Sensor-Level fusion

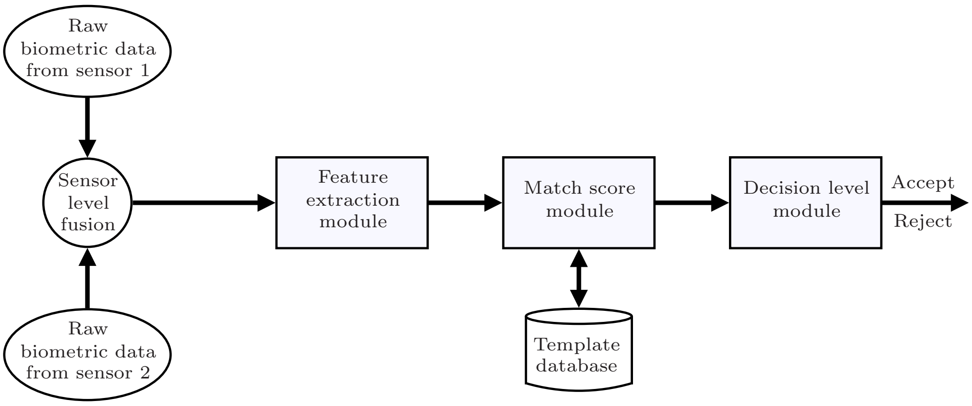

Raw biometric data from sensor 1
Sensor level fusion
Raw biometric data from sensor 2
Feature extraction module
Match score module
Decision level module
Accept
Reject
Template database

Figure 2.2: *Sensor level fusion*

Fusion at sensor level combines the raw data from various sensors Ross [31], as shown in Fig. 2.2. This type of fusion is best suited for multi-sensor and multi-sample systems. In sensor-level fusion, the multiple modalities must be estimated accurately or known in advance. In the raw data, these multiple modalities must be compatible with the feature level. For example, in face recognition, to generate a 3D texture image of a face, 2D texture information with a 3D depth information can be fused. This fused image can to be utilised for both feature extraction and matching Hsu [34]. Liu and Chen [35] used a mosaic method to combine multiple instances of faces captured using a single camera to obtain better recognition performance. Raghavendra [36] used particle swarm optimisation (PSO) to combine information obtained from palmprint and face image. For feature extraction and classification, nearest neighbour and Kernel Direct Discriminant Analysis (KDDA) are used.

Feature-Level Fusion

Feature-level fusion combines different feature sets extracted from multiple biometric modalities into a single feature vector, as shown in Fig. 2.3. The features extracted from different biometric traits can be concatenated to a single vector, as they are independent of each other. The combined feature vector possess a higher dimensionality and represents a person's identity in a different hyperspace [31]. Feature reduction techniques can be used to extract useful features from the larger set of features. Concatenation is not possible for feature sets that are not compatible, e.g. eigenface coefficients and fingerprint minutiae.

Feng et al. [37] developed a system by using two biometric traits called palmprint and face. For fusion, a concatenation method is applied using a feature-level fusion technique. Two feature-extraction approaches, Independent Component Analysis (ICA) and Principal Component Analysis (PCA), are used to compare the recognition performance. In both multimodal and unimodal validation frameworks, ICA performed better than PCA. A hand-based verification system proposed by Kumar et al. [38] combines geometric features of the hand with palmprint at the feature-level, and match-score level. Their experimental results show that match-score-level fusion performs much better than feature-level fusion. Fierrez-Aguilar et al. [39] used a max and sum rule for fusing the local and global features of the signature. Y. Wang et al. [40] used iris and fingerprint traits of an individual for their multimodal biometric system. Their method is based on

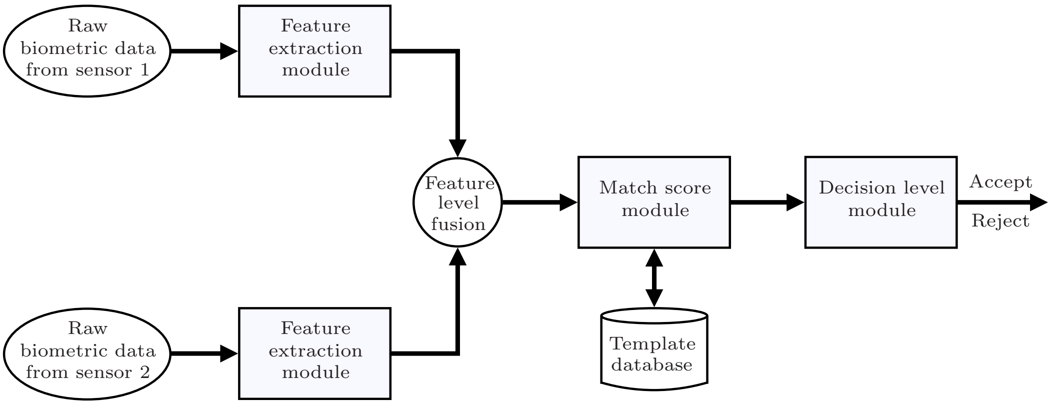

Raw biometric data from sensor 1
Feature extraction module
Feature level fusion
Match score module
Decision level module
Accept
Reject
Template database
Raw biometric data from sensor 2
Feature extraction module

Figure 2.3: *Feature-level fusion*

two strategies. In the first, they compute the weighted and un-weighted sum, and then Fisher's discriminant analysis and a neural network are used with a radial basis function. X. Lu et al. [41] used three different types of feature sets of face images. Three classifiers, namely Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA) and Independent Component Analysis (ICA) were integrated at the match-score level. For fusion, two strategies, radial base function and sum rate, were used. Ross and Govindarajan [42] used RGB colour channels of the face images to describe an intra-modal fusion. Linear Discriminant Analysis is independently used on each colour channel. The resultant feature sets were fused at both match-score and feature levels.

Matching-Score Level Fusion

In matching-score level fusion, the similarity score provided by a matching module, with input features and a template feature vector, is combined, as shown in Fig. 2.4. This method is also known as confidence-level fusion or measurement-level fusion. These techniques attempt to minimise the FRR for a given FAR [43]. Matching-score fusion can be classified by the two different approaches on the basis of match-score processing, either by combining the feature vector or by classifying the feature vector [44]. In match-score level fusion, the normalisation of the match score plays a vital role, because of the dissimilar match score that is generated by the multiple modalities. In the literature, various normalisation techniques have been proposed. L. Hong and A. K. Jain [44] fused fingerprint and face traits at the match-score level for person identification. To further improve the accuracy, they use a normalisation technique. A. Ross et al. [45] proposed a hybrid fingerprint recognition system that consists of minutiae and ridge flow information for fingerprint matching. Then the generated scores are combined with the match-score level fusion. Two different fingerprint acquisitions are used by Conti et al. [46] in their proposed multimodal biometric system. The matching module integrates fuzzy-logic methods for matching-score fusion. Their experimental results have shown an improvement of 6.7% over a unimodal authentication system. Yang and Ma [47] used palm print, fingerprint and hand geometry for personal identity verification. Unlike other multimodal biometric systems, these three biometric features can be taken from the same image. To establish identity, matching-score level fusion is implemented at different levels. First, the fusion of the palmprint and fingerprint features is performed, followed by a matching-score fusion between the palm-geometry unimodal system and the multimodal system. The system was tested on a database self-constructed for 98 subjects. Subbarayudu and Prasad [43] present experimental results of a unimodal palmprint system, a unimodal iris system and a multibiometric system (palmprint and iris). Each system provides a matching score that indicates the similarity of the feature vector with the template vector.

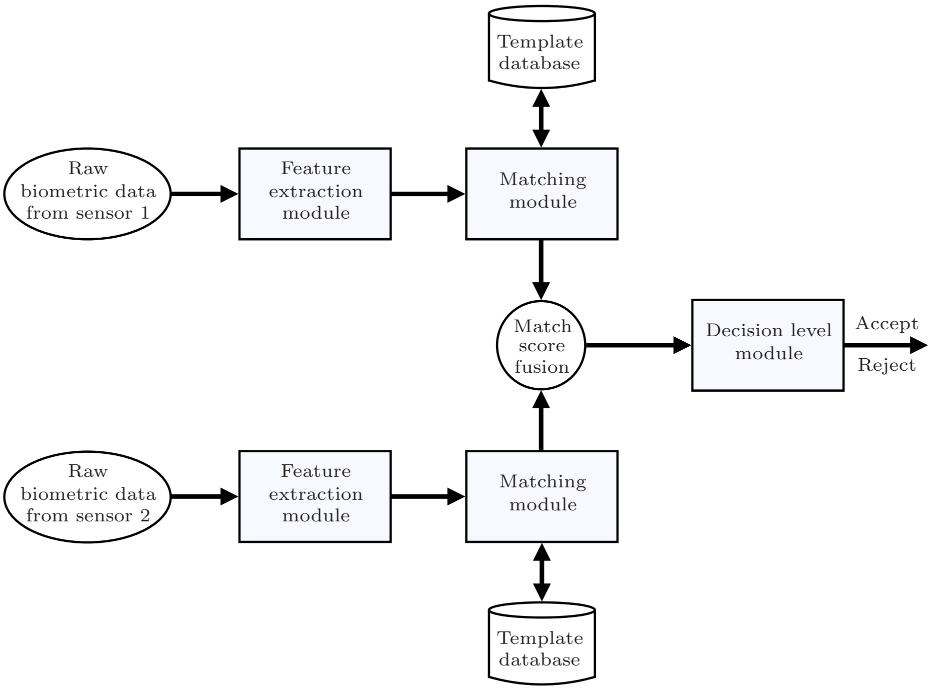


Figure 2.4: *Matching score level fusion*

Decision-Level Fusion

In decision-level fusion, decision thresholds are first estimated for each classifier from the matching score. The different biometric data is integrated at a later stage than the multi-biometric system using feature level fusion or matching score level fusion strategies, as shown in Fig. 2.5. Various techniques are used to obtain a consolidated decision including majority voting, AND/OR, decision table, weighted majority voting, Dempster-Shafer theory of evidence and Bayesian decision. Decision-level fusion operates only on binary information, therefore it is less sophisticated than other fusion methods [31]. Prabhakar et al. [18] proposed a multi-algorithm system that combines one texture-based classifier (match filter) and three minutiae-based algorithms (string, Hough and dynamic base) at the decision-level for fingerprint verification. To combine the biometric matching scores, rank-level fusion has been rarely used. Yu et al. [48] combined finger geometry, fingerprint and palmprint at the decision-level fusion. To perform fusion, three fusion rules, including majority voting, AND rule, and OR rule, are employed.

2.5 Hardware Designs of Multimodal Biometric

One of the disadvantages of a multimodal system is that it requires a larger amount of processing than a unimodal biometric system. As the biometric algorithms work on uncompressed images, a large number of operations are required for processing. This makes a multimodal system hard to implement as an embedded system, especially for a real-time application, although, in a multimodal biometric, most of the operations are independent. Because of the serial nature of most programming languages, especially the ones used in computers, these biometric operations can not be performed at the same time. The implementation of a multimodal biometric system in hardware can address these critical problems.

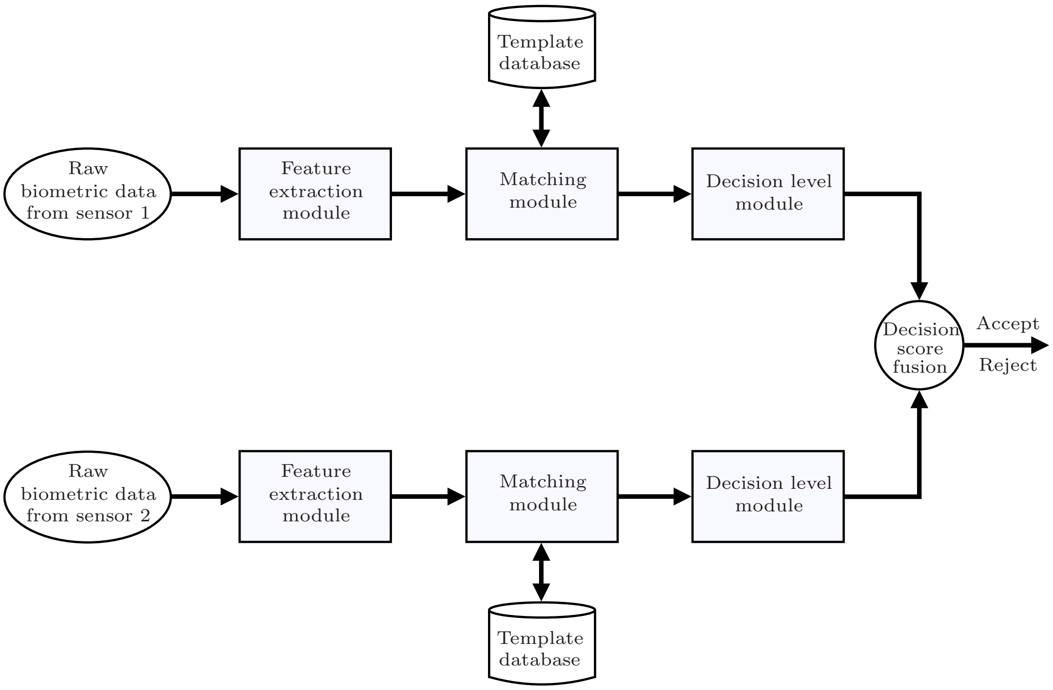


Figure 2.5: *Decision level fusion*

From the literature, it is found that only few multi-modal biometric systems are implemented as embedded systems. One reason is that a real-time embedded system in a resource-constrained environment poses great challenges, as it possesses limited computational resources and limited memory space. On the other hand, most of the existing multimodal biometric systems are computationally rather expensive and are not suitable for real-time implementation. Converting the software design to hardware is one of the most difficult tasks. Therefore it is least developed, more so with fingerprint and iris multimodal biometrics.

Sonal et al. [49] implement a palm-vein identification system in hardware. For hardware implementation a Blackfin ADSP-561 processor is used, whereas the C language is used for the algorithms used for matching of palm veins. Template matching and principal component analysis (PCA) are used as verification algorithms for palm veins and are integrated at match-score level. Yoo et al. [50] have developed two DSP systems for face-fingerprint and iris-fingerprint recognition. In their system, the most computationally expensive tasks are implemented on an FGPA in order to increase the system speed. They used a Xilinx XC3S4000 on-board FPGA and an ARM920T DSP clocked at 400 MHz, and a 128 MB SDRAM. However, no fusion strategy was applied in the embedded biometric system.

Audrey et al. [51] propose a contactless multimodal biometric system that combines two modalities: face and palmprint, by using fusion at the score level. This hardware architecture has been implemented on DSP and FPGA. Wang. J et al. [52] proposed a multimodal biometric system that implements fingerprint and voiceprint. Matching-score level fusion was applied to voiceprint and fingerprint. They used an ARM9-Core based S3C2440A microprocessor that works at 400 MHz and the Microsoft Windows CE operating system. R. Moganeshwaran et al. [53] used fingervein and fingerprint for their multimodal biometric system. Two biometric traits, fingervein and fingerprint, are used and the whole process is implemented in SOC FPGA. The biometric fusion strategy applies at the matching-score level. Conti et al. [54] proposed a multimodal technique for an embedded fingerprint recogniser. In this technique, fingerprint minutiae points along with fingerprint singularity points are used for robust user authentication. For biometric fusion a matching-score fusion module is used.

2.6 Potential Applications

Automated methods are required by the defence and intelligence communities that are capable of rapidly determining an individual's true identity. They should identify previously used identities and past activities. A law-enforcement community and homeland security require technologies to identify criminals in the civilian law-enforcement environment and to secure borders. Key applications include an interface for criminal and civil applications, border management, and first-responder verification.

Enterprise solutions require an oversight of technologies, processes and people. These days, the network infrastructure is playing a critical role to functions of business, government, and web-based business models. Ensuring one's identity and securing access to these systems are essential. On the other hand, business transactions and personal information require secure, cost-effective and user-friendly solutions to prevent a fraud. Key application areas include online customer verification, customer verification at the physical point of sale, etc.

The following Application characteristics drive the need for multi-modal biometrics:

- Usage Environment Suitability
- Risk and Viability of Spoofing
- Integrity/Accuracy Requirements
- Universal Enrolment Requirements
- Transaction Time Flexibility

The target applications of multi-modal biometrics are:

1. Strong Potential for Multi-modal Solutions
   (a) Civil ID
   (b) Physical Access
   (c) Criminal ID
2. Moderate Potential for Multi-modal Solutions
   (a) ATM/Kiosk
   (b) PC Access/Network

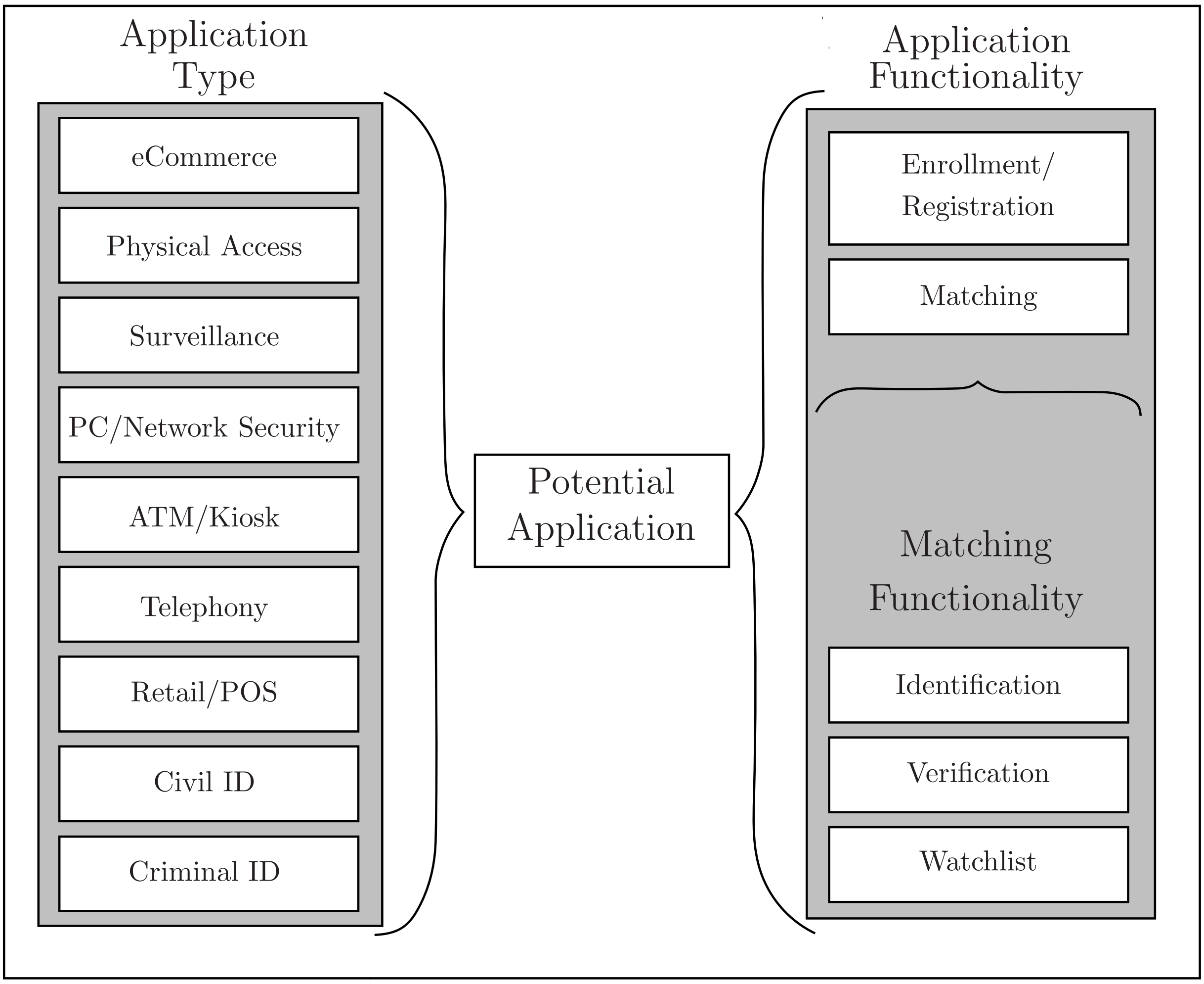


Figure 2.6: *Potential Applications*

3. Modest Potential for Multi-modal Solutions

   (a) POS/Retail

   (b) eCommerce

   (c) Surveillance

   (d) Telephony

2.7 Summary of Proposed Design

The aim of this dissertation is to build a reliable multimodal biometric system that respects multiple constraints: low-cost, real- time processing, hygienic, straightforwardness, user-friendliness, limited memory, etc. To achieve this, we present a hardware architecture of a multimodal biometric system that massively exploits the inherent parallelism. The proposed system is based on multiple biometric fusion that uses two biometric traits, fingerprint and iris. Both fingerprint and iris are highly accurate biometric traits. A fingerprint recognition system can create problems if the sensor is unable to capture a good fingerprint image due to uneven light effects or if scars or cuts are present on the fingertip. Sometimes, under certain conditions, a user is not able to give his/her proper iris image due to eye disease, problematic iris scanners, ambient light, or absence of that trait. These directly affect the recognition of a biometric system. In the literature, there is little consideration on combining these two biometric traits. The combination of these two traits can attain the best recognition performance for high-safety applications [55]. In this dissertation, these two traits are efficiently combined at decision level fusion and their hardware architecture is presented. Fig. 2.7 shows the block diagram of the proposed architecture. To the best of the author's knowledge, no other FPGA-based design exits that used these two traits. Other performance figures, such as overall computational speed, resource utilisation, recognition rate and memory usage, described in the following

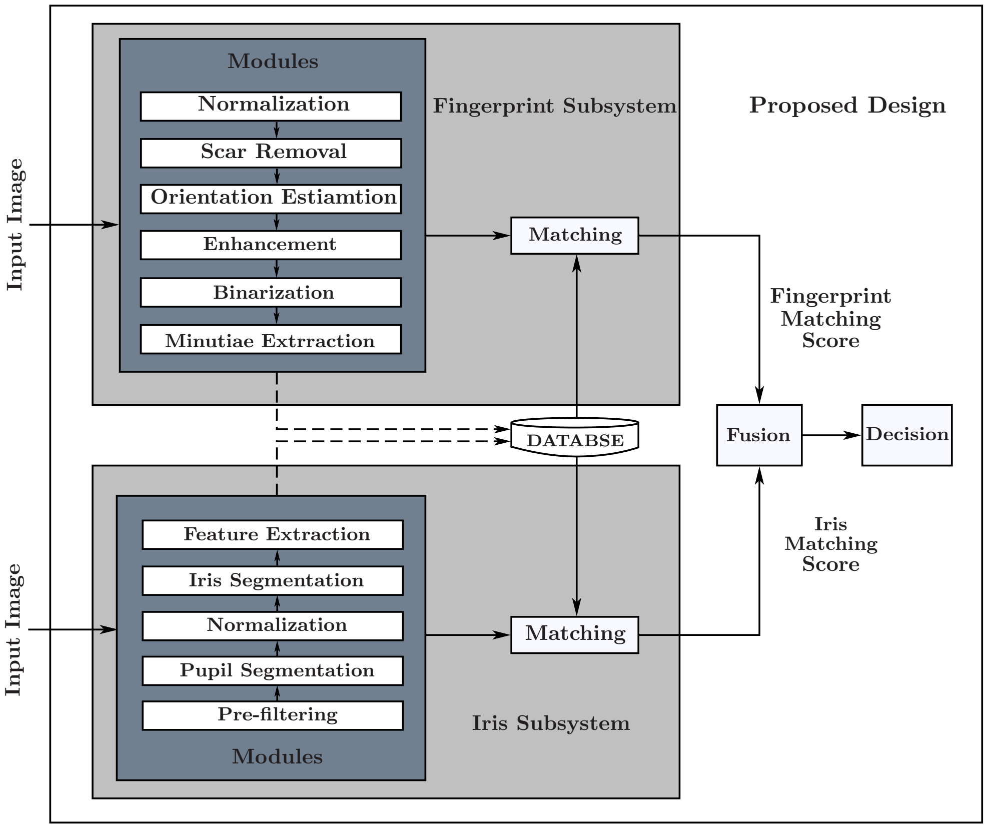


Figure 2.7: *Block diagram of proposed multimodal biometric system*

chapters, were also found to be very promising.

Chapter 3

A spatial domain scar removal strategy for fingerprint image enhancement[1]

This chapter is an adapted version of a journal article. The section headings from the journal article have been retained. Figures, equations, tables, and references have been re-numbered and are in line with the thesis format. This chapter presents a software-based technique for a scar-removal strategy for fingerprint image enhancement. With respect to our proposed design, this chapter falls in the fingerprint subsystem. It covers three modules of the fingerprint subsystem, as highlighted in Fig. 3.1.

[1] Published as: Mohammad A. U. Khan, Tariq M. Khan, D. G. Bailey and Yinan Kong,"A spatial domain scar removal strategy for fingerprint image enhancement," *Pattern Recognition, Elsevier* , vol. 60, pp. 258-274, 2016.

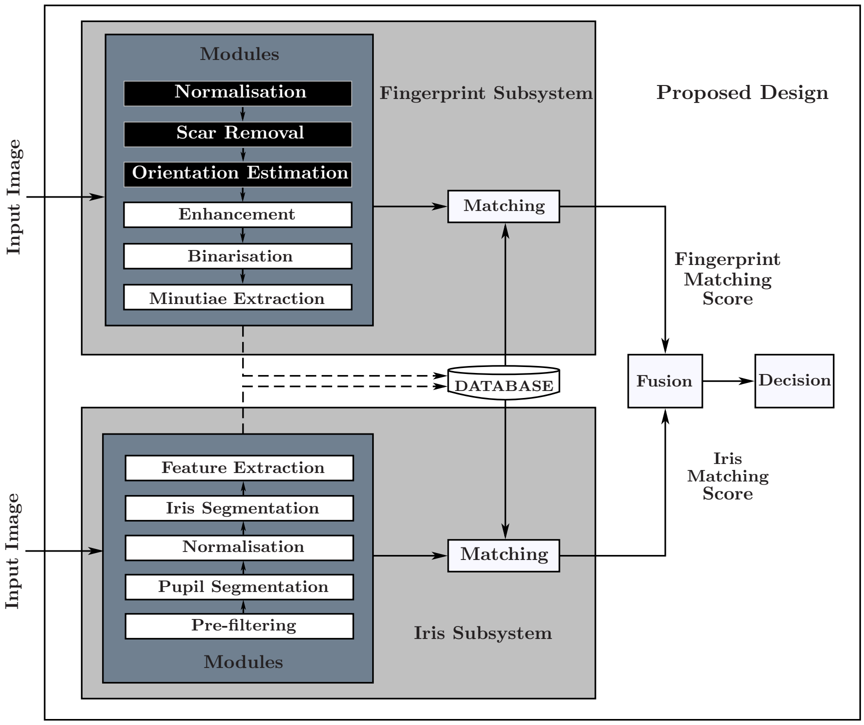


Figure 3.1: *Modules of the proposed design under study*

3.1 Abstract

Fingerprints are the oldest and most widely used form of biometric identification. Many researchers have addressed the fingerprint classification problem and significant progress has been made in designing automatic fingerprint identification systems (AFIS) over the past two decades. However, some design factors such as lack of reliable minutia extraction algorithms, difficulty in quantitatively defining a reliable match between fingerprint images, poor image acquisition, low contrast images create bottlenecks in achieving the desired performance. Noticeable among them is the fact that digitally acquired fingerprint images are rarely of perfect quality to be used directly with AFIS; one important step is fingerprint enhancement. Conventional fingerprint enhancement methods, such as Gabor and anisotropic filters, do fill the holes *and gaps in ridge lines but lack the necessary capability to tackle scar lines. For scar lines, an explicit filling process is proposed that is a mix of Fourier and spatial domain strategies. The proposed method is to make use of the Fourier domain directional field to trace an appropriate candidate for the scar pixels to be replaced with. The necessary components of the process are locating scars, estimating directional field, and the filling strategy. This process can act as front-end to the subsequent Gabor and anisotropic diffusion filtering. The simulation results for synthetic, as well as real fingerprints, show improved performance regarding better extraction of genuine minutia points.*

3.2 Introduction

Fingerprint biometric is getting increasingly employed in commercial, civilian, military, and financial institutions. A fingerprint is formed as an impression of the pattern of ridges on a finger. A ridge is defined as a single curved segment, and a valley is a region that lies in between ridges, as depicted in Fig. 3.2. Ridge have lower reflectance than a valley, and thus, appear darker. However, from the detection point of view, they both can be detected with a single ridge detector with opposing polarity. An ideal ridge/valley gray-level cross-sectional profile can be approximated by a Gaussian function. Ridges and valleys both run in parallel in most of the fingerprint, however, at some locations they either merge or terminate, resulting in important minutia points. The minutia, the local discontinuities in the regular ridge flow pattern, provide the necessary features for identification. Minutia comes in two types: ridge bifurcations and terminations. Ridge bifurcations are the points where a ridge splits into two branches, and terminations are where the ridge ends. Details such as the type, orientation, and location of the minutiae are taken into account when performing minutiae matching for identification [56].

Extraction of ridge-valley patterns from acquired images should be accurate and perfect in the sense that subsequent automated processing can take place reliably. In reality, acquired fingerprint images are rarely of perfect quality. They may be degraded and corrupted with elements of noise due to many factors including variations in skin and impression conditions at the time of acquisition that will result in scars, breaks, too oily or too dry conditions. The two major degradations, apart from small holes in ridge structure, include the presence of scar lines and falsely mixed ridges. Out of these major problems, the problem of scars needs special attention, as it can not be resolved by conventional enhancement. The presence of scars can lead to a significant number of spurious minutiae being created and genuine minutia being ignored. A scar removal strategy should be considered as part of more robust fingerprint image enhancement. In general, a Fingerprint image enhancement aim is to improve the overall performance of the AFIS by minimizing the number of verification or identification errors [57]. The importance of fingerprint image enhancement can be illustrated by the fact that only a few fingerprint recognition algorithms do not use this step. Analysis of the Fingerprint Verification Survey 2004 [58] “FVC2004” reveals that 25 of 29 participating algorithms perform image enhancement before feature extraction and matching.

Fingerprint image enhancement can start with conventional noise-reduction techniques present in image processing literature. General purpose noise-reduction techniques were not found to be as successful as expected. Their failure could well be attributed to the non-stationary nature [59] of a fingerprint surface. A fingerprint has a regular texture with well-defined orientation and frequency that only remains constant in a small neighbourhood. Later on, techniques emerged for fingerprints that include local neighbourhood information during the noise-removal filtering operation. Notable among them is the most widely cited method employed by Hong *et al.* [60], which is based on the convolution of

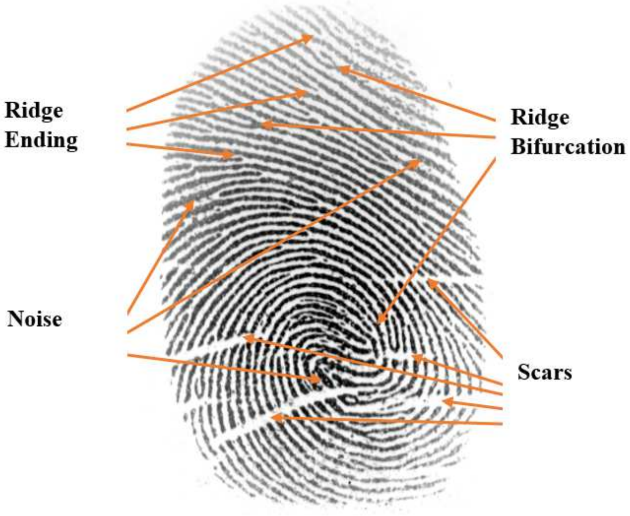


Figure 3.2: *Fingerprint image showing noticeable scars.*

the fingerprint with Gabor filters tuned to the local orientation and ridge frequency. The method comprises of normalisation, ridge orientation estimation, ridge frequency estimation and filtering stages. The normalisation step is to provide a uniform background to the acquired fingerprint that may not be the case due to imperfections in the fingerprint capture process. An orientation image is then calculated in the form of a matrix with directed vectors pointing the ridge orientations at each location. The orientation image associated with a fingerprint displays a slowly varying orientation character in a local neighbourhood; a fact found extremely useful while devising scar filling strategy in later sections. The next step is the estimation of ridge frequency by projecting grey level profile along an axis orthogonal to the local orientation. Based on local orientation and ridge frequency, an even symmetric two-dimensional Gabor filter is computed as given by:

$$g\left(x, y; \theta, f, \sigma_x, \sigma_y\right) = \exp\left\{-\left[\frac{x_\theta^2}{2\sigma_x^2} + \frac{y_\theta^2}{2\sigma_y^2}\right]\right\} \cos\left(2\pi f x_\theta\right) \tag{3.1}$$

where $x_\theta = x\cos\theta + y\sin\theta$, $y_\theta = y\cos\theta - x\sin\theta$ and θ is the orientation of the filter and f is the local ridge frequency. The x_θ and y_θ are the respectively rotated coordinates with fixed σ_x and σ_y as standard deviations, along the major and minor axes. In the spatial domain, a 2D Gabor filter is a product of 2D Gaussian kernel with a sinusoidal function. It has the attributes of frequency tuning and orientation selectivity. Therefore, it can be regarded as an excellent 2D bandpass filter with strong directionality [60]. Due to their plain wave filtering characteristics, they have been found appropriate for texture representation and discrimination [61]. Due to sinusoidal grating characteristics present in fingerprint images, Gabor filtering is successful in restoring distorted fingerprints [57, 62]. The method works on rectangular blocks of pixels and thus, results in fast computations and provided a pleasant look to the image contrast. However, in AFIS, the main thrust is to get better in terms of matching accuracy for detected minutia points. Whereas, the Gabor filter ignores minutia points for improving periodic texture. This fact can be easily observed in Fig. 3.4 where the minutia points are blurred in the final enhanced version. Later on, it was suggested that the ridge/valley pattern may not contain just one sinusoidal frequency but multiple harmonics. Therefore, authors in [63] proposed a modified Gabor filter by replacing the cosine function with an alternate periodic function having two periods T_1 and T_2, different for inter-ridge and inter-valley distances. Though this modification results in a better ridge-valley pattern representation, the matching score for minutia is not significantly improved. Both techniques are based on the sole assumption that oriented patterns in a local neighbourhood form a periodic plane wave with well-defined frequency range and orientation, which may not be true at minutia points and other singularities. Therefore, at singularities where the curvature is large, these models are bound to create spurious ridge structures. Although this can be avoided by using an adaptive strategy where one can force only low-pass filtering at singular points, the exact mechanism of this adaptation is a difficult task, especially in noisy fingerprints.

To avoid ridge frequency dependence, some researchers simplified the Gabor filter. They came up with a new contextual filter that makes use of exponential part of Gabor function and is changing only with local ridge orientation. These filters are called anisotropic filters. Perona and Malik [64] did the pioneering work. They suggested simulating heat diffusion process in a heterogeneous medium for edge enhancement. They selectively applied the low-pass filtering with complete inhibition at the edge points. Improving on this fundamental idea, Almansa [65] and Wickert [66], used low-pass filtering (diffusion techniques) based on a multi-scale analysis. They posed the ridge/valley pattern as a multi-scale structure by first applying an iterative process for local feature estimation with the appropriate scale and then use a low-pass filter. Yang *et al.* [67] proposed another kind of structure adaptive anisotropic filtering technique. Instead of using local gradients as a means of controlling the anisotropy of filters, it uses both a local intensity orientation and anisotropic measure to control the shape of the filter. Although the filters proposed by Yang [63] and Almansa [65] are both structure-adaptive anisotropic filters, Almansa's filter is iterative while that of Yang is not. In these approaches, whenever a ridge edge is encountered the circular filter kernel is deformed into an ellipse with the major axis aligned in parallel with the edge. Therefore, smoothing is performed along but not across the ridge lines [68]. Thus, instead of inhibition, steering and shape adaptation takes place. The success of anisotropic filters in preserving the ridge bifurcations is well-established. However, sometimes if the length of the major axis associated with the ellipse is not controlled, it may get too long and result in connecting the ridge endings with other nearby ridges. This results in converting the true ridge terminations into false ridge bifurcations. Although this does not seem at first to be a problem, as the minutia position and count are undisturbed, a type mismatch error will happen (that is, changing a ridge termination to ridge bifurcation). Moreover, since anisotropic diffusion filter only involves smoothing over an elliptical neighbourhood in the vicinity of a given pixel with total disregard to the sinusoidal wave pattern, their smoothing ability is limited, and also tend to reduce the overall image contrast. It can be concluded from the literature and our experience with Gabor and anisotropic filters that the Gabor filter is more aggressive in restoring the lost data while blurring singularities like minutia points. The anisotropic filter, on the other hand, proceeds more gently having better representation for singularities but encounters greater difficulty in large noisy areas for restoring the required shape for the subsequent minutia extraction module.

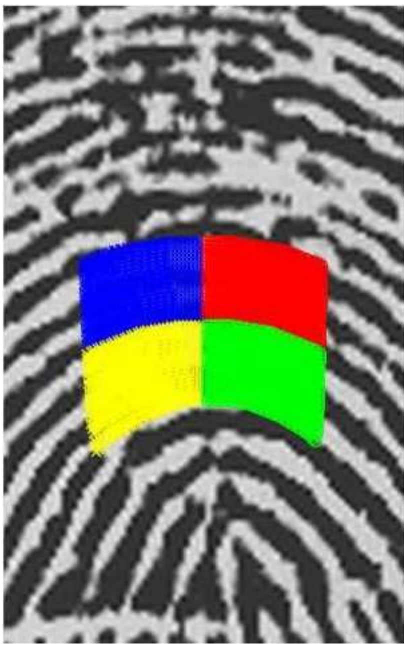
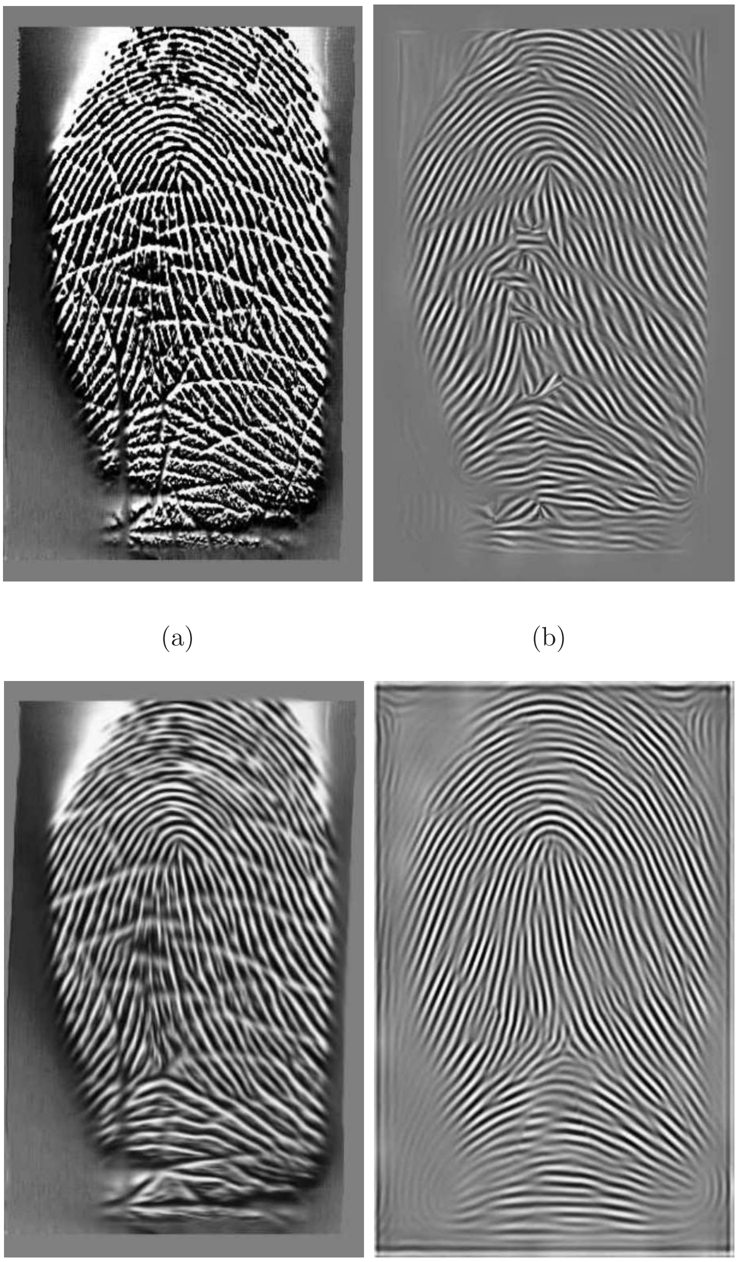


Figure 3.3: *Curved Gabor filter support region for a curved portion of a fingerprint. We see that it clearly capture the curvature of the ridges.*

In an attempt to further increase the smoothing power of Gabor function without affecting feature shape, a Curved Gabor filter is introduced [57]. The ridges and valleys are smoothed, but their shape is kept intact. This happens due to the simple reason that conventional Gabor and Anisotropic filters both relies on piece-wise straight filter region while the Curved Gabor takes advantage of natural curvature inherent in fingerprint images. This results in a curved filter kernel that goes along the bent ridges and valleys as illustrated in Fig. 3.3. Since it is made to follow the ridge curvature deliberately, the curved filter provides not only greater smoothness but also tone down scars. The scar suppression happens as the filter remains orthogonal to scar lines at all times. This puts Curve Gabor filter as one of the effective enhancement algorithms with an ability to implicitly dealing with scars. However, these desirable attribute of Curved Gabor filtering comes at a price, and that is its associated high computational burden. Curved Gabor filtering as compared to conventional straight Gabor filtering employs a considerable battery of pre-processing machinery. Finding the curved region of support for each pixel involves three primary stages: First, computing interpolation necessary to raise the resolution level of the fingerprint, Second: estimating a highly accurate underlying local orientations, that is orientation field, and Third: computing accurate local ridge frequencies. Out of three main requirements as mentioned, the accuracy estimating local orientations is paramount. The curved region can quickly go astray from its true trajectory if not provided with true orientation directions. To achieve this, the Directional-filter bank based orientation estimation [69] is employed here for simulating Curved Gabor filter results, which is then subsequently smoothed with iterated orientation diffusion [70]. This puts an extra computational burden on the scheme. Once the orientation field is sufficiently smoothed, it is used then to find the curved support region for Gabor filter to overlook most of the scar lines.

The Fig. 3.4 displays a comparison result for fingerprint image enhancement for a scarred image was taken from FVC2004 database [58]. The observation is that both Gabor and anisotropic filter do well regarding connecting broken ridges, but they break down once it comes to removing long linear scars or cuts in fingerprint images. Though Curved Gabor filter is doing a better job while dealing with scars but comes with a heavy computational price tag. Apart from these three classical techniques, there are other enhancement methodologies specifically suggested for fingerprint image enhancement such as Chikkerur [71], ROM [72], TwoStep [73], Oriented Deviation based [74], Improved Gabor-based [75] and Genetic approach [76], present in the literature. They do have their



(a)

(b)

(c)

(d)

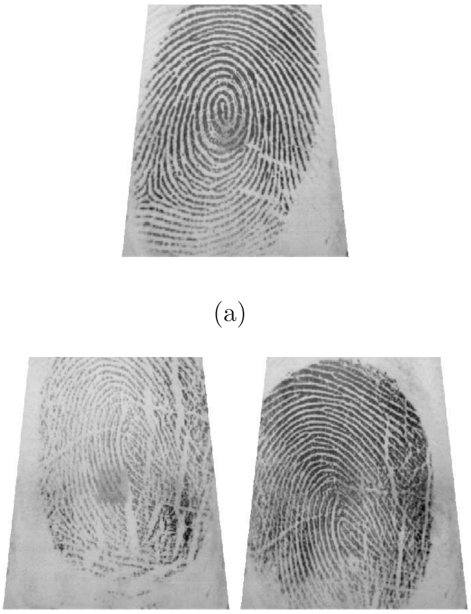
Figure 3.4: *A real life sample image containing scar tested on two well known fingerprint image enhancement methods. a) Scarred Image. b) Enhanced by Gabor filter. c) Enhanced by Anisotropic filter. d) Enhanced by Curved Gabor filter*

unique merits, but they did not take up the scar issue. To address linear compensable scars that happen over time, a computationally light but explicit strategy is investigated in this work that first marks scars pixels for compensation and then fills them from their surrounding neighbours. Duly compensated these scarred images are then ready to be treated with fast straight Gabor and Anisotropic filtering techniques to do remaining enhancement job.

Scars appear on fingerprints in many ways. Fig. 3.5 shows three noisy images from the FVC2004 DB2_B database containing scars of different modalities. Friction ridges are often broken by bending creases, as shown in Fig. 3.6(a) and (b), and sometimes by scars due to injury, as shown in Fig. 3.6(c) and (d). The focus of the chapter is to deal with scar lines while creases are unpredictable variations, hard to compensate. The scars can be categorised into two main types: the compensable scars and the non-compensable scars. Both look very similar except for two major differences that can be used to distinguish them:

1. Ridges broken by non-compensable scars are often misaligned and thus degrade the ridge pattern in a random way. Here the ridge/valley pattern is disrupted, and the immediate boundary pixels do not provide the consistent direction. Therefore, there is little hope to compensate these. However, ridges broken by compensable scars are still aligned very well across the scar region.

2. Non-compensable scars are not as stable as compensable scars. They can come and go unpredictably in subsequent scans taken over a time line. Non-compensable scars tend to become narrow or even disappear in inked impressions. They can also arise in live-scan images of wet fingers or impressions made by excessive pressure during fingerprint capture. There is not much that can be done with these scars and hope that they will disappear with enhancement, but algorithms can be devised to remove or reduce compensable scars.

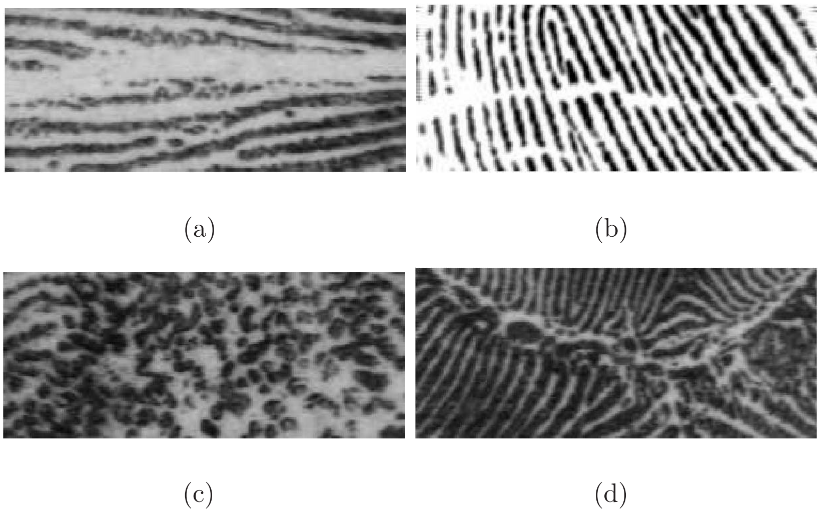


(a)

(b) (c)

Figure 3.5: *Three sample images containing scars taken from the FVC2004 DB2_B database [58] . a) sample image 105_5 b) sample image 107_1 c) sample image 107_2*

To find reasons for fingerprint scar generation, the origin of the pattern on fingertip themselves need to be investigated first. Though almost all scientists agree that finger patterns began developing quite early on after birth and are complete by the end of the fourth month, there is a lot of speculation about the precise process of their creation. One widely hold theory states that a middle skin layer, called the basal layer, is pressed between the inside layer (the dermis) and the outer skin layer (the epidermis). This causes strains against its neighbours, the pressure, forces the skin to get folded resulting in valleys and ridges structure [79] appearing on the fingertip. Since the pattern is carved deep at the interface between the dermis and epidermis, that it becomes nearly permanent throughout the life and cannot be destroyed by superficial skin injuries. Having said that, there are many documented cases of intentional fingerprint mutilation. But those involve pretty severe damage to the skin, more specifically between the generating



(a) (b)

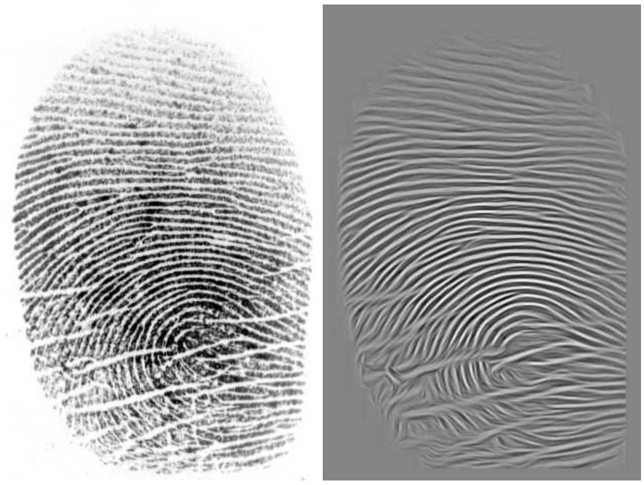
(c) (d)

Figure 3.6: *Flexion creases and scars in fingerprints. (a) A major scar between finger ridge lines, completely disrupting the flow, (b) a piece-wise linear scar, that can be compensated. It has kept intact the ridge flow pattern across the scar region. (c) large area scar with random perturbations, beyond compensation, and (d) a scar with random ridge/valley pixel distribution, beyond compensation. The image in (b) is cropped from fingerprint 10_3 in FVC2002 DB1 [77] and the other three images are cropped from fingerprints F0201, F1022 and F0693 in NIST SD4 [78], respectively.*

layer where the template of the fingerprint survives, and the upper layer, the epidermis. Accidental cuts with a sharp object also produce scars. But, the most amazing thing is that damaged skin can reproduce cells to form the fingerprints exactly as they were before they were damaged. This remedial action happens to the cuts that are on the surface, but with deep cuts that damaged the inner layer of skin (known as the dermis), then one gets a permanent scar. In a nutshell, any cut or burn that goes deeper than the outer layer of the skin can affect the fingerprint pattern in a permanent way. The permanent new scar can now become a unique aspect of his/her identity. The work presented here, deals specifically with cuts or scars that heal themselves over a period, and could pose hindrance in identity verification till that time it restores to its original pattern.

There is some literature on scar removal strategies. The researchers find scar lines hinder the matching performance of AFIS and have made some attempts to compensate major scar lines. Oliveira *et al.* [80] proposed spatial domain based inpainting technique. They used the diffusion matrix to reconstruct the broken ridges. Gottschlich [81] presented a line sensor based method to estimate the flow of ridges and valleys for removing scars. To reconstruct the oriented ridge patterns in an image, Xudong [82], Sulong [83] proposed a directional filter that strengthens the ridge pattern of local dominant orientation, while suppressing the scar lines that are oriented against the dominant direction. This method prefers a band-pass directional filter to remove the uneven background and suppress high-frequency noise. This scheme is devised based on the assumption that fingerprints exhibit everywhere a well-defined local ridge orientation. Although the scheme was successful to diminish the scar lines at some places, it was not able to eliminate them entirely from the print, as depicted in Fig. 3.7. As an alternative, a scar removal strategy is proposed that makes the best use of both worlds frequency domain and spatial domain. It mixes frequency domain based directional field and spatial domain image inpainting ideas. From experiments conducted, it is found that the proposed scheme can reconstruct the ridge flow in a region crossed by a scar.

Digital inpainting techniques are commonly used to reconstruct small damaged portions of an image. They serve a wide range of applications, such as reconstructing scans of deteriorated images by removing scratches or stains, removing text and logos from still images or videos, or creating artistic effects. Most inpainting methods work as follows. First, the image regions that need inpainting are selected. Next, the known image information is used to fill in the missing areas by propagating inward from the region boundaries. To produce a good reconstruction, an inpainting technique should attempt to continue lines of equal grey value as smoothly as possible inside the reconstructed region. In [84], the image smoothness information is propagated along the isophote directions, estimated by the image gradient rotated by 90 degrees. The image Laplacian is used to calculate the gradient. Later, [85] proposed a new technique for inpainting by propagating an image smoothness estimate along the image gradient. The missing regions are treated as level sets, and a fast marching method is used to propagate the image information.



(a) (b)

Figure 3.7: *Results of adaptive directional FFT filter on a noisy image containing scars. (a) input image; (b) enhanced by [82].*

A novel inpainting method is proposed here, to be used with both Gabor-based fingerprint enhancement and diffusion-based fingerprint enhancement. First, identify the scar region along its boundary. Next, an orientation field is computed using a directional filter bank in the frequency domain. Employing orientation field information, the movement can be made along the correct slope from a scar boundary pixel into a non-scar pixel territory and pick an appropriate replacement candidate there. The boundary scar pixel value is replaced with that of the candidate pixel value. This way all the boundary scar pixels filled are found. Now, a new boundary is identified, and filling is continued. This new boundary keeps moving inward until the scar region is filled.

The rest of the chaper is organised as follows. In Section 3.3, a detailed description of the proposed scar removal strategy is unveiled with a special focus on its components such as background variation correction, binary mask creation, the computation of the orientation image, filling scar region algorithm, and validation with synthetic images. Section 3.4 presents experimental results and their quantification for real images followed by concluding remarks in Section 3.5.

3.3 Scar Removal Strategy

The fingerprints present in public databases and the ones obtained from our campus students, confirm the presence of major scars with a varying degree of dominance over the regular ridge/valley pattern. In some cases, a lot more activity is observed associated with these scar lines to an extent where they become a feature set. The research literature and the industry connected with access technology agree on the necessity to remove these scar lines. Medically speaking, a scar is defined as fibrous connective tissue that forms at the location of an injury such as a cut. With the passage of time, scar tissue may replace injured skin, but usually paler in coloration due to reduced blood supply. Once scanned, scars lines will appear in fingerprint images with a clear interruption of the regular ridge/valley pattern. Since we are concerned with minutia matching score, the scar lines may produce false terminations. Also, if the scar happens right at the bifurcation point or termination point, then it is entirely up to the scar removal strategy to recover that minutia. Our assessment is that it is hard to recover these damaged minutia points. However, an attempt can be made to remove the false minutia due to scar lines.

Closely observing the scar lines present in most of the fingerprint images, a direction mismatch can be observed between scar lines and the ridge flow directions. That is; scar lines frequently occur in directions approximately orthogonal to those of regular ridge paths. Therefore, a Fourier-domain directional filter should be able to detect those fault lines. A Fourier-domain technique for compensating scars in fingerprint images is devised on the same theme of [86]. The authors proposed to divide the image into blocks. For each block, the directions are accumulated for the pixels. A specific direction having the maximum number of occurrences within the block is selected as the dominant direction. The block is then Fourier transformed and a wedge-shaped directional filter is applied to filter the contents of the block with the dominant direction. After filtering, the block is transformed back to the spatial domain with the sense that if there was a scar line present, then it will be largely diminished. The proposed Fourier-domain technique is heavily dependent on the pretext that the fingerprint ridge pattern direction changes slowly and will remain constant in a small neighbourhood. However, it was observed later in [87] from the examples that while such a scheme does reduce the contrast of scar lines, it cannot eliminate them entirely. This kind of image enhancement will put the stress on the binarization process used in later stages. Further, there is a chance that interpolation will also create some bifurcation points that were originally terminations. The use of the dominant direction linked with each pixel that is the orientation field provided the basis for our proposed spatial-domain scar removal strategy.

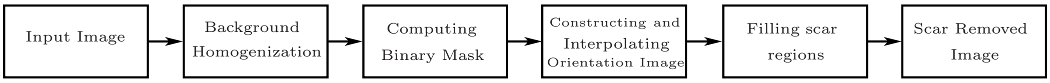

Input Image
Background Homogenization
Computing Binary Mask
Constructing and Interpolating Orientation Image
Filling scar regions
Scar Removed Image

Figure 3.8: *Block diagram of the proposed scar removal method*

Spatial-domain scar removal fills the scar region with pixels from nearby ridge/valley pixels, taking help from the already computed orientation field. The orientation field, computed by averaging over a block of pixels, is large enough to bypass the scar region, and its direction remains undisturbed by the scar pixels and follows the general ridge orientations shown in Fig. 3.9. Knowing the orientation of the scar pixels, the necessary guidance is obtained to move in a specific direction for finding the replacement. Now what is left is to pick the appropriate ridge or valley pixel and graft them back at the scar locations. This method can find some resemblance to medical procedures where the physical scar tissue is replaced with a similar albeit healthy tissue somewhere else in the body.

In a formal sense, the spatial scar is removed by image inpainting. For execution, the area of missing pixels is first defined in the original fingerprint image, named as a mask image. The mask image is a binary image, where only the missing scar pixels are labelled as one, the rest have the value zero. To create a mask image for our scarred fingerprints, a geometric filter is required that provides a strong response to our scar locations and low response for the rest of the image. Then a threshold will convert this filtered image into a binary mask image. The block diagram of the proposed method is shown in Fig. 3.8.

3.3.1 Background Homogenization

The acquired fingerprint images often show important background variations, poor contrast in some areas and gaps in ridge/valley regions. To reduce the first two imperfections,

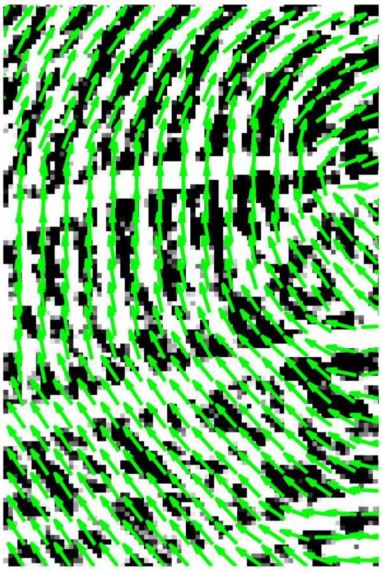


Figure 3.9: *Orientation field extracted by directional filter bank.*

and generate images more suitable for enhancement and minutia extraction, preprocessing comprising non-uniform background correction is carried out. The background variations occur due to the very process of scanning a finger. The finger's middle surface is thicker as compared to the surrounding region. This results in more pressure in the middle giving a darker impression, while the outer surface has less pressure giving a lighter impression. The fingerprint scanner registers this uneven pressure. Consequently, background variation will add unnecessary bias for different regions of the same image to disturb the ridge/valley contrast, especially in the central region. Since the ridge/valley pattern is identified and classified by its gray-level profile, this effect may worsen the performance of enhancement and ultimately results in missing genuine minutia.

Firstly, a 3×3 median filter is applied to smooth occasional salt-and-pepper noise. For estimating background image f_B, a large size low-pass filter is required to capture background variations, which is assumed to be larger in width to the regular ridge/valley sinusoidal pattern. To start developing a discrete low-pass Gaussian kernel, we choose $\sigma = 8$ pixels. The choice reflects the average size of the ridges in our database images.

Then making 3σ rule to have 99% confidence interval that two ridges and one valley fit well within the kernel size. That results in the width of the kernel as 24 pixels. Measuring across the whole background variation from one edge to the other, we end up with 6σ. However, as our kernel is in discrete domain, we need to accommodate the centre of the Gaussian as well. As such, the kernel size is thus: floor(6σ) + 1 or simply 49×49. When this filter is applied to the pixels outside the region of the fingerprint near the border, results are strongly biassed by the external light region. To overcome this problem, out of the region, grey levels are replaced by the average grey levels in the region of interest. The filtered image f_B is also called the background estimation image. Many algorithms are reported such as [88], to segment the image into foreground and background area. We adopted the local variance method that is applied block-by-block to identify the foreground blacks as opposed to background blocks. Accumulating all the foreground blocks, while eliminating any holes or gaps provide a contagious foreground region, referred to as a region of interest here. Then, the difference D between the image f and the background estimation image f_B is calculated for every pixel.

$$D\left(x,y\right) = f\left(x,y\right) - f_B\left(x,y\right) \tag{3.2}$$

To this respect, the literature reports two alternate ways to get down to the corrected image. One method is based on subtracting the background image from the original image [89], [90], [91] and the second deals with the division of the original image by the background image [92], [93]. Since division by darker regions will result in numerical unsuitability, the subtraction strategy is opted, as described in Eq. 3.2. Besides the background variations in images, there is also signicant contrast variations between database images due to different pressure conditions in the acquisition process. The difference results in negative values. By representing the difference in offset binary (by adding 128) the pixel values are brought back into the range 0 to 255. i.e.

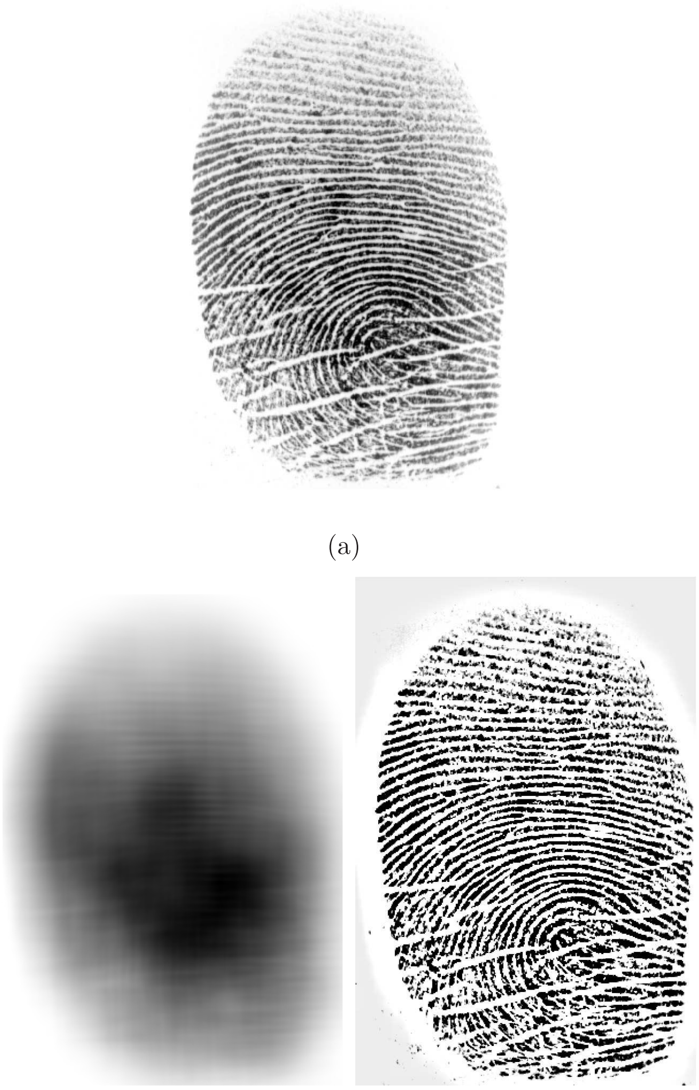
$$\text{f\_\{cc\}}\,(D) = \begin{cases} 0, & if\, D < -128 \\ 255, & if\, D > 127 \\ D + 129, & otherwise \end{cases} \tag{3.3}$$

A contrast-corrected image is obtained by linearly stretching the new image range locally to fit the range of possible grey levels. The local contrast is estimated by Gaussian filling abs(D) and using this to scale or scratch the background-corrected image. Fig. 3.10(c) shows the resulting image, f_{cc}, corresponding to the stretched and background homogenised image. The proposed contrast-correction algorithm is observed to reduce background intensity variations and enhance contrast in the middle region of the original fingerprint image. The method was validated for all the images that are processed in the database.

To validate the results of the background homogenization, binarization using histogram before and after homogenization is analysed. The histogram of an image represents the relative occurrences of the gray-level present in an image [94, 95]. The non-uniform background and contrast degrade the histogram of an image in a way that it will be hard to binarize it by a single global threshold. In other words, no single valley will appear in the resulting histogram. For binarization, Otsu's method [96] is employed to choose the threshold to minimise the intra-class variance of the background and foreground. It is used to find the binary threshold for the original fingerprint and that of the contrast normalised image. The results are displayed in Fig. 3.11.

3.3.2 Computing Binary Mask Image

Scars in fingerprints happen as thick linear white segments stretching from one corner to the almost middle part of the print, so it seems plausible to model them as two-dimensional



(a)

(b) (c)

Figure 3.10: *Background homogenization process: Fig. (a) shows an acquired digital fingerprint. Fig. (b) depicts the estimated non-uniform background. Fig. (c) is an output result after passing the image through homogenization operation and then linearly stretched. It can be observed that contrast has been corrected with clear ridge/valley structure.*

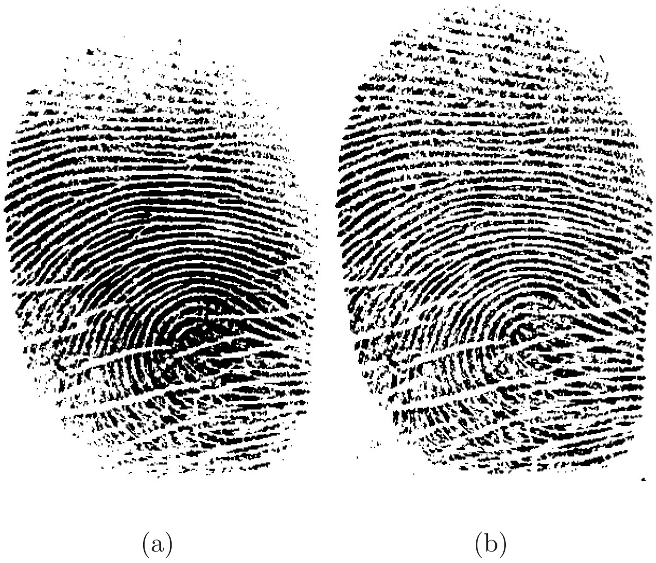


Figure 3.11: *Validation test for background removal output: Fig. (a) shows the binarization of an acquired digital fingerprint using optimal Otsu method. Fig. (b) depicts the binarization after homomorphic filtering, also using the optimal Otsu method. It can be observed that binarization results for filtering output show all the regions with ridge/valley structure intact.*

valley structures. However, we already have many valleys as a regular fingerprint pattern. The main question is how to separate scars from regular valleys. Fortunately, it was observed from a large database of acquired images that the scars differ from the regular valley patterns in the direction as well as width. Ridges and valleys are similar in shape but have complementary brightness. That is, ridges are darker while valleys are lighter in contrast. Therefore, a ridge detection filter with a proper width and direction but complementary in brightness should be able to detect scars while avoiding valley patterns. Several ridge detection filters are available in the literature [97–99]. The specific approach adopted here is to look for a filter that matches the geometric shape of the scar under investigation. The scar lines have a deep valley structure that matches well with the second-order Gaussian filter construction as shown in Fig. 3.12. Although a fourth-order Gaussian filter proposed in [97] was found to have better directionality property, these higher-order Gaussian filters are more suitable for junctions or overlapping texture. Since crossing scar lines are not considered, the proposed algorithm is restricted to second-order Gaussian derivative filters.

First, the empirical evidence about the width of scar lines is obtained from database

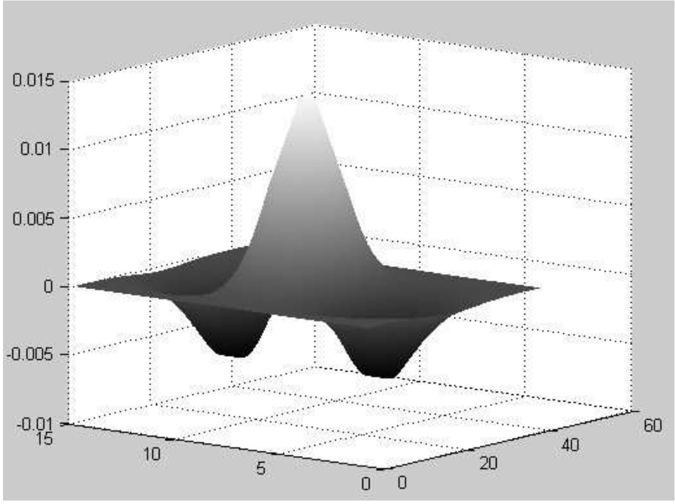


Figure 3.12: *A two-dimensional 2nd Order Gaussian Derivative Filter.*

images. Scars were found to vary in width from eight to ten pixels with an average width of nine pixels. The regular valley lines are almost always four pixels wide with plus or minus one pixel. This makes it easy for the second-order Gaussian derivative filter to locate scar lines. Once its parameter σ is defined in the range from four to five that results in overall width in the range eight to nine pixels, spanning 2σ distance. This parameter setting will provide us with a second order Gaussian derivative filter matching well with that of scar lines while avoiding the regular valley lines completely. For the chosen filter not to miss any scar line and to be flexible, a multi-scale elongated second-order Gaussian derivative filter is obtained. The elongated second-order Gaussian derivative filter for image features contains three parameters, that is, its length σ_u, and the width σ_v and orientation. For keeping the elongation, the length σ_u is tied to be multiple of width σ_v. The multiple comes from a set having values $\{2, 2.5, 3, 3.5\}$. The width parameter σ_v is chosen from a set $\{4, 5\}$. The maximum picked among all possible set of values for length, width and orientations.

For implementing a second-order Gaussian derivative filter, a generalized two-dimensional Gaussian function is used, given by:

$$g\left(u,v\right)=\frac{1}{2\pi\sigma_u\sigma_v}\exp^{\left(-\left(\frac{u^2}{2\sigma_u^2}+\frac{v^2}{2\sigma_v^2}\right)\right)}. \tag{3.4}$$

It is observed that this generalized Gaussian function has two independent parameters σ_u and σ_v. Now taking its second-derivative with respect to u only, the following expression is obtained:

$$g_{uu}\left(u,v\right)=\frac{1}{2\pi\sigma_u^5\sigma_v}\left(u^2-\sigma_u^2\right)\exp^{-\left(\frac{u^2}{2\sigma_u^2}+\frac{v^2}{2\sigma_v^2}\right)}. \tag{3.5}$$

After sampling the above continuous-function, the discrete kernel is rotated in a specific orientation. The rotation matrix will provide the transformation from xy co-ordinates to uv co-ordinates as follows:

$$u = x\cos\theta - y\sin\theta \tag{3.6}$$

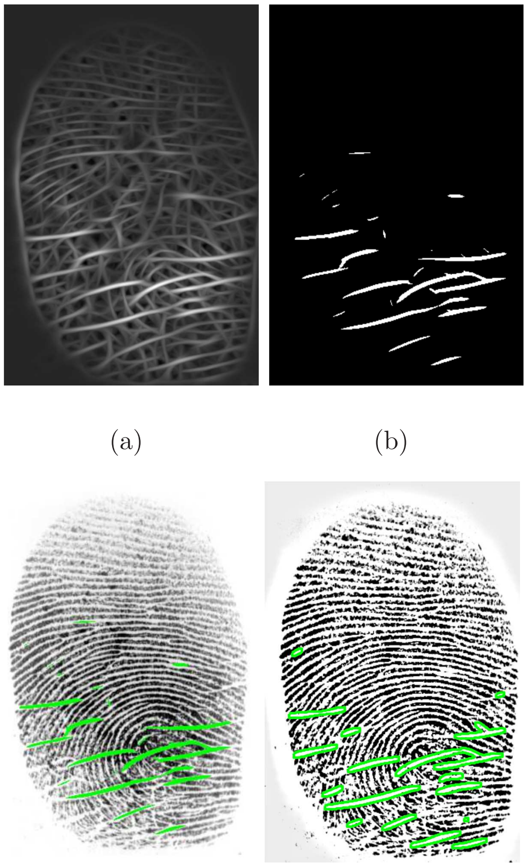
$$v = x\sin\theta + y\cos\theta. \tag{3.7}$$

Since the amplitude of the filter will reduce as the σ_u and σ_v parameters increase, its response needs to be normalised to have a fair comparison across all scales. The output response of the oriented kernel is normalized by multiplying with a factor $\sigma_u^{1.5}\sigma_v^{0.5}$. After normalisation, the maximum for each pixel is determined, probing all different combinations of length, width and orientations. Let us say, $f(x, y)$ is our fingerprint image, and $g_{xx}(x, y, \theta, \sigma_u, \sigma_v)$ is the elongated second-order Gaussian derivative filter. Applying filter the on the image, a new image F is obtained:

$$F(x, y, \theta, \sigma_x, \sigma_y) = g_{xx}(x, y, \theta, \sigma_u, \sigma_v) * f(x, y). \tag{3.8}$$

The detection process is illustrated in Fig. 3.13. Since the normalised filter is used, the absolute value of F peaks for those particular σ_u and σ_v that match the scar line profile. The response of the Gaussian derivative filter to possible scar regions is displayed in Fig. 3.14(a).

It is observed that while creating a mask image for the scar region, sometimes the Gaussian derivative filter detects the valley part of the fingerprint as a possible scar region. This happens because valleys in fingerprints vary in width, and will pose a possible false alarm for our scar extraction process. One possible way to circumvent this is to use the directional ridge field, computed at a pixel that provides local orientation. Now the probing filter can be steered to avoid the direction of the valley using directional ridge field. The local ridge/valley directions are already available through the use of directional filter banks. With this reaction, the Gaussian derivative filter response is shown in Fig. 3.14(b). It can be clearly seen that the filter now avoids valley regions as possible scar regions.

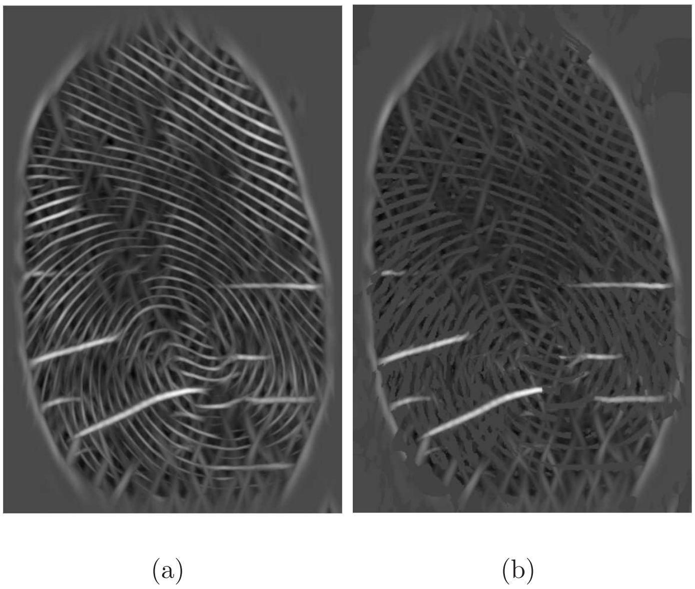


(a) (b)

(c) (d)

Figure 3.13: *Binary mask preparation: Fig. (a) shows the filtering output for a fingerprint with a second-order Gaussian derivative filter. The Fig. (b) depicts the thresholding result to convert the filtering output into a binary mask. Fig. (c) shows a validation step for overlaying the binary mask values for correctly locating the scar locations. Fig. (d) overlays the boundary of the scar regions on top of the fingerprint image, showing that it did capture effectively most of the scars.*

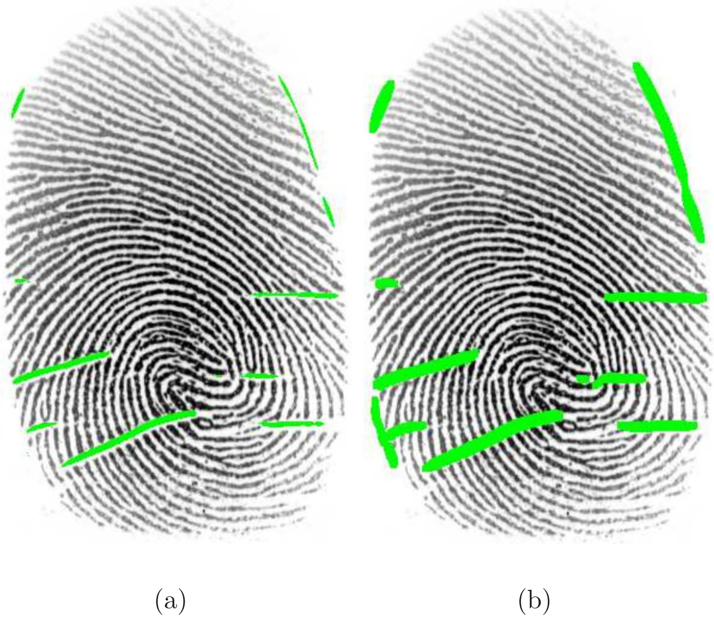
Although a possible precaution is adapted to avoid regular valleys as false scars, our detection is still far from providing a binary mask for the scar regions. This can pose a potential problem for our subsequent scar filling procedure. Therefore, region growing morphological operation is used to make up this deficiency [94].



(a) (b)

Figure 3.14: *The effect of avoiding valley regions while detecting scar regions: a) Shows that the scar detection filter do pick some valley regions in addition to the required scar regions. b) Depicts the scar detection filter output where steering is limited to avoid possible valley directions.*

A region growing process is an elegant segmentation strategy that starts with initial seed points and then starts adding pixels that are 8-connected neighbours with a close intensity to that of the seed points. The closeness of intensity is measured by using a reduced threshold. Thus, the region growing algorithm needs two thresholds, one to create possible seed points, and the second to enlarge or grow the seeded area. For implementing region growing, the initial seed points are associated with the already captured scar region of Fig. 3.15(a) from our second-order Gaussian derivative filtered fingerprint. For this search, the original threshold is reduced by 10 percent and provide the second threshold. The final region growing result is shown in Fig. 3.15(b).



(a) (b)

Figure 3.15: *a) Gaussian Derivative Response Thresholded, b) Region Growing Result*

3.3.3 Constructing and interpolating orientation image

The orientation image of a fingerprint defines the local orientation of the ridges at each pixel. The orientation estimation is a fundamental step in the proposed scar-removal strategy as the subsequent scar pixel replacement stage relies on the local orientation to effectively select an appropriate neighbour. This section investigates two popular orientation image estimation techniques. One operates in the spatial domain and the second in the frequency domain.

The spatial domain technique is based on least mean square estimation suggested by [100], and later on, employed by Hong *et al.* [60] to compute the orientation image. However, instead of estimating the orientation block-wise, their method is extended into a pixel-wise scheme, which produces a finer and much more accurate estimation of the orientation field. The steps for calculating the local orientation at pixel (x, y) are as follows:

1. For each pixel in the block, compute the gradients $G_x(x, y)$ and $G_y(x, y)$, which are

the gradient magnitudes in the x and y directions, respectively. These gradients are calculated using conventional Sobel or Prewitt masks.

2. The local orientation at pixel (x, y) can then be estimated by averaging the orientations within a $W \times W$ window using the following equations:

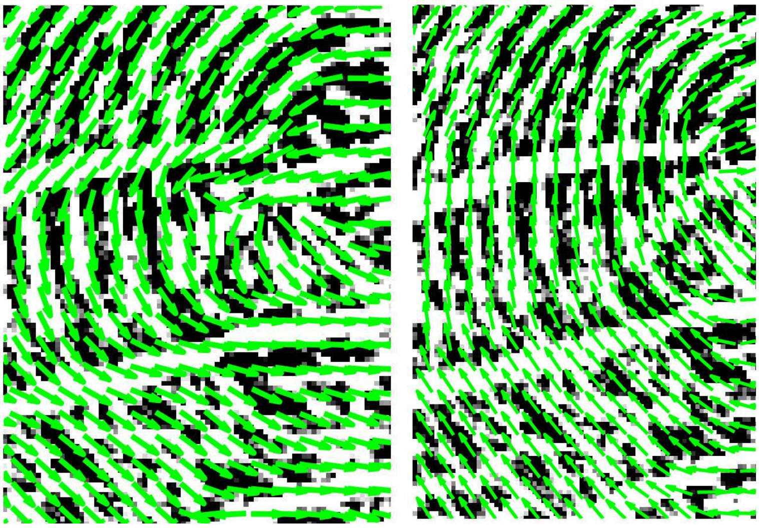
$$G_1 = \sum_{u=x-\frac{W}{2}}^{x+\frac{W}{2}} \sum_{u=y-\frac{W}{2}}^{y+\frac{W}{2}} 2G_x(u,v)G_y(u,v), \tag{3.9}$$

$$G_2 = \sum_{u=x-\frac{W}{2}}^{x+\frac{W}{2}} \sum_{u=y-\frac{W}{2}}^{y+\frac{W}{2}} (G_x^2(u,v) - G_y^2(u,v)), \tag{3.10}$$

$$\text{then } \theta(x,y) = \frac{1}{2} \arctan \frac{G_2}{G_1} \pm \frac{pi}{2}, \tag{3.11}$$

where the choice $\pm\frac{pi}{2}$ is added to give consistency of the calculated orientation image such that all orientations are chosen to be either clockwise or anti-clockwise directions.

The second strategy makes use of the Fourier domain [70]. It starts with decomposing the given image into multiple directional images using a directional filter bank structure. For each directional image, a multiscale tower is constructed to look for various linear segments. A directional energy parameter is computed for each directional image across all pre-defined scales. The final directional image is computed by comparing the directional energy strength measure across all possible combinations of scale and direction for each pixel. This characterises an optimal orientation estimation with low noise power and reasonable feature localization, as depicted in Fig. 3.16. The Fourier domain method provides better orientations that align well with ridge directions. Especially in the scar portion, the Fourier method provides orientations that cross over the region while maintaining their alignments with the ridge pattern as compared to jagged directions in the scar regions provided by the spatial domain.



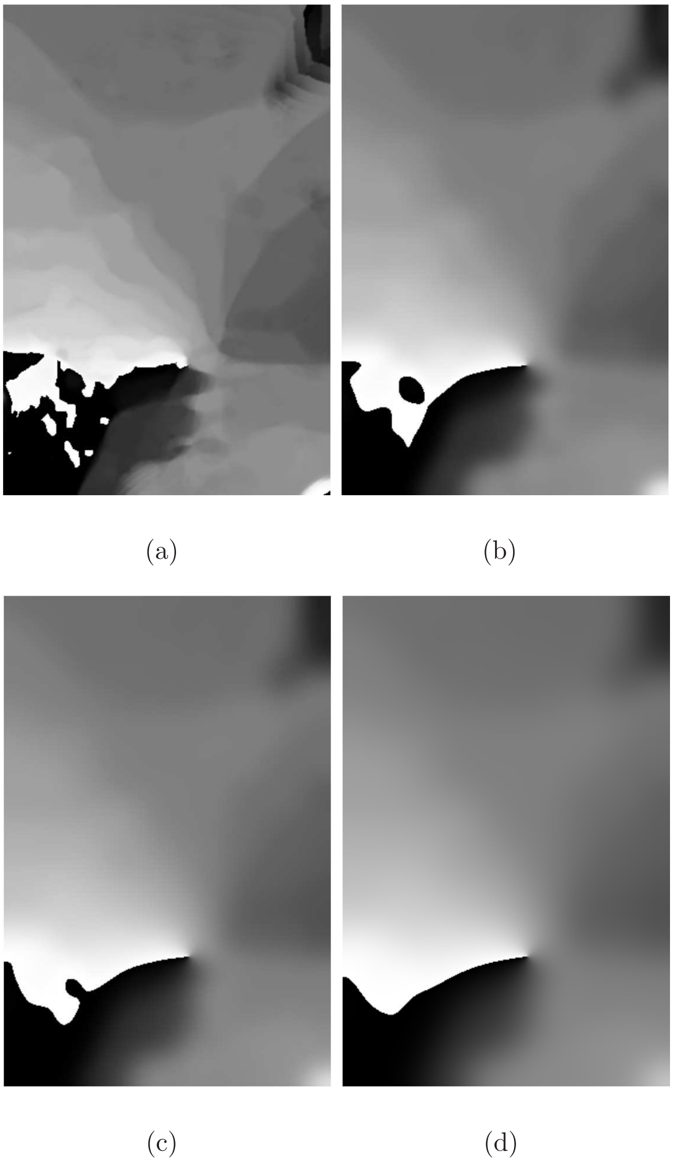
(a) (b)

Figure 3.16: *Comparison of spatial domain vs Fourier domain orientation estimation: Fig. (a) spatial domain method, Fig. (b) Fourier domain method.*

The orientations obtained from the previous step are somewhat noisy. Therefore, the orientation map needs to be further smoothened. For this purpose, the orientation-diffusion method is employed as suggested in [70]. The method of orientation-diffusion is preferred over that of local averaging using a Gaussian filter because it fits better here due to the cyclic and repetitive nature of our data. The results from conducting orientation diffusions on a fingerprint directional image are shown in Fig. 3.17.

The obtained orientation image, though less noisy, lacks the proper resolution to be used in tracing the curved path via connected linear segments that simulate the curved walk. The curved walk is extended from an initial scar pixel to a pixel of normal territory. For this purpose, the orientation image is projected into its sine and cosine components. Then each component is first up-sampled four times along rows and columns using a 23-point FIR interpolation filter. The coefficients of this filter are given by the formula:

$$b_k = \frac{\sin(\pi(k-11)/4)}{\pi(k-11)/4} w_k \quad k = 0, 1, 2, \cdots, 22 \tag{3.12}$$



(a) (b)

(c) (d)

Figure 3.17: *Stages of orientation diffusion process: Fig. (a) shows the orientations as a gray-level image, where black represents the 180 degrees and white is for zero degree direction. A considerable number of discontinuities or steps can be observed. Fig. (b) shows the image after ten iterations. The orientation image has been diffused with fewer discontinuities. Fig. (c) shows the progress after 50 iterations. Finally, a smooth orientation image with 90 iterations can be seen in Fig. (d).*

where w_k is a raised cosine window function that will truncate the infinite length sinc equation, is given by:

$$w_k = 0.54 - 0.46\cos(\frac{2\pi k}{22}) \quad k = 0, 1, 2, \cdots, 22 \tag{3.13}$$

also, $b_{11} = 1$ is forced to avoid indeterminate form. The interpolated sine and cosine components are then used in the inverse tangent formula to derive an interpolated orientation image.

From the interpolated orientation image and the scarred fingerprint, we can hop four times from a scarred image pixel to the appropriate neighbour in the unscarred portion of the fingerprint. The interpolated orientation image guides the movement along piece-wise linear segments with changing slope to reach the nearest point. More specifically, following equations are used to approach the neighbour of the scar pixels in the immediate vicinity.

$$\begin{aligned} x_1 &= x_0 + 0.25\cos(\theta(x_0, y_0)) \\ y_1 &= y_0 + 0.25\sin(\theta(x_0, y_0)) \end{aligned} \tag{3.14}$$

Here x_0 and y_0 are the co-ordinates of the scar pixels, detected by the second-order Gaussian derivative filter. The x_1 and y_1 are the new locations found by traveling 0.25 pixels in a straight line with slope $\theta(x_0, y_0)$. This process is repeated for four steps with new locations until it reaches the neighbour pixel in the non-scar region. The reached pixel is the very next pixel on the same curved trajectory as the scar pixel itself, providing the appropriate value to be placed at the scar pixel.

To validate the process of tracing curved neighbours with an interpolated orientation field, the tracing mechanism mentioned above is tested by plotting twenty curved neighbours on each side of a pixel for a steepest ridge tracing of the fingerprint image. Fig. 3.18 depicts the results. A reasonable match for the first few neighbours is observed, but the results become quickly unreliable as we travel far from the scarred pixel.

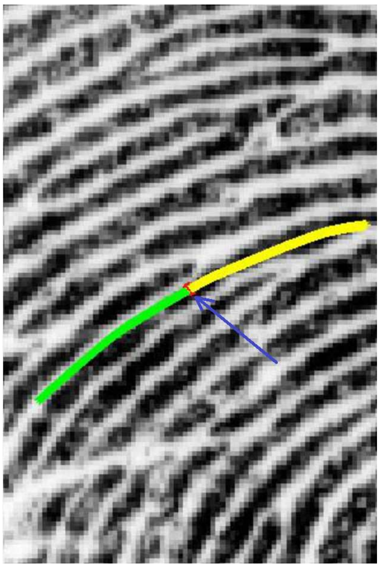
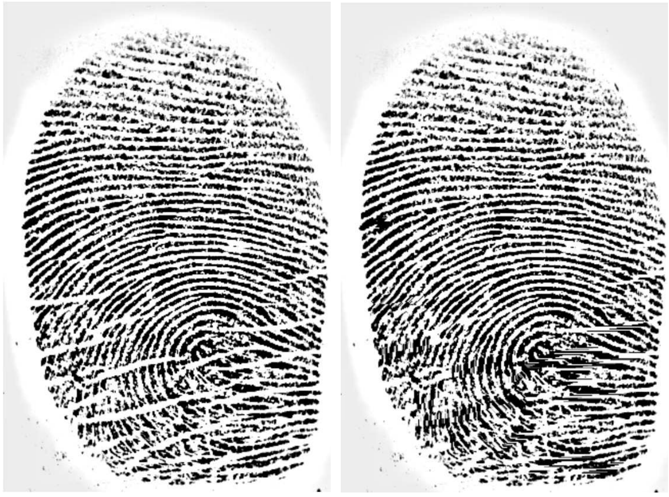


Figure 3.18: *Validation test for tracing a curved path: the red pixel is traced for 40 curved neighbours in clockwise (yellow) and anticlockwise (green) directions. A better tracing with the initial five neighbours on each side can be observed, however, as we move away the tracing starts losing the ridge.*

3.3.4 Filling scar regions

Filling scar regions in a fingerprint can be initiated once the binary mask is generated, and the interpolated orientation surface is available. The process is started with the boundary pixels of the scar regions. Then all the boundary pixels are recorded. For each boundary pixel, its replacement using the interpolated orientation image is found to reach its immediate curved neighbour of that same ridge or valley trajectory. After filling the scar pixel, the scar pixel location in the binary mask image is reset to zero. Thus, the binary scar region mask reduces in size after each iteration. The boundaries start enclosing on the scar region as shown in Fig. 3.19. Finally, after a couple of iterations, all the scar regions are appropriately filled, and the fingerprint image has been compensated.



(a) (b)

Figure 3.19: *Result of scar region filling Process: Fig. (a) shows the pre-processed fingerprint image, depicting considerable scarring. Fig. (b) shows the final result of the scar filling process, where most of the scars are compensated.*

3.3.5 Testing with synthetic scars

For evaluating the effectiveness of our scar removal strategy, synthetic scars for two test images are generated. The parameters for generating these patterns are chosen. High spatial frequency is avoided and the spatial frequency region used is matched to the normal range found in fingerprint images.

The first consists of a sine wave pattern superimposed on a polar plot. The test image is 500 rows of 500 columns, with the pattern made as a function of angle. The amplitude of the sine wave is fixed to half while the wavelength of the sine wave varies with radius. The spatial frequency, therefore, varies from high at the centre of the test image decreasing radially to the lowest at the corners. The interesting patterns observed in the centre of the image below result from *aliasing*. Specifically, the following sine function adopted.

$$f\left(x,y\right)=\frac{1}{2}+\frac{1}{2}\sin\left(50\times\arctan\left(\frac{y}{x}\right)\right), \tag{3.15}$$

where x and y are columns and rows respectively relative to the centre of the image. The radial sinusoidal pattern generated in this way is depicted in Fig. 3.20.

The second test image consists of a concentric circular pattern, where the intensity varies radially outward. The spatial frequency is made to vary with increasing radius. The frequency is low at the centre but gradually increases as the radius increases. Such an image of concentric rings is referred to as the Jahne test pattern [101]. The mathematical expression governing the pattern is depicted as:

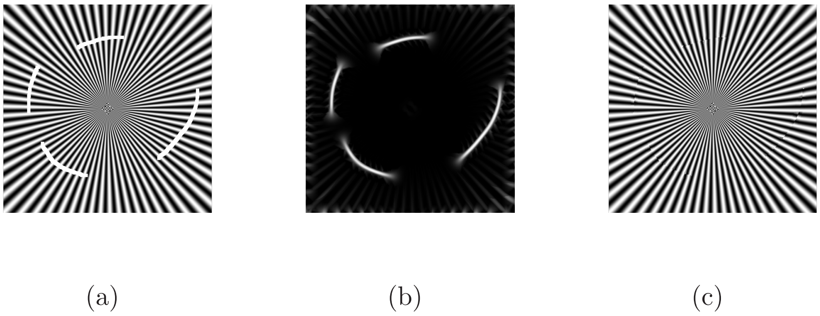


Figure 3.20: *Result of scar removal strategy for radial sinusoidal pattern: Fig. (a) shows the radial pattern with scars artificially created at four different spatial locations. The scars vary in direction as they progress from start to end. Fig. (b) shows the result of scar detection using second order Gaussian derivative filter. It can be observed that only the scar regions are highlighted while suppressing all other image features. The compensated result is shown in Fig. (c), where the scars regions are filled using directional field of the image. Though stitches as jagged response can be seen at some filling locations, the overall pattern is restored reasonably well.*

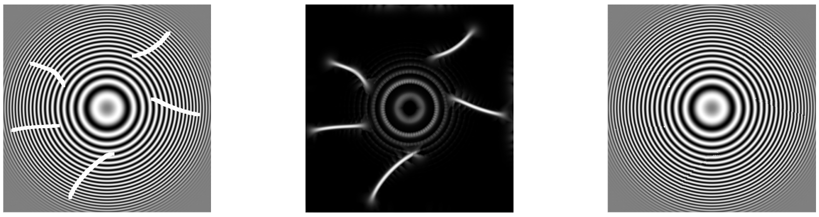
$$g(\mathbf{x})=g_0\sin(\frac{k_m|\mathbf{x}|^2}{2r_m})\left[\frac{1}{2}\tanh(\frac{r_m-|\mathbf{x}|}{w})+\frac{1}{2}\right], \tag{3.16}$$

where r_m is the maximum radius of the pattern, $\tanh(\frac{r_m-|\mathbf{x}|}{w})$ as an approximation to a step function with r_m as the location of the step and w is the width of the transition. As we move out to the corners, we set the parameters so that the maximum instantaneous frequency of $k_m = 0.5\pi$ is reached in the centre of the image edges, and the tapering function prevents aliasing artefacts from appearing [101]. The curvature at each pixel is constantly changing, making this a closer approximation to fingerprint images.

Arteficial scars are created in these test patterns approximately perpendicular to the simulated ridge pattern and then fed to the proposed scar removal process. Fig. 3.20 and 3.21 show the results of the proposed scar removal strategy. The strategy shows promise in compensating the scars and restoring to a large extent the original test patterns at scar locations. The jagged behavior at some scar locations is a manifestation of the limitation of representing angled patterns on an underlying rectangular grid.

3.4 Experimental Results

An interesting application, which motivated the whole development, was to look for solution to the access problem faced by a couple of students at university entrance. The administration installed U.areU4500 fingerprint scanners to record students entering and leaving the campus. Some complained of facing difficulties in getting through these scanners. Upon investigation, it was revealed that their fingerprints contain pronounced scars, which were not there a couple of months back at the registration time. Three samples od such scared fingerprints are shown in Fig. 3.23. These scared samples were passed on to the explicit scar removal strategy on a testing basis, and output of scar removal process is shown in the same figure. The scar removed fingerprints were presented to the straight Gabor filter for enhancement, and the outputs are as shown in Fig. 3.22. The matching score was now calculated for these treated and enhanced fingerprints and was



(a) (b) (c)

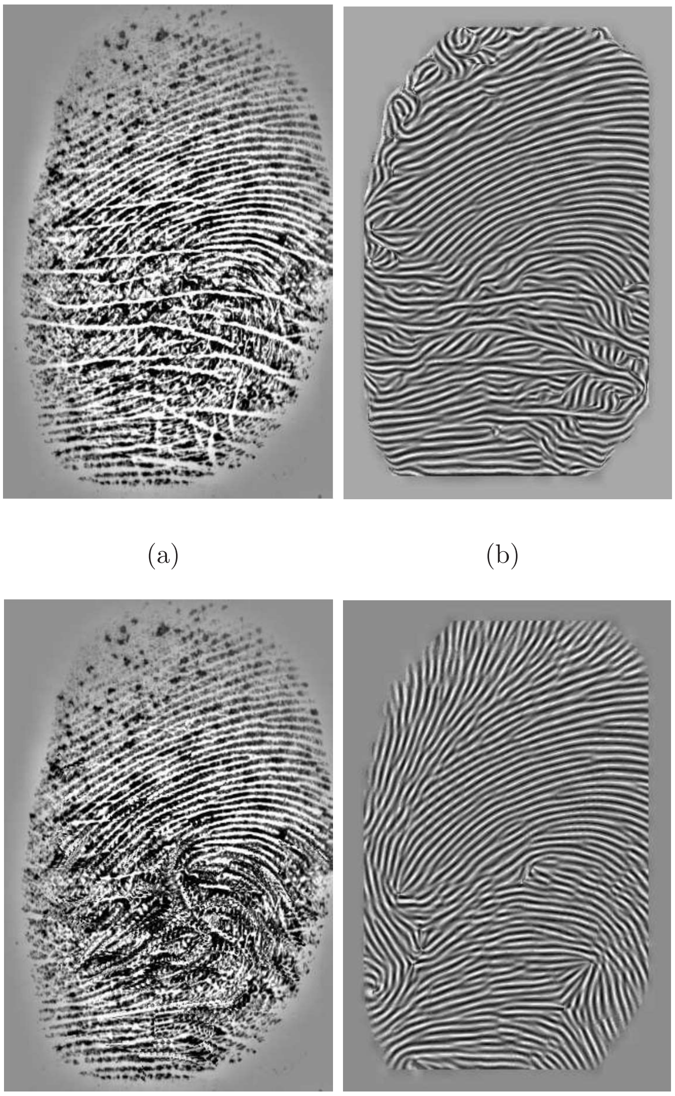
Figure 3.21: *The result of scar removal strategy for a circular sinusoidal pattern: Fig. (a) shows a radial pattern with scars artificially created at five different spatial locations. The scars vary in the direction as they progress from start to end. Fig. (b) shows the result of scar detection using second order Gaussian derivative filter. It can be observed that in addition to scar regions, some image features are also being picked by the filter. However, the image features picked are lower in intensity than the scar regions, and can be removed from the binary mask using an appropriate threshold. The compensated result is shown in Fig. (c), where the scars regions are duly filled with a circular directional field of the image. Though a jagged pattern is visible, that may be the contribution of following curved patterns on an underlying rectangular grid.*

found to be passing the threshold barrier to provide these genuine students with some relief in accessing the campus facilities.

As stated for Curved Gabor filtering, the explicit scar removal strategy also suffers from the accuracy of the orientation field. This could act as a major limitation of the proposed strategy. There are two major effects as follows:

1. Since the orientation field is helping in detection process by avoiding valleys to be part of scars, in the absence of true orientation field, many valleys appear as scars.

2. The role of orientation field is fundamental to find the neighbouring appropriate curved pixels to fill the scar region.



(a) (b)

(c) (d)

Figure 3.22: *A student sample, with Gabor filter enhancement with and without explicit scar removal.*

To visualise the impact of estimating orientation field on the scar removal strategy, three different orientation fields are simulated to be used with the proposed explicit scar removal strategy. One is the method used by Hong. et. al. [60]. The second method is estimating orientation field with a directional filter bank (DFB) as outlined in [102]. The third is the orientation field diffusion method as illustrated in [70]. A campus student challenging

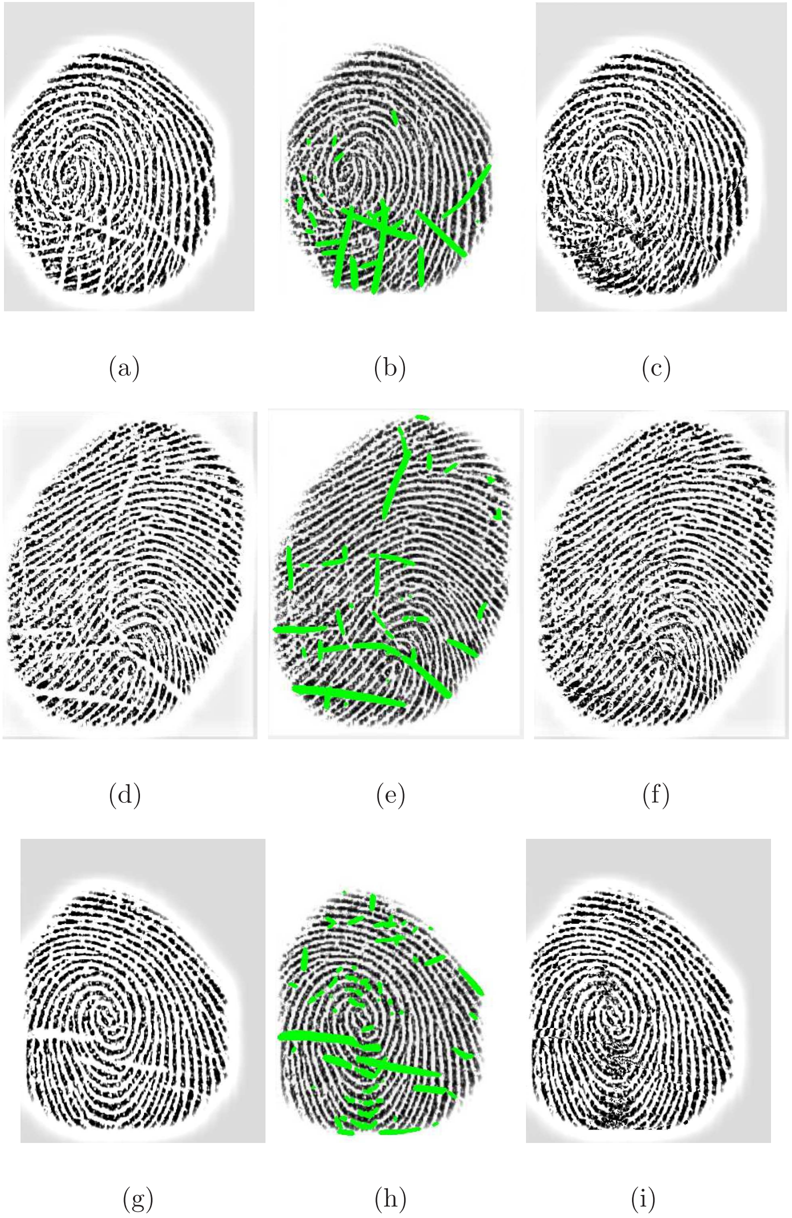


Figure 3.23: *Three Scared Fingerprints: The first column shows the original scared fingerprints as obtained at the entrance, the second is the output of Scar detection process, and third column shows the explicit scar removal strategy output.*

scarred fingerprint as shown in Fig. 3.24 is selected to see the impact of orientation field on the final output of the explicit scar removal strategy. The Fig. 3.25 reveals that Hong-based orientation field is producing poor results for scar removal strategy, while the

DFB based and the orientation diffusion based are doing almost the identical job. The orientation diffusion method is the clear winner for the explicit scar removal work, but it is an iterative algorithm, which requires an appropriate stopping rule.

To quantitatively assess the performance of explicit scar removal strategy on a public database, following procedure is adopted. A biometric fingerprint system works in either verification or identification modes. In verification, a user presents his identity, and the biometric device verifies that the identity matches with the user, this usually happens for passport checks at airports. In identification mode, no assumption of identity is made, and comparison with all templates has to be made, this could be a case of police investigating a crime scene. The assessment is made in verification mode. A matcher algorithm is selected, that assigns all authentication attempts a matching score from the interval [0, 1], where 0 means no match and 1 means full match. Next, a threshold is also selected from the interval [0,1]. If the lowest threshold 0 is selected then not only all genuine users are admitted but also all impostors are also allowed. On the other hand, if the threshold

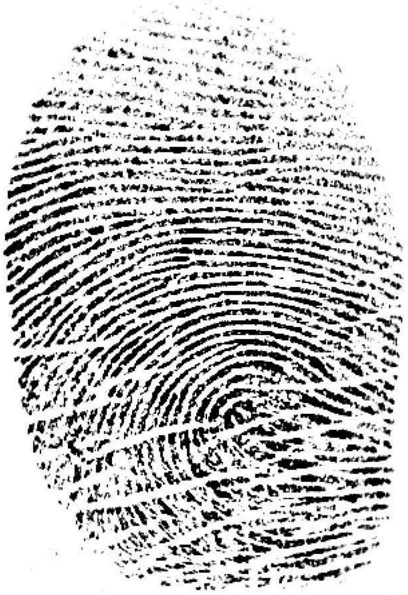
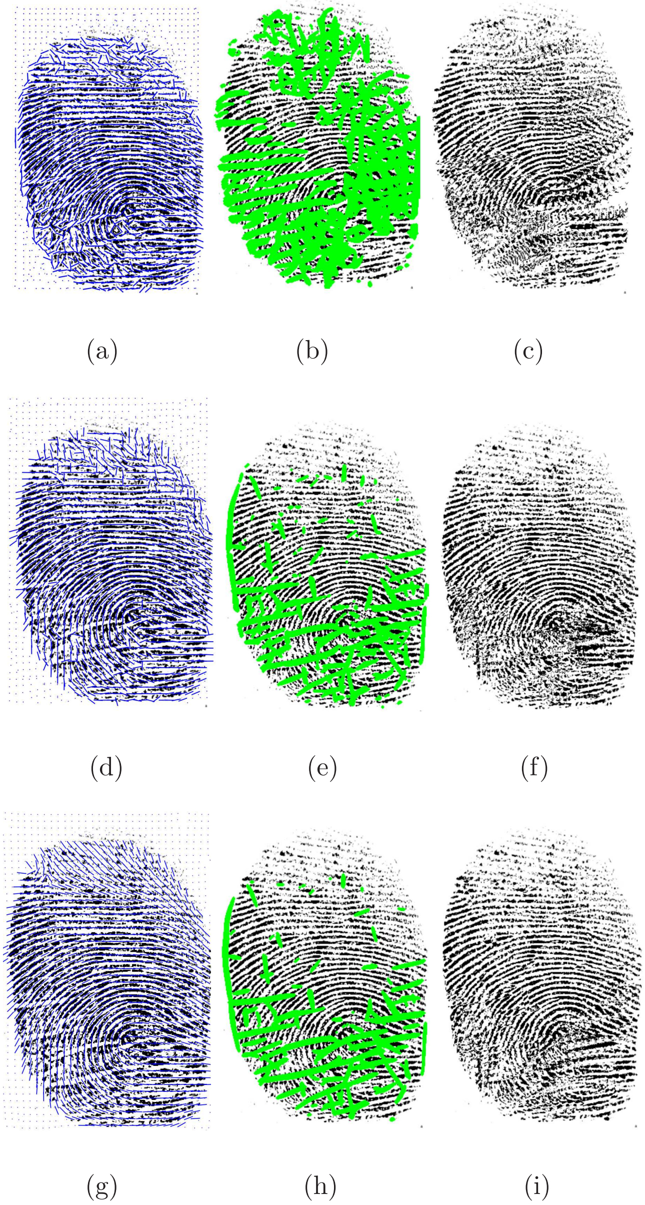


Figure 3.24: *A campus student scarred fingerprint to be used for orientation field impact experiment on explicit scar removal strategy*



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 3.25: *The Impact of Orientation Field Estimation on Scar Removal Strategy: The first column shows the estimated orientation field, the second is the output of Scar detection process, and third column shows the explicit scar compensation strategy outputs. The first row is for Hong method, the second row is the DFB method output, and the third row for the orientation diffusion.*

is set to 1, no one is admitted. So for real usage, a threshold in between [0, 1] need to be chosen. For testing, a whole bunch of operating values, a range of thresholds is selected starting from 0.2 and reaching up to 0.8 with an equal step size of 0.2. A loop is then run through these threshold values. For each value of the loop, that is for each threshold from the range selected, a pair of error rates are computed: FAR (False Accept Rate) and FRR (False Reject Rate). FAR is calculated as a fraction of impostor scores exceeding the threshold. FRR is calculated as a fraction of genuine scores falling bellow threshold. Once all the pairs corresponding to each value of the threshold is computed, a new measure with the name the Equal Error Rate (EER) can be defined. The EER value happens at a particular threshold where false accept rate (FAR) and false reject rate (FRR) are equal. In general, the FAR and FRR will rarely be equal to a threshold, so a minimum distance pair (FAR, FRR) is selected. We also note mean max/min value of the minimum distance pair.

The enhancement of low-quality images (occurring, for example, in all databases of FVC2004 [103]) and very low-quality prints such as latent prints (e.g., NIST SD27 [104]) is still a challenge. For the purpose of describing EER values for the proposed explicit scar-compensation algorithm, the publicly available fingerprint databases FVC2002, and FVC2004 were searched to gather a scarred fingerprint database of sixty-five users with more than two scars on their fingerprints. The choice of fingerprints with more than two big linear cuts is made to see the reduction in EER values before and after scar compensation. The database consists of 65 legitimate users, and each user provided his finger eight times ($65 \times 8 = 520$ images in total). Each fingerprint provides one template. Out of 8 samples, one sample of a user/person is used to create the template. The rest of the fingerprints are used for verification to receive seven genuine matching scores. All images of other users are used as impostors. Therefore, a matching score of $64 \times 8 = 512$ impostors is generated. This template generation process is repeated for all images and all users and in total $65 \times 7 = 455$ genuine scores and $512 \times 65 =$ impostor scores are generated. These scores are then used to generate pairs (FAR, FRR) to choose the best threshold, a threshold that provides the EER. In our test, the matcher "VeriFinger 5.0 Grayscale" is derived from the Neurotechnology VeriFinger 5.0 software development kit. For the verification tests, the FVC protocol is followed to ensure comparability of the results with other researchers [105], with the results listed in Table 3.1. Table 3.1 shows an improvement in EER values with the use of Explicit scar removal strategy for both Gabor as well as Anisotropic diffusion methods. Generally speaking, lower the value of EER, the better is the corresponding enhancement method. The Curved Gabor filter is found to provide best EER values for the scarred database, as it is well equipped to treat not only the scars but also separates the conglutinated ridges due to increased bandpass filtering effect.

Another set of experiments was performed on the FVC2000 database DB3_A that contains 800 fingerprint images. The program was written in Matlab and runs on a 3.40GHz Core i7 processor with 16Gb memory. The code is available in the form of a zip file that will be upload to the MATLAB Central site. Image enhancement for the FVC2000 database was performed via proposed scheme, Khan [106], Khan [107] and STFT method [71]. For feature extraction binarization/thinning based minutiae extraction method and smoothed orientation image creation proposed by Hong *et al.* [60] were used. The core points are detected by the algorithm based on the method described in [108]. Finally, fingerprint matching method using a hybrid shape and orientation descriptor [109] is used for matching purpose. This method is chosen as it outperformed many well-known methods on the FVC2002 database in the FVC2002 competition, considering that the feature set was not in pristine condition due to the chosen extraction and filtering methods. The authors of this method also claim that all known competing matching algorithms tested by the University of Bologna database were beaten by this fingerprint matching

Table 3.1: *EERS in percent for Matcher VeriFinger 5.0 on the original and enhanced fingerprints of scared database.*

Enhanced Method	Minimum	Average	Maximum
Original images	7.4	10.5	12.2
Gabor Filter [60]	4.5	5.3	7.1
Anisotropic Diffuison Filter	4.2	4.8	6.3
Gabor Filter with Scar Compensation	3.3	3.7	4.3
Anisotropic Diffuison Filter with Scar Compensation	3.5	3.8	4.2
Curved Gabor Filter [112]	2.8	3.1	3.3

method. The code of the method is also available from [110]. Table 3.2 shows the minutiae matching analysis of STFT with the proposed scar spatial domain scar removal. Table 3.2 shows that by using scar removal as pre-processing before fingerprint image enhancement can improve the overall performance of the system. In this case, on a noisy database of FVC2000 containing many images with scar type noise, scar removal enhanced the overall accuracy of the system from 56.6% to 69.2%. Also, the Youden's index [111] is increased from 0.4032 to 0.5032, where a perfect test would have a Youden index of +1.

3.5 Concluding Remarks

The research presented in this chapter is to compensate the effect of scars present in acquired fingerprints. The scars may happen due to injury where tissue is changed in coloration, and the ridge/valley structure is interrupted, or due to bending the finger by

Table 3.2: *Analysis of minutiae matching of proposed scar removal method with some existing fingerprint image enhancement techniques.*

Analysis	Proposed	Khan [106]	Khan [107]	STFT [71]
Accuracy	69.2%	59.2%	57.5%	56.6
Mis-classification	33.8%	38.7%	39.9%	43.4%
False Negative Proportion	37.6%	42.3%	48.6%	43.3%
Youden's Index	0.5032	0.4503	0.4103	0.4032
Positive Likelihood Ratio	2.3	2	1.8	1.8

putting uneven pressure during acquisition process with a fingerprint scanner. An explicit scar removal strategy is proposed, that can be regarded as a pre-processing step where the orientation field is used to select the right candidate for filling scar regions. The strategy relies on the fact that in these linear scars, the ridge/valley pattern is still intact across the scar region. Using this information, the scar boundary is filled with appropriate normal region pixels using the local orientation field. The explicit scar removal method shows promise in terms of better matching scores associated with minutia points. It is believed that these pre-processed scar-compensated fingerprints will boost the reliability of an AFIS to be used in public access systems.

One serious contender to the Explicit scar removal strategy is the Curved Gabor filter (GF). The curved GF has the capability to do both: remove scars and separate falsely conglutinated ridges in wet fingerprints. The limiting factor for Curved GF is the extraction of the orientation field and estimation of Ridge frequency (RF). Computations can quickly rise if Curved GF is employed for the whole fingerprint. On the other hand, using explicit scar removal algorithm at specified scar locations followed by flat Gabor filtering can save computations with performance closer to that Curved Gabor filter. The limitation of the Explicit scar removal strategy is the fact that it is of little use for falsely conglutinated ridges cases.

To deal with scars and false conglutinated ridges at the same time, a better strategy could be suggested in the form of first using scar removal strategy to fill big linear scar cuts, spread across the surface of the image. Then a follow-up smoothing performed with, Curved Gaussian filter (a variant of Curved Gabor Filter), instead of actual Curved Gabor filter, to separate conglutinated ridges to a large extent. A Curved Gaussian filter can be defined along the bent ridge and valley structure, therefore, will be equally effective but computationally less expensive than Curve Gabor filter due to its limited kernel support region. Once the scars are filled and conglutinated ridges are diminished, the flat Gabor filtering can now be called in to impart the required clarity to ridge/valley parallel structure for subsequent binarization. This can constitute a computationally cheaper but overall effective solution for low-quality fingerprint enhancement and can well be posed as a viable alternative to the Curved Gabor filter. This is the topic of our future research.

Chapter 4

Stopping Criterion for Linear Anisotropic Image Diffusion: A Fingerprint Image Enhancement Case[1]

This chapter is also an adapted version of a journal article. While the section headings from the journal article have been retained, the figures, equations, tables, and references have been re-numbered and are in line with the thesis format. This chapter presents a software-based technique for fingerprint image enhancement in which the entropy change for an anisotropic diffusion is investigated. A unique peak is found, associated with a blurring of the dominant structure, that provides a reasonable stopping rule for the anisotropic diffusion process. With respect to our proposed design, this chapter falls in

[1] Published as: Tariq M. Khan, Mohammad A. U. Khan, Yinan Kong and Omar Kittaneh,"Stopping Criterion for Linear Anisotropic Image Diffusion: A Fingerprint Image Enhancement Case," *EURASIP Journal on Image and Video Processing* , (2016) 2016: 6. doi:10.1186/s13640-016-0105-x

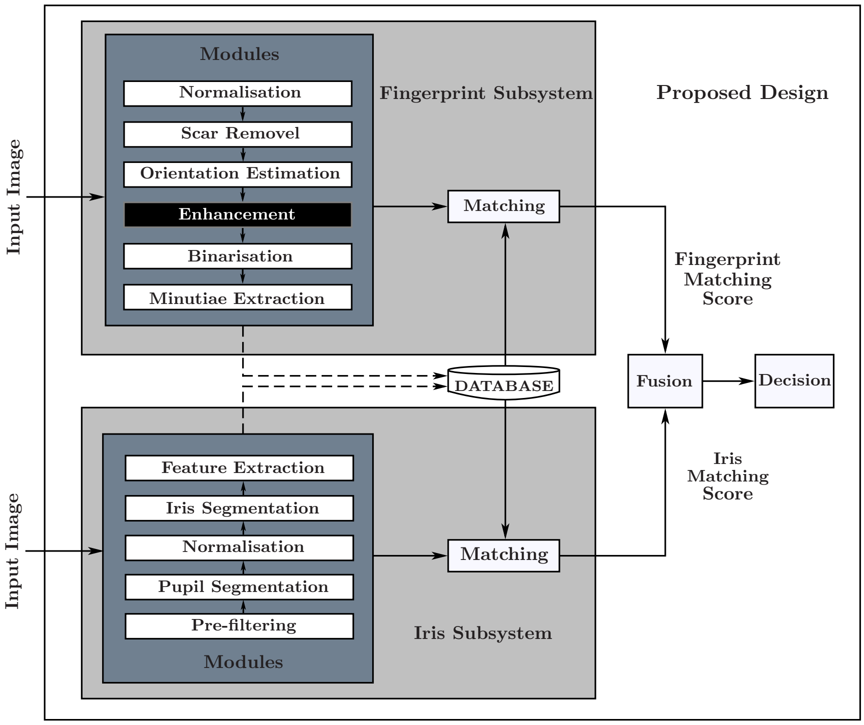


Figure 4.1: *Modules of the proposed design under study*

fingerprint subsystem. It covers one module of fingerprint subsystem, as highlighted in Fig. 4.1

4.1 Abstract

Images can be broadly classified into two types: Isotropic and Anisotropic. Isotropic images contain largely rounded objects while anisotropics are made of flow-like structures. Regardless of the types, the acquisition process introduces noise. A standard approach is to use diffusion for image smoothing. Based on the category, either isotropic or anisotropic diffusion can be used. Fundamentally, diffusion process is an iterated one, starting with a poor quality image, and converging to a completely blurred mean-value image, with no significant structure left. Though the process starts by doing a desirable job *of cleaning noise and filling gaps, called under-smoothing, it quickly pass into an over-smoothing phase where it start destroying the important structure. One relevant concern is to find the boundary between the under-smoothing and over-smoothing regions. The spatial entropy change is found to be one such measure that may be helpful in providing important clues to describe that boundary, and thus provides a reasonable stopping rule for isotropic as well as anisotropic diffusion. Numerical experiments with real fingerprint data confirm the role of entropy-change in identification of a reasonable stopping point where most of the noise is diminished and blurring is just started. The proposed criterion is directly related to the blurring phenomena that is an increasing function of diffusion process. The proposed scheme is evaluated with the help of synthetic as well as the real images, and compared with another state of the art schemes using a qualitative measure. Diffusion of some challenging low-quality images from FVC2004 are also analysed to provide a reasonable stopping rule using the proposed stopping rule.*

4.2 Introduction

In image processing problems, many times one come across the task to enhance flow-like structures, for instance, the automatic assessment of wood surfaces or fabrics, fingerprint image analysis, scientific image processing in oceanography [113], seismic image analysis [114], or sonogram image interpolated for Fourier analysis [115]. All image as mentioned above have one thing common; they contain elongated structures [107, 116, 117]. Such images can be referred to as *anisotropic*. The isotropic, by contrast, is an image category having largely round objects. The isotropic as well as anisotropic images, once acquired from their respective sources are mostly noisy. The noise treatment is different based on the category they belong. The case of noise smoothing for anisotropic images is more interesting and is the focus of research presented here.

Classifying images into their category will help to devise a proper noise removal strategy for them. The authors in [118] suggested to use local anisotropy strength as a measure for an image to classify as anisotropic or isotropic. They later extended their anisotropy strength definition to construct a complete flow-coordinate system for anisotropic images. Their proposed anisotropy strength measure computation can be summarised as follows. First, the image $L(x, y)$ is smoothed with a Gaussian of small standard deviation. The result $C(x, y)$ is then differentiated in $x-$ and $y-$ direction to form $C_x(x, y)$ and $C_y(x, y)$, respectively. Next the covariance matrix components $J_1(x, y) = 2C_x(x, y)$ and $J_2(x, y) = C_x^2(x, y) - C_y^2(x, y)$, and $J_3(x, y) = \sqrt{C_x^2(x, y) + C_y^2(x, y)}$ are computed. The components are smoothed again with a larger Gaussian. The local orientations and their anisotropy strength measure are computed as:

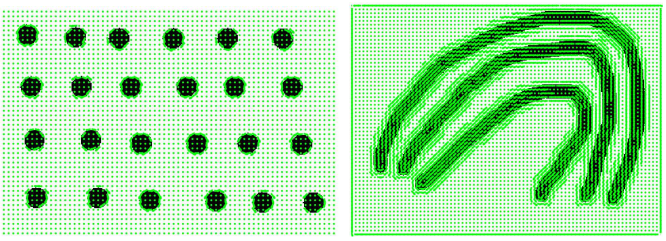
$$\theta(x, y) = \frac{\arctan\left(\frac{J_1(x,y)}{J_2(x,y)}\right)}{2}, \tag{4.1}$$

and

$$\chi(x, y) = \frac{\sqrt{J_1^2(x, y) + J_2^2(x, y)}}{J_3(x, y)}. \tag{4.2}$$

Applying this definition to our test images, that is, Blackball and Curves image, will result in a graphical display as shown in Fig. 4.2. The local flow directions are depicted by the orientations of the small needles superimposed on the image. The length of each needle is drawn proportional to the amount of local anisotropy at that pixel point. It is noted that the Blackball image is largely isotropic with no preferred local directions, whereas the Curves image showed a profound anisotropic character, largely in the vicinity of the elongated structures. This justifies labelling Blackball image as isotropic and the Curves image as anisotropic.

The rest of this chapter is organised as follows. Related work is discussed in Section



(a) (b)

Figure 4.2: *Anisotropy strength measure. The figure reveals anisotropy strength measure in the form of an array of needles on top of the image. The length of needles is representative of anisotropy strength and the needle direction is an estimation of local flow. The blackball image is largely isotropic with little amount of anisotropy at almost all the points in the image. However, the curve image on the right is largely anisotropic, with a strong strength measure appearing around the elongated feature of interest. a represents the black ball image. b is the curve image*

4.3. In section 4.4, a discrete image as a spatial distribution is discussed. Spatial Entropy of Linear Isotropic Diffusion Process is described in section 4.5. Section 4.6 talks about Spatial Entropy of a Linear Anisotropic Diffusion Process followed by Results and Discussion in Section 4.7. Finally, the chapter is concluded in Section 4.8.

4.3 Related Work

The research concerned here is to smooth noise present in fingerprint images (a representative of anisotropic class) without affecting their ridge/valley pattern. This aim can be conveniently served in a *scale-space* construction. A scale-space framework describes noisy image as a stack of progressively evolving many smooth images, each one with their corresponding scale [119]. The stack is ordered in increasing smoothness scale, where the scale varies in fine-to-coarse. The fine-to-coarse transformation is implemented, in general, by a linear isotropic diffusion process, governed by a partial differential equation (PDE) as follows.

Let $L(x, y)$ denote a noisy grayscale input image and $L(x, y; t)$ be an evolving image at scale t, initialized with $L(x, y; 0) = L(x, y)$. Then, the linear isotropic diffusion process can be defined by the equation

$$\frac{\partial L}{\partial t} = \nabla \cdot (c\nabla L) = c\nabla^2 L. \tag{4.3}$$

This equation appears in many physical processes [120,121]. In the context of heat transfer, it is referred to as the famous *heat equation.* For image processing, the amount of heat is replaced with the intensity value at a certain location. The diffusivity parameter c is constant across the image, making it a *linear* isotropic equation. The linear isotropic equation has an elegant solution $L(x, y; t) = G_{\sqrt{2ct}}(x, y) * L(x, y)$, where $G_\sigma = \frac{1}{2\pi\sigma^2}\exp(-\frac{x^2+y^2}{2\sigma^2})$. This solution provides the required interpretation in the form of low-pass filtering. Due to low-pass nature of this diffusion, as it progresses from fine scale images to coarser images, the blurring intensify and may result in removing significant image structure, typically edges, lines or other details, well before it had taken care of the noise. To protect the structure in a diffusion process, the diffusivity parameter should be made dependent on some characterization of image structure. This results in the famous nonlinear isotropic diffusion process, proposed by [64]. The diffusivity now becomes a function of gradients, so at the edge point the diffusion is completely inhibited and in smooth regions diffusion allowed. However, computing gradients for a noisy image is an ill-posed problem. A remedy was pointed out by [122], that suggests the use of Gaussian smoothing before computing gradients. This modification lays the foundation for a well-behaved *non-linear* isotropic diffusion process. Later on, instead of inhibiting diffusion at edge points, it was thought of to steer the diffusion in the direction parallel to the edge [106,123–125] rather than across it. This paved the way for the use of Diffusion matrix. This evolved the current form of *non-linear* anisotropic diffusion. The diffusion matrix based equation is defined as:

$$\partial_t L = \nabla \left(D \nabla L \right), \tag{4.4}$$

where D is the 2×2 diffusion matrix. The eigenvectors of the diffusion matrix provide the required steering while the eigenvalues as a function of gradients, add the non-linearity character. In our wish to keep connected with the Gaussian convolution interpretation that provides a mathematical tractability to the whole process, the research reported here is restricted to the linear anisotropic diffusion case. For that, the eigenvalues of the diffusion matrix are kept fixed. It is found that the Gaussian convolution connection is also useful for linking anisotropic diffusion with its earlier counterpart isotropic diffusion in a more natural way. The support for this modification, came from the argument made in [124], that a non-uniform Gaussian can act as a solution of the Anisotropic Gaussian scale-space as long as the diffusion matrix is *spatially constant*, i.e. it does not depend on (x, y) spatial location. Keeping in line with this argument, only spatially-invariant Diffusion matrix is used, however, the steering was allowed. This leaves us with so-called *linear anisotropic diffusion* process. The constant eigenvalues are responsible for the linear part of the name, while the steering of the eigenvectors is what provided the word anisotropic in the nomenclature. The linear anisotropic diffusion equation has a convolution solution with a non-uniform Gaussian of the form:

$$G_{\lambda_u,\lambda_v}(u, v) = \frac{1}{\sqrt{2\pi}\lambda_u} \exp(-\frac{u^2}{2\lambda_u^2}) \quad \frac{1}{\sqrt{2\pi}\lambda_v} \exp(-\frac{v^2}{2\lambda_v^2}) \tag{4.5}$$

where (u, v) are the rotated coordinates obtained using eigenvectors of diffusion matrix. The eigenvalues λ_u, λ_v represent the standard deviations of the Gaussian in u and v direction, respectively. Normally, for noisy images, one of the eigenvalue is set to be much smaller than the other one, resulting in a non-uniform Gaussian function with more generalized elliptical support.

Searching for a suitable linear anisotropic diffusion strategy for noisy images in literature, we stumble upon considerable activity regarding the impact of a non-linear anisotropic diffusion equation on noisy images. The non-linear anisotropic literature is used as a stepping stone to reach a linear anisotropic diffusion strategy. The idea of non-linear anisotropic diffusion was pioneered by Nitzbeg *et al.* [126] and Cottet *et al.* [122]. Later on, Weickert [115] put forward a formal method for enhancing the elongated structure, referred to as Coherence Enhanced Diffusion (CED). The CED works by steering the diffusion process in a particular direction with the help of a spatially varying diffusion matrix. The design was further generalised by adopting a diffusion matrix to learn the local structure iteratively [127]. Since smoothing elongated structure is desired, the CED procedure comes in handy. The CED is adopted as it is, but with one major modification. That is, the eigenvalues are forced to be independent of spatial position without disturbing the eigenvectors. Thus, our proposed linear anisotropic diffusion process will steer the non-uniform Gaussian to lay along the structure, but its size will remain constant regardless of the position. Towards the end, we will describe another variant of CED, where even the steering part of the diffusion matrix will also be pre-computed and kept constant throughout the evolution process. This is referred to as linear oriented diffusion process.

The suggested linear anisotropic process for anisotropic images are confronted with one basic problem: When to stop the diffusion. For the case of a noisy image, the diffusion process initializes with an under-smooth situation that ultimately turns into an over-smooth one (the mean-value image at the end with no structure). Over-estimating stopping time will result in an over-smoothed blurry image while under-estimating may leave significant noise in the image. Therefore, it is crucial that an appropriate time is selected in an automatic way. The literature activity in this respect can be divided into two broad categories. One that deals with stopping criterion selection in additive noise model setting. These methods adopt the stopping time by treating the noisy image as the result of a noise addition, where the correlation between the diffused image and the initial noisy image minimised [115]. The authors in [128] introduced a multigrid algorithm using a normalised cumulative periodogram. A frequency approach to the problem was presented in [129]. Whereas, [130] uses the extent of noise smoothing in every iteration as a stopping parameter for diffusion. Later on, a spatially-varying stopping method was introduced that increased the computational cost significantly [131]. By identifying it as a Lyapunov functional of a large class of scalar-valued nonlinear diffusion filters, Weickert [132] introduced decreasing the variance of an evolving image as a stopping tool.

Since additive noise, model may break down for some real-world images, where noise manifests itself in the form of gaps in regular ridge structures. Therefore, a second category of stopping rule was evolved. The category deals with examining entropy profile of the diffused image and proposed stopping criterion for the evolving image entropy distance from that of the entropy of the original noisy image [115]. The idea of local image entropy was introduced in [133], where the measure of local entropy defines the segmentation boundaries in multiple-object images. Local image entropy definition can be extended to define a global characteristic of the scale-space image, that is spatial entropy [134].

The research work reported here takes an investigative look at the stopping rule concerning the change in spatial entropy of an image as it goes through diffusion process. The connection, between last peak in spatial entropy curve and the size of the image structure, is found to be related to the start of significant information loss. This observation paves the way to the hypothesis that peak entropy change will happen at the time instant on diffusion time axis when dominant image structures just start blending with the background right at their boundaries. This finding, substantiated by extensive empirical evidence provided here, motivated us to put forward the idea that a maximum entropy change may well be posed as a good stopping time for the diffusion process.

4.4 A discrete image as spatial distribution

Consider a discrete fingerprint image $L(x, y)$, where x is the row index and y is the column index. This discrete image can be realised as spatially distribution light intensity [135]. Each spatial location that is (x, y) in the image registers the number of light quantum-hit. In this way, we may define

$$p(x, y) = \frac{L(x, y)}{\sum\limits_{x}\sum\limits_{y} L(x, y)} \tag{4.6}$$

This spatial probability perspective was found to corresponds very nicely with the theory of scale-space [136]. As we move higher in scale-space for an image, and the spatial smoothing is high, or equivalently the spatial uncertainty increases. In the limit, the spatial distribution becomes close to uniform distribution. The spatial entropy of an image is given as

$$H_t(L) = -\sum_{x}\sum_{y} \frac{L(x, y; t)}{\sum\limits_{x}\sum\limits_{y} L(x, y; t)} \log\left(\frac{L(x, y; t)}{\sum\limits_{x}\sum\limits_{y} L(x, y; t)}\right) \tag{4.7}$$

As stated in [135], the spatial entropy of the image increases monotonically towards an equilibrium state $\log N$, where N is dimension $N = rows \times columns.$

4.5 Spatial Entropy of Linear Isotropic Diffusion Process

The linear diffusion process implemented by so-called heat equation is the oldest and well-investigated noise smoothing process in image processing domain. The linear diffusion process can be visualised as an evolution process with an artificial variable t denoting the *diffusion time*, where the noisy input image is repeatedly smoothed at a constant rate in all directions. No preference to any direction is what justifies the name *isotropic*. This evolution results in *scale space* representation of the noisy image. As we move up to coarser scales, the evolving images become more and more simplified since the diffusion process removes the image structures present at finer scales. In the process, noise also gets smoothed as it is considered a smaller size object while diffusion just reaches the point of touching the boundaries of the large dominating structure.

During the process of diffusion from fine scale image to the higher coarser scale images, the mean of the resulting image remains constant with a monotonic decrease in variance (a second-order statistics [123]). Later on, it was found that spatial entropy associated with linear isotropic diffusion process also rises smoothly in a monotonic fashion [134]. Motivating by the smoothness of the spatial entropy graph for the diffusion process, the first derivative of the entropy function on natural scale parameter $\tau = \log(t)$ was investigated. It was shown that entropy change graph do show important peaks related to dominating structures present in the original fine scale image. However, their experiments did not involve smoothing noisy images, and the authors fell short of suggesting to use these peaks as stopping criterion. The empirical evidence is provided here to show that once a linear isotropic diffusion process is involved in smoothing noisy images, these peaks will come at a much later stage in diffusion time. Therefore, most of the noise being low size structure already wiped by the process, and thus the peaks could be regarded as a suitable stopping time. This proposition is tested by tracking experimental data.

To provide a quantitative measure for checking our test results, two binary statistical measures are used: Sensitivity and Specificity. This is due to the use of a binary image as input test, and the final diffused image is thresholded to come up with the final binary output image. Since we are dealing with binary images, the two measures suits us. The measures deals with comparing the output binary image A with a standard ground truth image B. Let us first define four related quantities: True Positive TP (The black pixels in image A are also black in image B), False Positive FP (The black pixels in image A are white pixels in image B), False Negative FN (The black pixels of image B are identified as white in image A, that is we missed the true black pixels), and True Negative TN (The white pixels in image A are same as white pixels in image B). Sensitivity is given by:

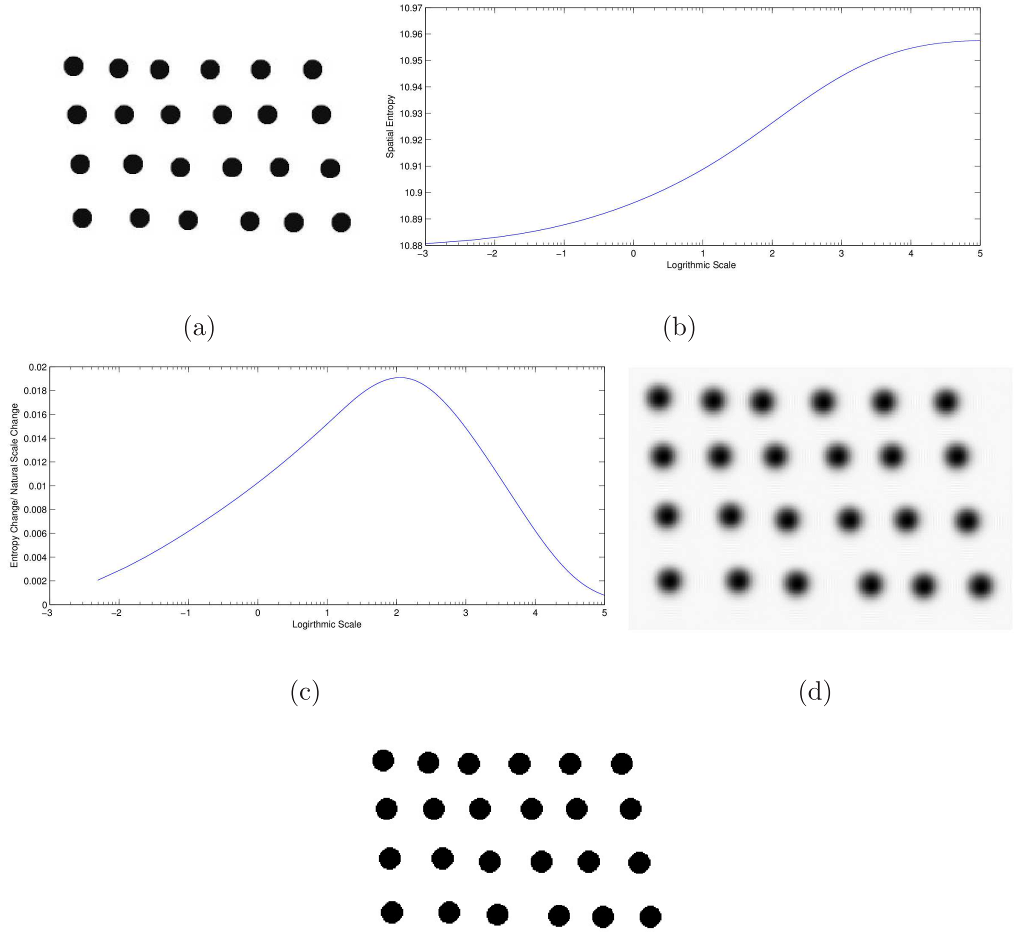
$$Sensitivity = \frac{number\, of\, TP}{number\, of\, TP + number\, of\, FN}. \tag{4.8}$$

Specificity is more concerned with

$$Specificity = \frac{number\, of\, TN}{number\, of\, TN + number\, of\, FP} \tag{4.9}$$

First, a linear isotropic diffusion process is conducted for the image without noise. Fig. 4.3(b) shows the entropy curve with natural scale parameter. The monotonic behaviour of entropy curve is noted. The curve starts increasing from a low value and moves onwards to an almost stable asymptotic value on a much larger scale. The regularity of the entropy curve motivates us to compute its derivative on the natural scale parameter. The entropy change curve for this image diffusion process is depicted in Fig. 4.3(c). One clear peak in the graph is observed, corresponding well with the radius of the black balls. If the linear diffusion process is stopped at a scale where the peak in entropy-change happens, then output resulting diffused image is displayed in Fig. 4.3(d). It is observed that diffused image is still intact with all the black balls showing their characteristic black colours, with diffusion just started at the boundaries of these balls. Hoping that this peak in entropy change will remain fixed at this scale with the noise added to the image, the best possible stopping time will be the scale of the peak. The sensitivity and specificity numbers for the comparison of the output diffused binary image with the original are 88% and 96%.

To investigate the shape and location of the peaks in entropy change with noise added images, we start with lower SNR images. The black balls image is considered with Gaussian noise added, such that its SNR reduces to 2. The black ball image with SNR=2



(a)

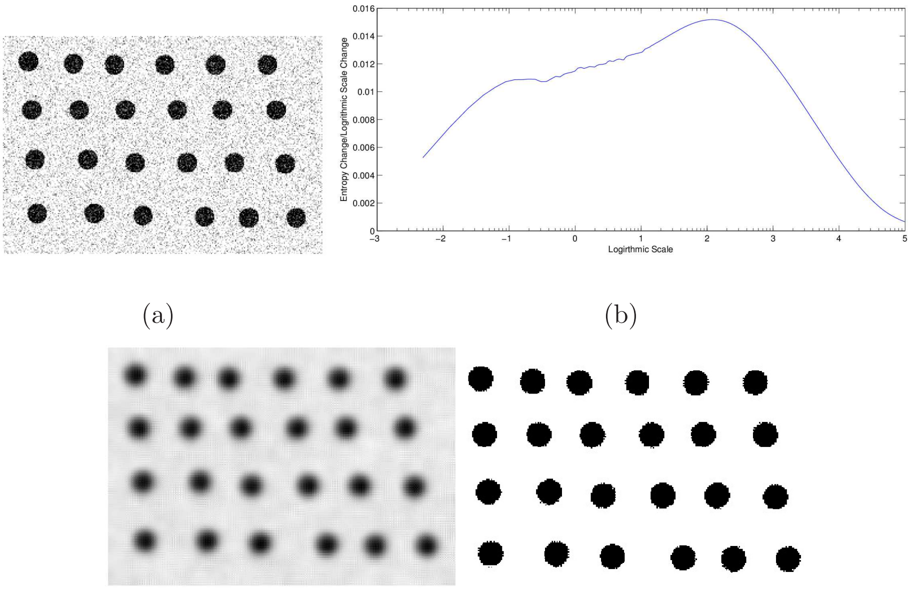
(b)

(c)

(d)

(e)

Figure 4.3: *Linear isotropic diffusion process. a) shows a black ball test image with white background. The features present in the image are isotropic in shape with a constant radius of two pixels. b) shows the smooth spatial entropy graph resulting from diffusion process on natural scale parameter. The entropy change with natural scale change is displayed in (c), where the peak corresponds to the size of the black balls. The diffused image resulting from stopping the diffusion process at the location of the peak in entropy change is shown in (d). The diffused image is converted to binary image using Otsu optimal threshold of 0.63. The final binary image is displayed as (e).*



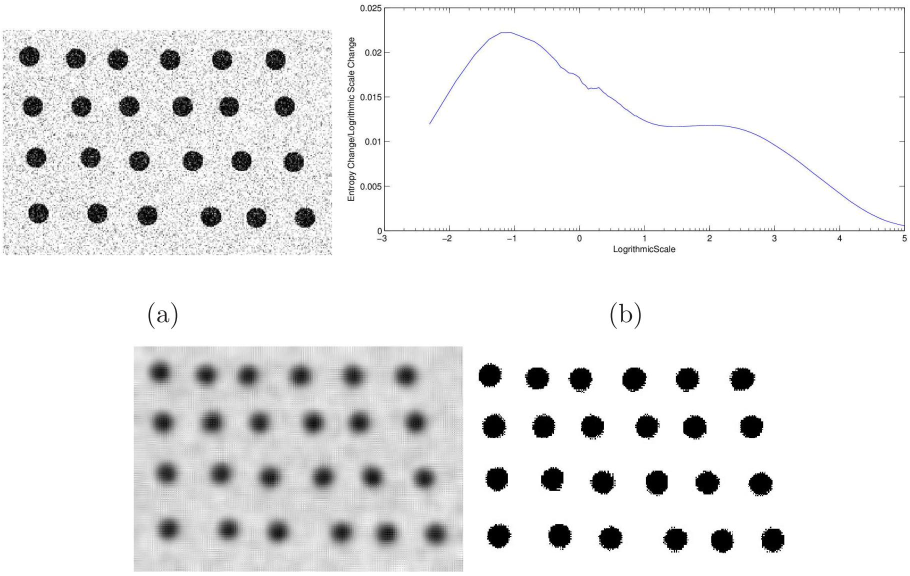
(a) (b)

(c) (d)

Figure 4.4: *Linear isotropic diffusion process for noisy image. a) shows a noisy black ball test image with white background. The zero mean Gaussian Noise added such that SNR reduces to 2 dB. b) shows the spatial entropy-change graph resulting from diffusion process on the natural scale parameter for a noisy image. Two peaks can be observed, where the first peak is the result of adding Gaussian noise, and the second peak is representing the characteristic size of the black balls. The diffused image resulting from stopping the diffusion process at the location of the second peak in entropy change is shown in (c). Binarized image as a result of the threshold, set to the mean value of the diffused image results in (d)*

dB is depicted in Fig. 4.4(a). The linear diffusion process was conducted for this noisy image, to mitigate the effect of Gaussian noise. The resulting entropy change graph is displayed in Fig. 4.4(b). We see two peaks in the graph. The first peak is largely the contribution of the noise added to the image. The second peak is due to the presence of black balls, at the same location where we saw it before in the clean image entropy-change graph. This validated the claim made in [134], that peaks in entropy change graphs are representative of the corresponding sizes of the structures present in the images. The linear diffusion process can stop at the location of the second peak, the resulting output diffused image is shown in Fig. 4.4(c)). The image clearly shows a diffused image where largely the noise is smoothed with the black balls still intact. The diffused image can be binarized by using its histogram, clearly showing a valley between black and white bars. Doing so, the image of Fig. 4.4(d) is reached, with sensitivity and specificity numbers, are 85% and 91%.

To further investigate the entropy change graph of a noisy image, the black ball image are severely degraded with a large amount of Gaussian noise till its SNR drops to -3 dB. The noisy black ball image is depicted in Fig. 4.5(a). The linear diffusion process is applied to this noisy image, with the resulting entropy change graph displayed in Fig. 4.5(b). The presence of two peaks is observed, as previously did in a less noisy image. However, this time, the peak associated with noise is much large in amplitude to the peak of the black balls. This clearly is the outcome of a large amount of noise added to the image pixels. The second peak, though small in amplitude, is still present at the same location as that of clean image entropy-change graph. By stopping the linear diffusion process at the second peak location, we get the diffused image is shown in Fig. 4.5(c). By converting this diffused image by selecting a threshold from its histogram, we reach the binary result as displayed in Fig. 4.5(d), having sensitivity and specificity numbers as 78% and 88%.



(a) (b)

(c) (d)

Figure 4.5: *Linear isotropic diffusion process for noisy image. (a) shows a noisy black ball test image with white background. The zero mean Gaussian Noise added such that SNR reduces to -3dB. b) shows the spatial entropy-change graph resulting from diffusion process with respect to natural scale parameter for noisy image. Two peaks can be observed, where the first peak is much larger in amplitude than the second peak. The diffused image resulting from stopping the diffusion process at the location of the second peak in entropy change is shown in (c). Binarized image as a result of threshold, set to the mean value of the diffused image results in (d)*

4.6 Spatial Entropy of a Linear Anisotropic Diffusion Process

In this section, spatial entropy analysis is carried out for the anisotropic diffusion process. What we are looking for is the finding whether we will get a smooth spatial entropy increasing function, and then will we get a distinct peak in the entropy change curve for the anisotropic diffusion process.

The anisotropic scale-space for the image $L(x, y)$ can be constructed by the diffusion equation:

$$\frac{\partial L}{\partial t} = \nabla \left(D \nabla L \right), \tag{4.10}$$

where D is the 2×2 diffusion matrix, adapted to the local image structure, via a structural descriptor, called the second-moment matrix μ, defined as:

$$S = \begin{pmatrix} s_{11} & s_{12} \\ s_{12} & s_{22} \end{pmatrix} = \begin{pmatrix} L_{x,\sigma}^2 & L_{x,\sigma} L_{y,\sigma} \\ L_{x,\sigma} L_{y,\sigma} & L_{y,\sigma}^2 \end{pmatrix} \tag{4.11}$$

where L_x^2, $L_x L_y$ and L_y^2 represents the 2nd order Gaussian derivative filters, in the x and y directions. This symmetric 2×2 matrix has two eigenvalues λ_1 and λ_2, given by:

$$\begin{aligned} \mu_1 &= 1/2 \left(s11 + s12 + \alpha \right) \\ \mu_2 &= 1/2 \left(s11 + s12 - \alpha \right), \end{aligned} \tag{4.12}$$

where

$$\alpha = \sqrt{\left(s11 - s22 \right)^2 + 4 s12^2} \tag{4.13}$$

The second-moment matrix comes with two eigenvectors. The first normalized eigenvector can be written as $(\cos\theta, \ \sin\theta)^T$, and the second orthogonal eigenvector comes out to be as $(-\sin\theta, \ \cos\theta)^T$. One of these eigenvectors is parallel, and the other is perpendicular to the structure. The parameter θ represents the local orientations of the given image. What observed here is that eigenvalues are dependent on the local structure. In order to transform CED process into a linear anisotropic process, fixed values are assigned to the eigenvalues. Specifically, the eigenvalue associated with eigenvector that goes parallel to the structure have given a larger value than that of the eigenvalue of an eigenvector that is perpendicular to the structure boundary. Our specific choice of λ_1 and λ_2 for this experiment are

$$\begin{aligned} \lambda_1 &= 0.1 \\ \lambda_2 &= 1 - 0.1, \end{aligned} \tag{4.14}$$

with a step size of 0.01 to provide a stable diffusion process.

The diffusion matrix D can now be reconstructed with help of its structure-invariant eigenvalues and structure-dependent eigenvectors as:

$$\begin{aligned} d11 &= \lambda_1 \cos^2\theta + \lambda_2 \sin^2\theta \\ d12 &= (\lambda_1 - \lambda_2)\sin\theta\cos\theta \\ d22 &= \lambda_1 \sin^2\theta + \lambda_2 \cos^2\theta \end{aligned} \tag{4.15}$$

Once the diffusion matrix is constructed, the evaluation process is set to start. The diffusion process proceeds in four steps.

1. Calculate the second-moment matrix for each pixel.

2. Construct the diffusion matrix for each pixel.

3. Calculate the change in intensity for each pixel as $\nabla(D\nabla L)$.

4. Update the image using the diffusion equation as:

$$L^{t+\triangle t} = L^t + \triangle t \times \nabla(D\nabla L) \tag{4.16}$$

This monotonic decreasing behaviour of the image variance is also evident in the graph depicted in Fig. 4.6 when we are diffusing our fingerprint image shown in Fig. 4.6. What can be seen from the graph is that it is fast decreasing in the beginning, but towards the end, it becomes saturated, providing convergence. Thus, by bounding the relative change in the variance one can define the diffusion stopping rule. However, this rule does not guarantee an optimal time to stop the process. It is based on the user defined ratio of diffused image variance to that of initial image variance. This ratio might be useful if we want to compare various diffusion schemes. Its utility to provide a well-diffused image with all the important structure cleaned but intact may be limited. Under the CED process, the fingerprint image becomes strongly coherent as the number of iterations increased. In other words, as the scale increases, the image becomes diffused with a corresponding

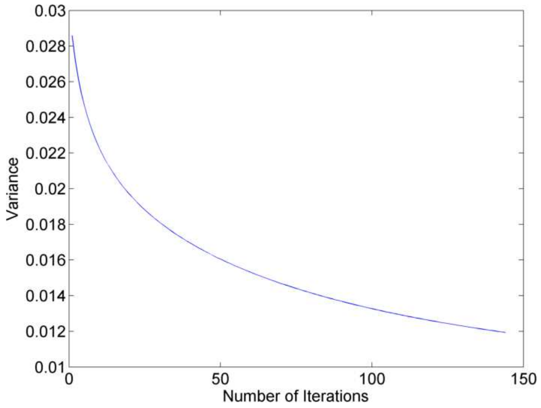


Figure 4.6: *This graph shows the monotonic decreasing behavior of the variance of the image for coherence enhanced diffusion(CED)*

change in its spatial distribution. Taking $p_t\left(x,y\right)=\frac{L(x,y;t)}{\sum\limits_x\sum\limits_y L(x,y;t)}$ and $C=\sum\limits_x\sum\limits_y L\left(x,y;t\right)$, we get

$$H_t\left(L\right)=-\sum_x\sum_y p_t\left(x,y\right)\log p_t\left(x,y\right) \tag{4.17}$$

Now, we track the change in entropy with respect to natural scale parameter $\tau=\log t$. The natural scale parameter is defined in [137]. The entropy change is thus,

$$\frac{dH_t\left(L\right)}{d\tau}=-\sum_x\sum_y\frac{d}{d\tau}\left(p_t\left(x,y\right)\log p_t\left(x,y\right)\right) \tag{4.18}$$

After some mathematical manipulations, reach to the equation:

$$\frac{dH_t\left(L\right)}{d\tau}=-\sum_x\sum_y\left[1+\log p_t\left(x,y\right)\right]\frac{d}{d\tau}p_t.\left(x,y\right) \tag{4.19}$$

Using chain rule $\tau=\log t$ and $d\tau=\frac{1}{t}dt$

$$\frac{dH_t\left(L\right)}{d\tau}=-\sum_x\sum_y\left[1+\log p_t\left(x,y\right)\right]\left(\frac{d}{dt}p_t\left(x,y\right)\right)t \tag{4.20}$$

Now as $p_t\left(x,y\right)=\frac{L(x,y;t)}{\sum\limits_x\sum\limits_y L(x,y;t)}=\frac{L_t(x,y)}{C}$

$$\frac{dp_t\left(x,y\right)}{dt}=\frac{1}{C}\frac{dL_t\left(x,y\right)}{dt} \tag{4.21}$$

$$\frac{dp_t\left(x,y\right)}{dt}=\frac{1}{C}\nabla\left(D\nabla L_t\left(x,y\right)\right) \tag{4.22}$$

$$\frac{dp_t}{dt} = \frac{1}{C}\nabla D \nabla L_t \tag{4.23}$$

The Eq. 4.20 lends itself now as

$$\frac{dH_t}{d\tau} = -t \sum_x \sum_y \left(1 + \log \frac{L_t}{C}\right) . \frac{1}{C} \nabla D \nabla L_t \tag{4.24}$$

$$\frac{dH_t}{d\tau} = -t \sum_x \sum_y (1 - \log C + \log L_t) . \frac{1}{C} \nabla D \nabla L_t \tag{4.25}$$

$$\frac{dH_t}{d\tau} = -\frac{t}{C} \sum_x \sum_y (k + \log L_t) . \nabla D \nabla L_t \tag{4.26}$$

The rate of change in the entropy for the linear isotropic diffusion case is the special case of 4.26, and this happens when the diffusion matrix D is replaced by a scalar diffusivity, say c. Spatial entropy change for linear isotropic diffusion process is given by:

$$\frac{dH_t}{d\tau} = -\frac{ct}{C} \sum_x \sum_y (k + \log L_t) . \nabla^2 L_t . \tag{4.27}$$

For both, anisotropic as well as isotropic cases, the spatial entropy change equation contains the same constant $k = 1 - \log C$.

The same tests, as were performed earlier for linear isotropic diffusion process, are conducted for *linear anisotropic diffusion* process. The test anisotropic image for this purpose consists of three curves, as shown in Fig. 4.7. At the heart of the anisotropic process is the construction of diffusion matrix D. The diffusion matrix handles steering the elliptical Gaussian to go around the structure. The geometric visualisation in the form of ellipses corresponding to point-wise diffusion matrix is displayed in Fig. 4.7, where it can be seen that they align well with the local flow of the curve. The diffusion parallel to the edges is enabled due to the large eigenvalue while avoiding the cross-over edge problems due to small eigenvalues. The linear anisotropic diffusion character is made evident by having constant eccentricity for all the ellipses across the image. The term anisotropic used here is related to changing direction of the ellipse at each pixel due to the diffusion matrix eigenvector adaptability with the given local structure. Therefore, with each iteration, the ellipse do grow without changing the eccentricity ratio and for a given diffusion time, the size of the ellipse remains constant throughout the image. Since the major axis of the ellipse is parallel to the edge of the curve, so no harm in increasing it. The minor axis of the ellipse is aligned with the width of the curve. So increasing the ellipse minor axis will eventually make the ellipse protrude outside the boundary of the curve, and the disturbed structure is obtained, and that is precisely where the diffusion should stop eventually.

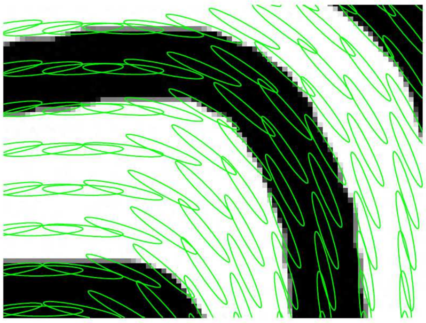
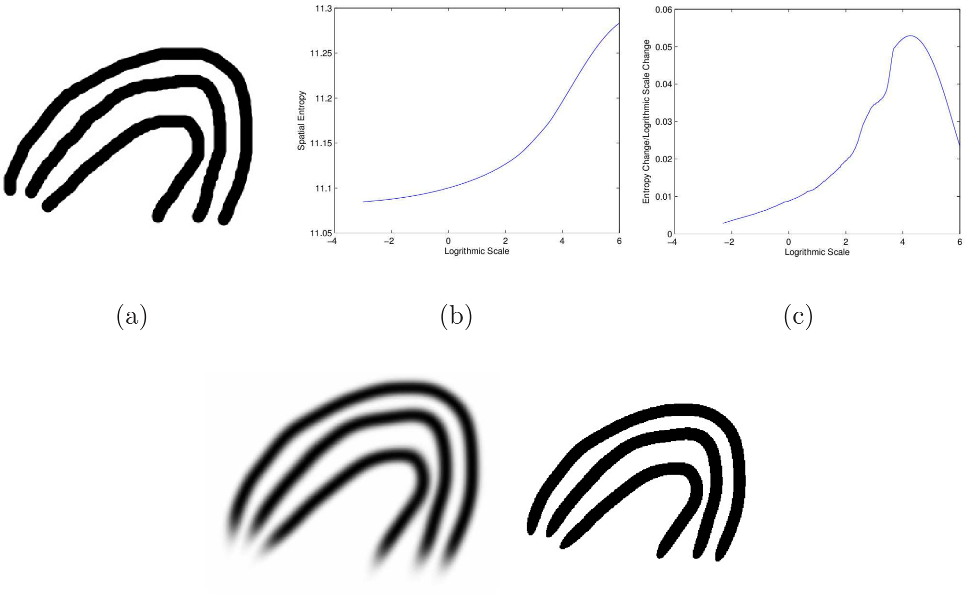


Figure 4.7: *Geometric interpretation of diffusion matrix. The figure shows part of the anisotropic curve image. The diffusion matrix associated with each pixel is depicted as ellipses on top of the image. It is observed that ellipses are steered to follow with the curve flow direction*

First, linear anisotropic diffusion process was applied to a clean curve image. The entropy and entropy change graphs as depicted in Fig. 4.8(b),(c). Both graphs are smooth and well-behaved, validating the notion that the linear anisotropic diffusion process is a lot like their isotropic counterparts. A prominent peak is located at $\tau = 4$ in the entropy change graph, representing the characteristic width of the curves present in the image. By stopping the diffusion process by that peak location, the diffused image is shown in Fig. 4.8(d). The image is largely undisturbed with small diffusion effects at the boundaries and ends of the curves. The quantitative measures, of sensitivity and specificity, for the



(a) (b) (c)

(d) (e)

Figure 4.8: *Linear anisotropic diffusion process. a) shows a flow-like test image having three black curves with a white background. The features present in the image are elongated in shape with a constant width of two pixels. b) shows the smooth spatial entropy graph resulting from diffusion process with respect to the natural scale parameter. The entropy change with natural scale change is displayed in (c), where the peak corresponds to the width of the curves. The diffused image resulting from stopping the diffusion process at the location of the peak in entropy change is shown in (d). The diffused image is converted to the binary image using Otsu optimal threshold of 0.63, as shown in (e)*

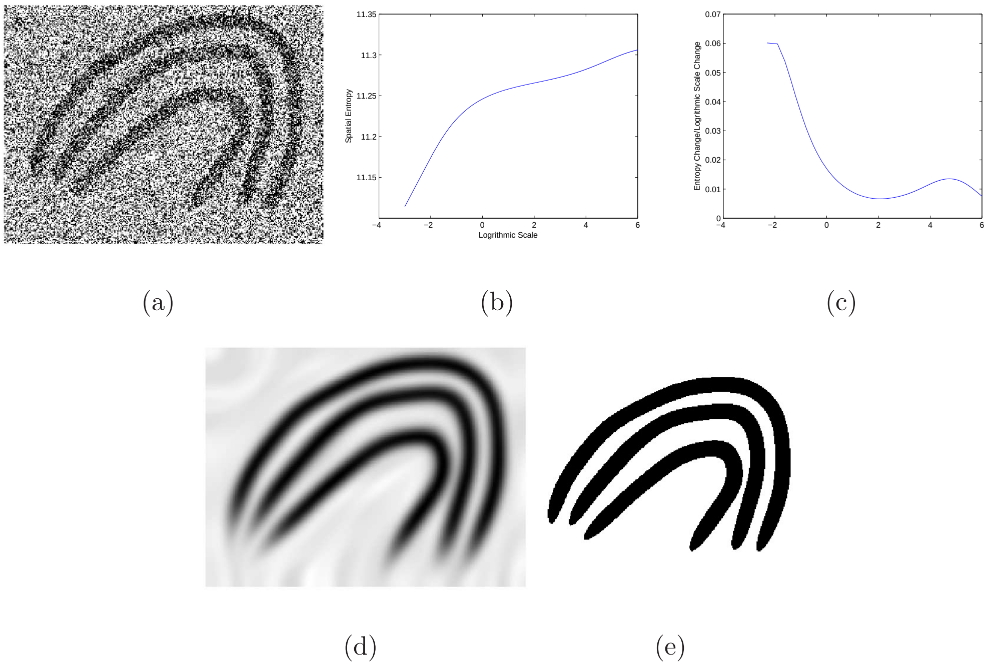
output image, are computed as 82% and 89%. The peak in entropy change graph, thus, presents itself as a suitable stopping time for the linear anisotropic diffusion process.

The experiment for linear anisotropic diffusion process was also conducted for an extremely noise situation. A Gaussian noise is added to the original curve image such that the resulting SNR is lowered to become -10dB. The noisy curve image is displayed in Fig. 4.9(a). After the completion of the linear anisotropic diffusion process, the entropy change graph is obtained as depicted in Fig. 4.9(b) and (c), respectively. It is clearly observed that the curve for entropy change is steeply coming down in the beginning and then hits a bottom. After the minimum is reached, it rises again to display a peak at the characteristic width of the curves in the noisy image. The noise can be largely curtailed by stopping the diffusion process at the peak. The diffused image stopped by the peak is shown in Fig. 4.9(d). The image do show a large smoothing of the noise with minimum disturbance to the structure of interest. Thresholding the image by Otsu method, a final binarized image is obtained, as shown in Fig. 4.9(e). The quantitative measures of sensitivity and specificity for the binarized output image are recorded as 75% and 84%.

4.7 Results and Discussion for Real Fingerprint Images

This section deals with real fingerprint images. We look into their acquisition process. Then process them for uniform background and later investigate their spatial entropy characteristic as the image evolves under linear anisotropic process. The first test that we performed is to check the anisotropic strength measure for the acquired fingerprint. The Fig. 4.10 shows results of the test. It is observed that the regular ridge/valley pattern found in the fingerprint image is largely anisotropic in nature. This justifies the employment of linear anisotropic diffusion process for smoothing these images.

The acquired fingerprint images often show important illumination variations, poor contrast in some areas and gaps in ridge/valley regions. To reduce the illumination imperfections and generate images more suitable for enhancement and minutia extraction, a preprocessing comprising the non-uniform illumination correction is applied. It occurs due to the very process of scanning a finger. The middle finger surface is thicker as compared to the surrounding region. This results in blocking the light in the middle while the outer surface is fairly highly illuminated. The fingerprint scanner registers



(a) (b) (c)

(d) (e)

Figure 4.9: *Eight linear anisotropic diffusion process for noisy image. a) shows a flow-like noisy test image having three black curves with a white background. The Gaussian noise is added to bring down the SNR of the resulting image to be -10 dB. b) shows the smooth spatial entropy graph resulting from diffusion process on the natural scale parameter. The entropy change with natural scale change is displayed in (c), where a distinct peak is still observable. The diffused image resulting from stopping the diffusion process at the location of the peak in entropy change is shown in (d). The diffused image is converted to binary image using Otsu optimal threshold of 0.55, as shown in (e).*

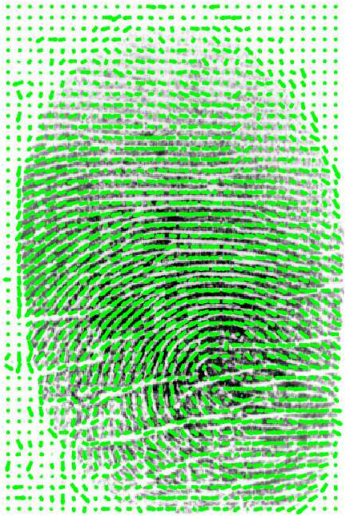


Figure 4.10: *Anisotropic strength measurement for real fingerprint images. This figure shows an acquired digital fingerprint with local anisotropy strength displayed as length of the needles on top of the image. We observe a large presence of significant anisotropy in the image*

this uneven illumination. Consequently, background variation will add bias for different regions of the same image to disturb the ridge/valley contrast. Since the ridge/valley pattern is identified and classified by its gray-level profile, this effect may worsen the performance of diffusion and disturb our spatial entropy analysis. With the purpose of removing this disturbing factor from our experimental analysis, a Homomorphic filtering approach is adopted. The process is described below.

In basic terms, homomorphic filtering assumes that an image can be represented in terms of product of illumination and reflectance. That is,

$$L(x, y) = i(x, y) \times r(x, y), \tag{4.28}$$

where $L(x, y)$ is the fingerprint image, $i(x, y)$ is the background illumination image, and $r(x, y)$ is the reflectance image [138]. Reflectance r arises due to the object itself, but the illumination image i is independent of the object, is a pure representation of lighting conditions at the time of the image capture. To compensate for the non-uniform illumination, the illumination image part has to be made constant. Illumination is assumed to be slowly varying lending itself in the low-frequency region as compared to the reflectance image that contains abrupt changes, showing a considerable high-frequency attitude.

For implementing homomorphic filtering, we first transform the multiplicative model of image formation to additive model by moving to the log domain.

$$ln(L(x,y)) = ln(i(x,y)) + ln(r(x,y)). \tag{4.29}$$

Then, a low-frequency filter is used with an appropriate cutoff to get a background illumination image $i(x,y)$ estimate. The difference $d(x,y)$ between original image $L(x,y)$ and background illumination $i(x,y)$ is calculated for every pixel,

$$d(x,y) = L(x,y) - i(x,y). \tag{4.30}$$

To this respect, literature reports illumination-correction methods based on the subtraction of the background illumination image from the original image [89], [90] and [139]. The background image is shown in Fig. 4.11. After subtraction, a greyish look image is obtained, as depicted in Fig. 4.11(b). Finally, an illuminated-corrected image is obtained by transforming linearly new image pixels into the whole range of possible grey levels [0-1] using the linear stretch. Fig. 4.11(c) shows the new image corresponding to stretched and uniformly illuminated image. The proposed illumination correction algorithm is observed to reduce background intensity variations and enhance contrast in the middle region than the original fingerprint image. The method was validated for all the images that processed in the database.

To validate the effect of the homomorphic filtering, the histogram analysis is investigated before and after homomorphic filtering stage. Histogram of an image represents the relative occurrences of the gray-level present in an image. According to [95, 138], the non-uniform illumination will modify the histogram of an image in a way that it can not be binarized by a single global threshold. For this purpose, the Otsu's method [96] is used,

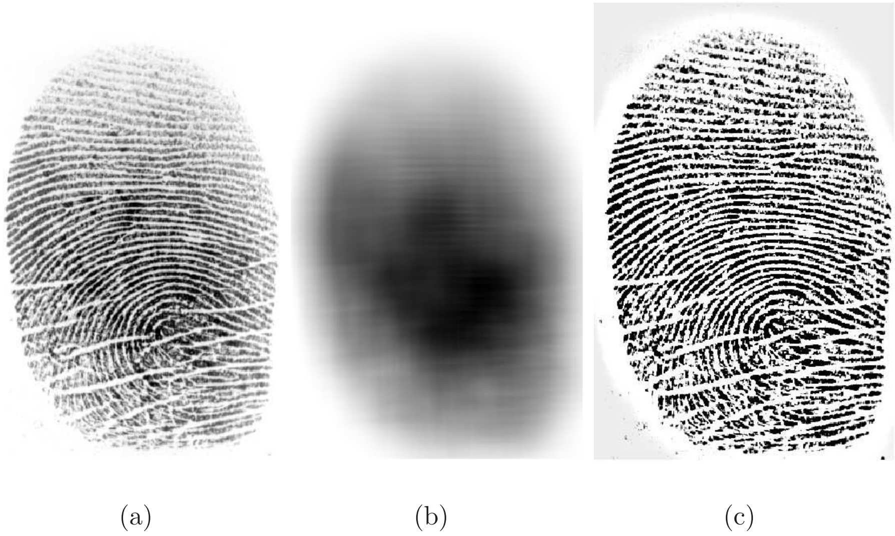


Figure 4.11: *Non-uniform illumination correction. a) shows an acquired digital fingerprint. b) depicts the estimated illumination surface, clearly showing non-uniform background lighting conditions. c) is an output result after passing the image through homomorphic filtering operation and then linearly stretched. We observe that illumination has been corrected with clear ridge/valley structure*

which chooses the threshold to minimise the intraclass variance of the background and foreground, to compute the binary threshold for the original fingerprint and that of the uniformly illuminated image. The results are displayed in Fig. 4.12. The uniformly illuminated fingerprint image is now fed to the linear anisotropic diffusion process. The image went through diffusion evolution process from a small scale $\tau = \log(t) = -3$ till $tau = \log(t) = 5$. The normal width of the ridges found to be 9, with half the width equal to 4.5. The spatial entropy graph is depicted in Fig. 4.13. We see a smooth curve with ever increasing entropy values. The entropy change graph in Fig. 4.13(b) displays a clear peak at $\tau = \log(t) = 1$, that results in $t = 2.13$. The scale value t in fingerprint images is linked to the width of the ridges as proposed in [124]. By stopping the process at $\tau = 1$, a diffused image is obtained as shown in Fig. 4.13(c). If we let the diffusion process continues for long time ($\tau = 5$), we get a mean image as shown in Fig. 4.13(d).

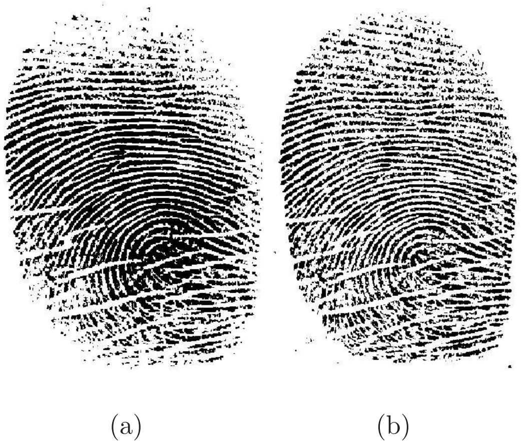
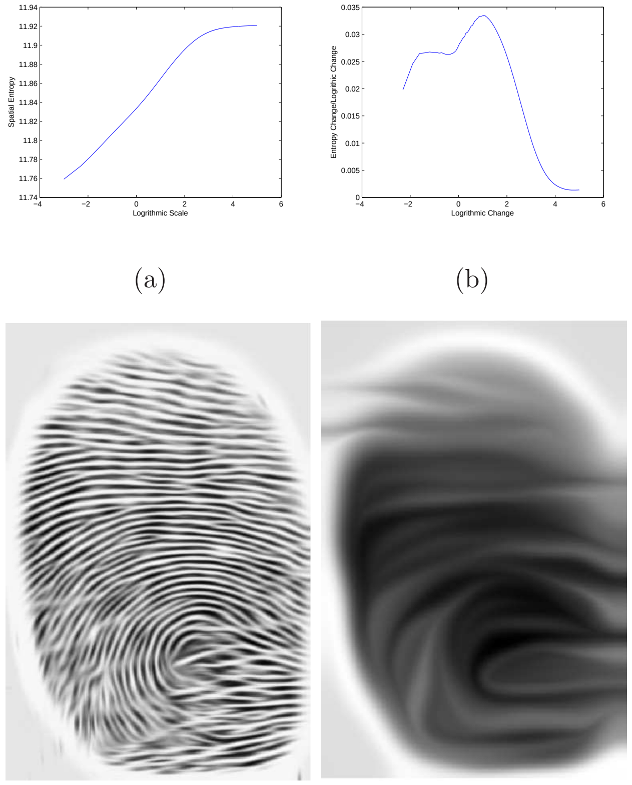


Figure 4.12: *Validation test for homomorphic filtering output. a) shows the binarization of an acquired digital fingerprint using the optimal Otsu method. b) depicts the binarization of the uniformly illuminated fingerprint with homomorphic filtering, also using the optimal Otsu method. We observe that binarization results for filtering output shows all the regions with ridge/valley structure intact*

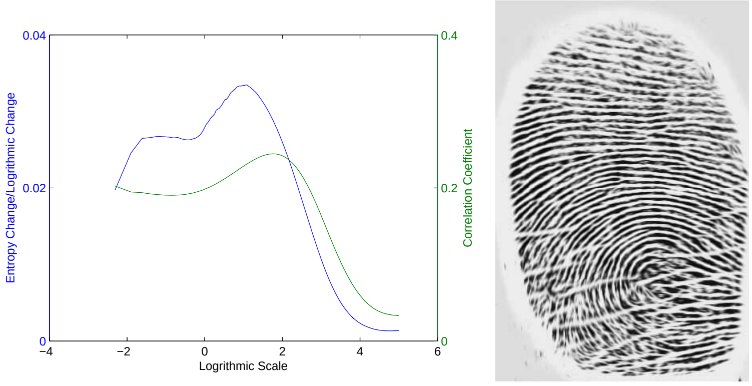
What remains to be tested is the comparison of entropy-change based stopping crite-



(a) (b)

(c) (d)

Figure 4.13: *Spatial Entropy for a real fingerprint image under linear anisotropic diffusion process. a) displays spatial entropy graph of an acquired digital fingerprint. b) depicts the entropy-change graph with one clear peak. the peak goes well with half-width of the average ridge present in fingerprint image. The diffused image obtained by peak of entropy-change is depicted in (c). While the image shown in (d) is the image we will ultimately get if we let the diffusion go on for a long enough diffusion time*



(a) (b)

Figure 4.14: *Comparison between spatial entropy-based and correlation-based stopping rule. a) displays spatial entropy change graph of an acquired digital fingerprint as a black curve and the correlation coefficient between (input noisy image - diffused image) and diffused image. b) shows stopping the diffusion process at the minimum of the correlation coefficient curve. The diffused image still shows signs of interrupted ridges*

rion with that of correlation-based method, presented in [140]. If the unknown additive noise n is uncorrelated with the unknown signal $u(t)$, it could be reasonable to minimize the covariance of the noise $u(0) - u(t)$ with the signal $u(t)$. The covariance is represented by the correlation coefficient and is given by,

$$\begin{aligned} & corr\left(u\left(0\right) - u\left(t\right), u\left(t\right)\right) \\ & = \frac{corr(u(0)-u(t), u(t))}{\sqrt{\mathrm{var}(u(0)-u(t)\,)\cdot\mathrm{var}(u(t))}} \end{aligned} \tag{4.31}$$

and choose the stopping time T so that the expression 4.31 is as small as possible.

Later on, the authors in [141] proposed to use the quality of the edges in the process of finding the optimal time to stop the diffusion process. To assess the quality of our fingerprint edge structures, the edge contrast measure is used which is defined in [142]. The edge quality index is referred to as the edge based contrast measure (EBCM). The EBCM is based on the observation that human perception mechanisms are very sensitive to contours (or edges). The larger the width of the edge pixels, the larger will be this

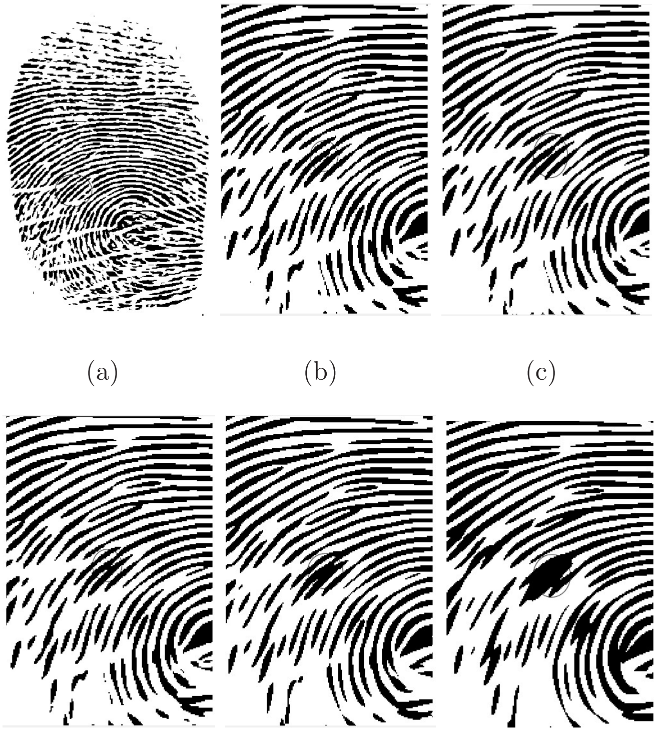
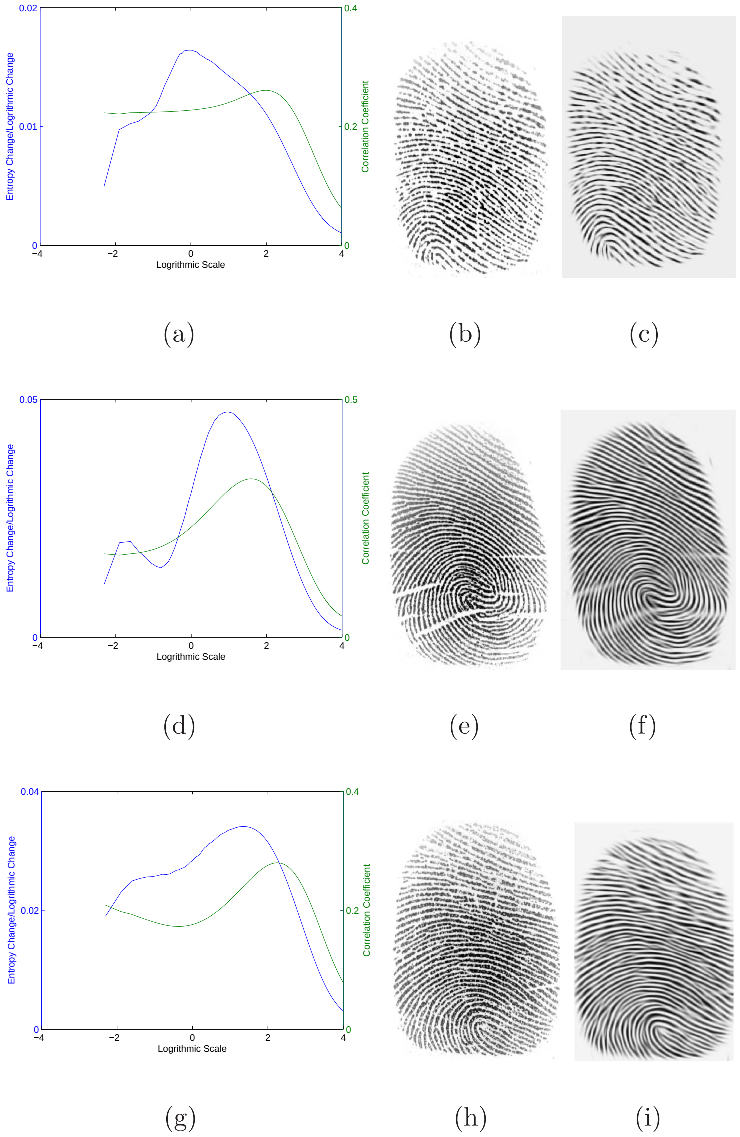


Figure 4.15: *Progression of Diffusion for A fingerprint image. Image binarized using global threshold using Otsu method with respect to various location of the entropy-change graph: a) displays image at $\tau = -1.2$, the location specified by the correlation method. Similarly, (b) at $\tau = 0.9$, (c) at $\tau = 1$, (d) at $\tau = 1.1$, (e) at $\tau = 1.4$, and (f) at tau $= 2$. We observe that as the diffusion increases, the gaps within ridges started to fill. However, after a certain limit as $\tau = 1$, the closer ridges started to get merged into one. The ellipse is drawn of the portion of the fingerprint to facilitate observation*



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 4.16: *Comparison of fake minutia point of the proposed method with correlation-based and EBSM methods. a), (d), and (g) are the graph shows the comparison of stopping time for correlation-based in green and proposed method in blue. b), (e),(h) are the fingerprint images diffused and stopped by the correlation-based optimal stopping method. c), (f), and (i) are the final optimal stopped images for the proposed method*

quality index. In our diffusion process, the edges are larger in width due to the poor image quality, so this EBCM is larger at the beginning of the diffusion process. After certain iterations, the smoothness of the noise happens, and the edges improve with less width and a lower value for the EBCM. After reaching a certain minimum, the edges again starts to widen due to over-smoothing, and the corresponding EBCM values increase. The best stopping time could be the minimum of the EPCM values, as shown in Fig. 4.14.

Image enhancement for fingerprint images is essentially to raise the contrast of ridge/valley structure, such that enhanced version is more suitable for binarization that will eventually be used for automated identification system. To perform the evaluation of the real fingerprint image after diffusion, the third party minutia extractor as provided in [143] is used. The noisy acquired images were stopped at three different time instants due to correlation method, EPCM, and the proposed entropy-change based, and the resulting three output diffused images were then compared quantitatively. Analysis of the diffused image yields a list of candidate minutiae. However, due to the use of non-optimal stopping time, there are usually a large proportion of false minutiae, i.e. points that have been incorrectly identified as minutiae. This diffusion process directly affects the binarization which creates wrong minutiae, as shown in Fig. 4.15. Therefore, the total number of candidate minutiae detected in three types of diffused images indicate the relative degree of noisiness still present in them, and will cause false minutiae. Fig. 4.16 depicts a comparison of fake minutiae of the proposed method with correlation-based and EBSM method. Table.4.1 has been generated for the six test images from the university campus students, that indicate that correlation based stopping method and EPSM-based stopping criterion had detected considerable more minutiae, indicating the immature diffusion of the noisy input image. The correlation-based stopping generated on the average 350 minutiae per image (4 times the ground truth image) while EPSM provided 210 minutiae per image (2.4 times the ground truth). The proposed entropy-change generated 145 minutiae per

Table 4.1: *A comparison: Total minutiae found by the detection algorithm enhanced by Edge Width Based, Correlation Based and Entropy-change based. The sample image are used from FVC2004 DB2_B 101_1 to 101_6*

	Edge Width Based	Correlation Based	Entropy-change based
Image1	220	367	155
Image2	200	333	135
Image3	222	370	150
Image4	208	330	140
Image5	224	380	160
Image6	206	320	130

image (1.65 time ground truth).

Another set of experiments was conducted to assess the suitability of proposed stopping criterion for some extremely low-quality fingerprint images present in the FVC2004 database to assess the ultimate strength of the proposed stopping rule. One such challenging image is displayed in Fig. 4.18(b). The fingerprint shows broken ridges, salt and pepper noise, non-uniform illumination, and on top of it a dark square patch right at the centre. The image was preprocessed first with small median filter of size 3 × 3 to tackle salt and pepper noise and was then made to go through Homomorphic filtering to eliminate to a larger extent the non-uniform background variations.

After initial treatments, the image was passed on to a linear diffusion process to join broken ridges while avoiding the mixing of ridge/valley pattern. A modified coherence enhancing diffusion (CED) as suggested earlier in linear anisotropic section proves to be of little success for diffusing low-quality fingerprints. This is due to the finding that our earlier attempts at introducing constant eigenvalues with CED process (to transform CED into a linear anisotropic process) seems to inadequate for low-quality fingerprint im-

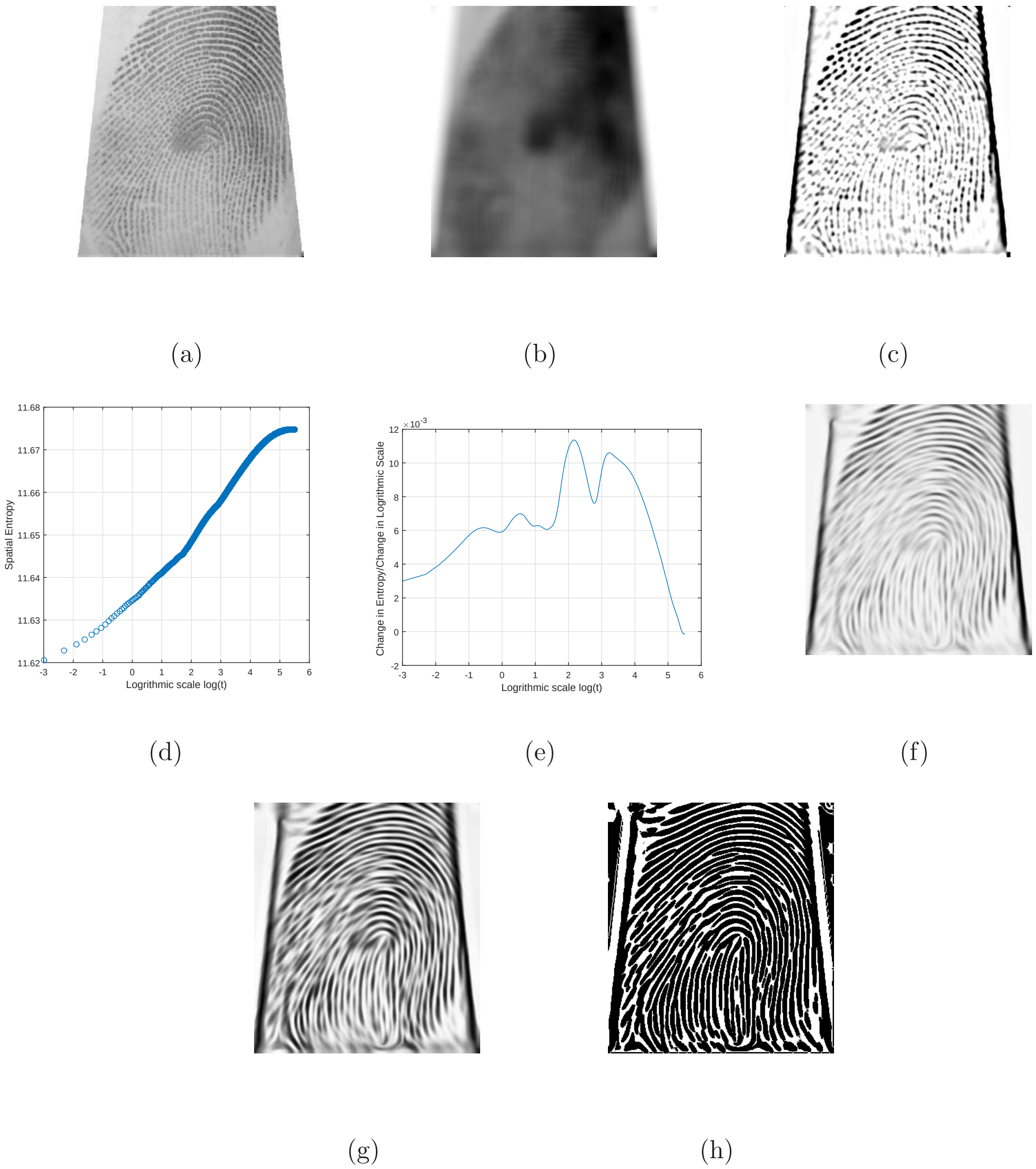
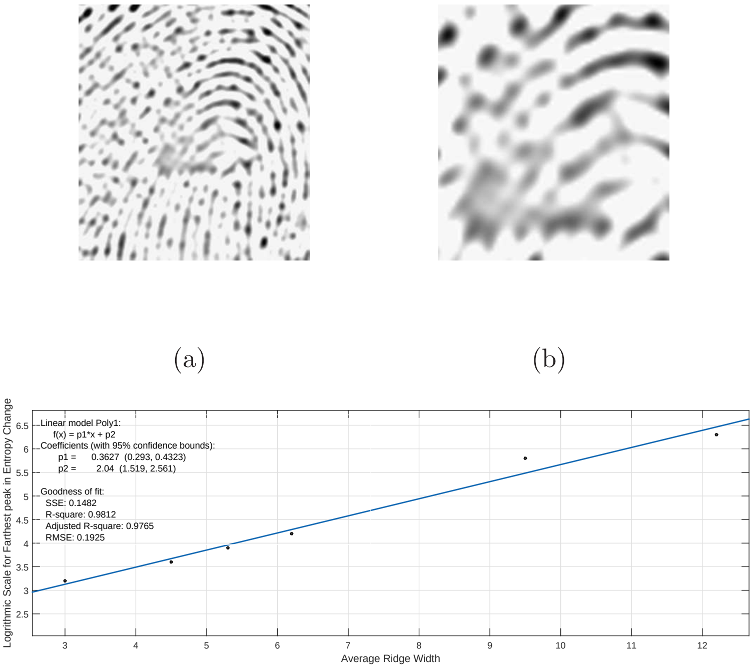


Figure 4.17: *Performance of proposed stopping rule for a low-quality image. A sample image from FVC2004 database is displayed in (a). Its non-uniform illumination image is extracted as shown in (b). The (c) depicted the uniform image. Spatial entropy points for the uniform image are plotted in (d). A piecewise smooth spline was fitted due to noisy nature of the entropy points, and subsequently, its derivative is computed as shown in (e), proving a smoothed entropy-change curve with increasing logarithmic scale. The optimally diffused image stopped at the farthest peak in entropy-change curve is displayed in (f). The contrast-adjusted image through linear stretch is shown in (g). Finally, a 9 × 9 block-based binarization was used to come up with a clean binary image as depicted in (h)*



(a) (b)

(c)

Figure 4.18: *Linear relationship between stopping point and average ridge width. The figure shows the fitting of a linear curve through some discrete points for the stopping point of the entropy-change versus logarithmic scale curve, corresponding to the furthest peak. For creating increasing large ridge widths, the center-cropped zoomed images of same dimension are being employed. The figure shows the first and the last such zoomed images. Fig. (a) Shows the first zoomed image and (b) displays the last image in zoomed series. c) gives a comparison of average ridge width with Logarithmic scale for farthest peak in Entropy change*

age diffusions. The spatial entropy curve was found to be increasing in the beginning but show a dip in spatial entropy values towards the end (large logarithmic scales). A search was conducted to look into some recent robust variant of CEDs while dealing with low-quality fingerprints. The search culminated into a new class of diffusion process that was developed specifically for low-quality challenging fingerprints. The new process deploys a pre-computed orientation field to transform the Coherence-enhancing diffusion process into that of linear oriented diffusion process [144], much more robust to the extremely noisy situations. The new process was studied with special care for its spatial entropy behaviour while smoothing low-quality fingerprints. The spatial entropy was found to be monotonically growing quantity as a function of increasing logarithmic scale. This desirable behaviour was found to be consistent across many database images that were tested here. The large part of the stable behaviour for entropy graph can be attributed to the injection of pre-computed orientation filed that was extremely helpful to steer the diffusion matrix in right direction in sensitive later stages of the diffusion process, where large scales were involved. Specifically, the linear oriented diffusion process was adopted for experimentation here with two fixed eigenvalues as $\lambda_1 = 0.01$ and $\lambda_2 = 1 - 0.01$. The diffusion matrix was constructed as before:

$$d11 = \lambda_1 \cos^2(\theta) + \lambda_2 \sin^2(\theta), \tag{4.32}$$

$$d12 = (\lambda_1 - \lambda_2) \sin(\theta) \cos(theta), \tag{4.33}$$

$$d11 = \lambda_1 \sin^2(\theta) + \lambda_2 \cos^2(\theta), \tag{4.34}$$

but with one major change that is θ is now pre-computed orientation field from the use of directional filter bank framework for the image [102]. The orientation field θ was kept constant in the whole evolution process. The diffusion process was evolved starting from scale $\tau_i = \log(t = \exp(-3))$ and reaching final scale$\tau_f = \log(t = \exp(5.5))$ (providing mean value image) with a step size of $t = \exp(-3)$. The spatial entropy was computed along the way and reported to be growing entity with steady value at the end, as depicted in Fig4.17(d). The entropy graph contains a multitude of discontinuities corresponding to a small leftover noise particles in the fingerprint after preprocessing. The curve can be smoothed by fitting a piecewise spline while caring for some real big discontinuities. To do so, a smoothing spline function was fitted to the noisy entropy curve with a coarser soothing parameter of value 0.95 on a scale of $[0, 1]$. The entropy change curve is constructed from fitted spline curve and is depicted in Fig. 4.17(e). It shows a number of peaks representing different structures dominating at different scales. There may well be some small broken parts of otherwise long ridges. The last peak at the farthest end represents the largest dominating structure that may be linked tom average ridge width of the fingerprint. By stopping the linear diffusion process at that peak $\tau = 3.2$, the diffused image is displayed in Fig. 4.17(f). The uneven image contrast can be straightforwardly improved using well-known block-based contrast enhancement scheme such as Contrast limited adaptive histogram equalisation (CLAHE) [145], to provide an evenly-contrasted image, as in Fig. 4.17(g). The contrast-adjusted image was then binarized with a block-by-block process to result in Fig. 4.17(h). The binarized result shows a clear fingerprint with ridge/valley structure largely intact (minimum mixing of nearby ridges) with greatly diminishing the intensity of noise. Most of the genuine minutia points (ridge ending and bifurcation points) are still valid and can be easily detected by the subsequent extraction process.

To quantitatively assess the performance of proposed stopping rule for image diffusion, a measure goodness index(GI), was adopted from an earlier fingerprint image enhancement[1]. This goodness index (GI) is defined as follows:

$$GI = \frac{\sum\limits_{i=1}^{r} q_i \left[p_i - a_i - b_i\right]}{\sum\limits_{r=1}^{r} q_i t_i}, \tag{4.35}$$

where, p represents the paired minutiae (between the manually extracted and machine extracted), a represents the missing minutiae, b represents the spurious minutiae and t represents the true minutiae. The measure is suppose to give a number between 0 and 1. This goodness index is applied on Fig. 4.18(b). The GI without enhancement is found to be 0.34, with enhancing using CED [127] is 0.45 and after applying the proposed method is 0.52. A larger test is performed on the 40 images of FVC2000 DB4_B (101 to 105). The averaged GI without enhancement comes out to be 0.26, with enhancing using CED [127] is 0.37 and after applying the proposed method is 0.43.

The proposed stopping rule being an iterated process can be analysed with its computation complexity profile. The stopping rule involves three nested loops. First one is the do-while loop that let the process runs till it reaches the farthest peak in the entropy change graph, and the remaining two are FOR loops that span the dimensionality of the fingerprint. Therefore, an estimate of the computational complexity associated with the proposed stopping rule can be described as a product $N \times M \times ITERATIONS$, where N and M represents the rows and columns of the fingerprint and ITERATIONS are the count of repetitions to reach the required peak. Since the peaks represent the dominating structure, which is this case is the width of the ridges, an experiment was conducted to see that linkage more explicitly. A sequence of same dimension fingerprint images was created by increasing zoom values and centre cropping the resultant image. For each of these images, an identical linear diffusion scheme with pre-computed orientation filed was run to locate the desired peak in their respective entropy-change graphs. A plot in Fig. 4.18 is shown connecting logarithmic scale at which the process stopped and the average width of the ridges in the respective zoomed images. The graph in fig shows the dots, obtained from this experiment, and were fitted with a linear curve having 95% confidence interval. The logarithmic scale, at which the diffusion process stopped, in turn, can provide the number of iterations knowing the step size involved in the diffusion process.

Thus, given dimension of the input fingerprint and an estimate of the average ridge width, a reasonable guess at the computation complexity of the proposed stopping rule can be reached.

4.8 Conclusion

In this chapter, the entropy-change for an anisotropic diffusion of a fingerprint image is investigated. A unique peak is found, associated with blurring of the dominant structure. This provides a reasonable stopping rule for the anisotropic diffusion process, whose goal is to smooth the image without disturbing the structural information. The numerical results validated the existence of the boundary between under-smooth and over-smooth regions of anisotropic diffusion.

Chapter 5

A fast and accurate iris segmentation method using an LoG filter and its zero-crossings[1]

This chapter is an adapted version of a journal article. The section headings from the journal article have been retained. Figures, equations, tables, and references have been renumbered and are in line with the thesis format. This chapter presents a hybrid approach to accurately localise the iris from an eye image. With respect to our proposed design, this chapter falls in the iris subsystem. It covers four modules of the iris subsystem, as highlighted in Fig. 5.1.

[1] Published as: Tariq M. Khan, Mohammad A. U. Khan, Donald G. Bailey and Yinan Kong,"A fast and accurate iris segmentation method using an LoG filter and its zero-crossings,"*IEEE Transactions on Information Forensics and Security* submitted

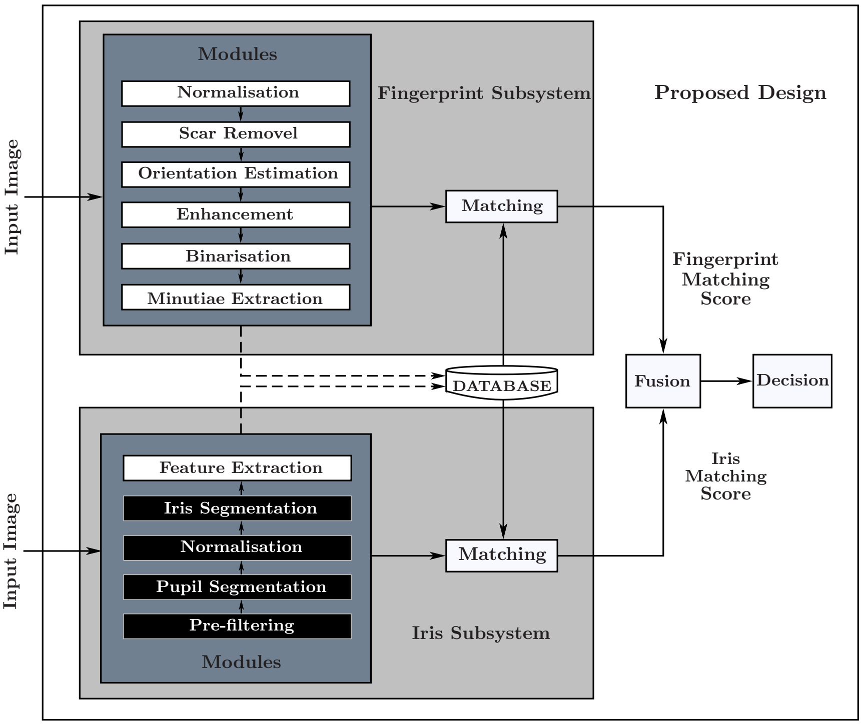


Figure 5.1: *Modules of the proposed design under study*

5.1 Abstract

This chapter presents a hybrid approach to achieve iris localization based on a Laplacian of Gaussian (LoG) filter, region growing, and zero-crossings of the LoG filter. In the proposed method, an LoG filter with region growing is used to detect the pupil region. Subsequently, zero-crossings of the LoG filter are used to accurately mark the inner and outer circular boundaries. The use of LoG based blob detection along with zero-crossings makes the inner and outer circle detection fast and robust. The proposed method has been tested on three public databases: MMU version 1.0, CASIA-IrisV1 and CASIA-IrisV3-Lamp. The experimental results demonstrate the segmentation accuracy of the proposed method. The robustness of the proposed method is also validated in the presence of noise, such as eyelashes, a reflection of the pupil, *Poisson, Gaussian, speckle and salt-and-pepper noise. The comparison with well-known methods demonstrates the superior performance of the proposed method's accuracy and speed.*

5.2 Introduction

The use of fraudulent identities is considered to be a key enabler of serious organised crime and even terrorism. Biometrics is a fast developing science that can provide a higher level of security, convenience, and efficiency to protect against identity theft than traditional password-based methods for user authentication. Humans have many biometric traits such as a face, hand geometry, fingerprint, voice and iris [146, 146, 147], that can be used for identity verification. However, iris recognition is found to be accurate and one of the more reliable methods due to its high degree of uniqueness and randomness, even between identical twins, and remains constantly stable throughout an adult's life [148, 149].

The iris is a nearly circular shaped region between the sclera and the pupil. It is an internal organ that is well protected from the environment and can be seen from outside the body [150]. It consists of many features like furrows, freckles, stripes, coronas, ligaments, arching, zigzag collarette, ridges, rings, and crypts [151–153]. These features are statistically stable, unique, and are randomly distributed in the human iris region [154, 155]. These properties make the iris a secure and reliable source of personal identification [156].

Generally, the essential steps within an iris recognition system are: eye image acquisition, segmentation of the inner and outer boundary of iris, extraction of unique features, feature matching and finally the recognition of a person [148], as shown in Fig. 5.2. Of these steps, the segmentation of the inner and outer boundary of iris plays a vital role towards system accuracy. Iris segmentation localises two different boundaries. First, it segments out the pupil's outer boundary, known as the pupillary or inner boundary of the iris, and then the outer or limbic boundary of the iris. Iris segmentation is a computationally intensive task in iris recognition [148].

Although state-of-the-art methods [157–160] are very effective for iris recognition, their performance is greatly affected by iris segmentation. The reasons are as follows:

- Iris segmentation defines the contents of the features that are subsequently processed, by normalisation, feature extraction, and matching. Thus the accuracy of iris recognition is directly related to iris segmentation. Inaccurate iris segmentation is reported to be the main cause of failure in iris recognition systems [153].
- Processing speed is a bottleneck in practical applications, while in iris recognition the most time-consuming module is iris segmentation [157, 161].

There are several challenges in practical iris segmentation. Some of them are the image acquisition angle, pupil dilation, occlusion, focus, and image clarity. Pupillary and limbic boundaries are nonmuscular, which can lead to inaccuracy if fitted with simple shape assumptions. All of these challenges makes the iris segmentation process difficult. Therefore a fast, accurate and robust iris segmentation algorithm is highly desirable.

In the literature, different techniques are proposed for fast and accurate iris segmentation. These can be divided into two broad categories: Shape-based detectors [152, 157] and intensity-based thresholding [146, 148, 149]. Generally, shape-based detection gives better accuracy but is slower in processing. On the other hand, intensity-based threshold methods are fast, but they are less accurate than the shape-based detectors.

In this chapter, we investigate the combination of a fast shape-based detector with an intensity-based threshold to accurately segment an iris. A Laplacian of Gaussian (LoG) filter is used as a shape detector along with region growing and an intensity-based threshold is efficiently used to locate the true pupil region. Then the zero-crossings of the LoG filter are used to mark the true inner and outer boundaries of the iris. The combination

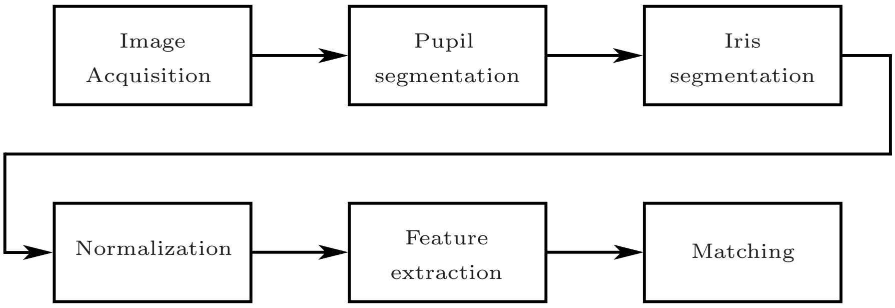


Figure 5.2: *Block diagram of an iris recognition system*

of these filters not only gives better accuracy than intensity-based threshold methods but also gives much better processing speed than shape-based detectors.   The rest of this chapter is organized as follows.Background and related work is presented in Section 5.3. Section 5.4 give details of proposed method. Experimental results are presented in Section 5.5. Computational cost and limitation are discussed in section 5.6 followed by the conclusion of this chapter in Section 5.7.

5.3 Background and related work

In iris segmentation, the first step is to find the pupil centre and then to fit a circle to localise the pupil boundaries [148]. The centre of the iris lies within the pupil, but the two circles are not concentric. An accurate localization of the pupil reduces the search space for the centre of the iris. Duagman proposed an edge-based detector using an integral differential operator (IDO) [162]. The operator searches over the image domain (x, y) for the maximum in the blurred partial derivatives with respect to increasing radius r, of the normalised contour integral of $I(x, y)$. The complete operator behaves as a circular edge detector, blurred at a scale set by σ, searching iteratively for the maximal contour integral. A circular Hough transform is used to detect the inner and outer boundaries of the iris. Three parameters (x_0, y_0, r) are used to define each circle, where (x_0, y_0) is the center and r is the radius of the circle. Similarly, a mixture of the gradient edge-map and the circular Hough transform was used by Wildes [152] to pinpoint the iris boundaries.

The literature provides evidence that histogram and threshold based techniques are faster than using the Hough transform [163]. In histogram-based techniques, thresholding is used for locating the pupil considering it as the darkest region in an eye image [146]. For pupil detection, Zhang [164] first divided the eye image into small rectangular blocks of fixed size and then found the average intensity value of each block. Dey [165] used down-sampling on the input image before pupil and iris localization. To find all the connected components, first, contrast scratching is applied to the down-sampled image followed by the image binarization.

Ibrahim *et al.* [148] used histogram-based and standard deviation based adaptive thresholding to localise the pupillary boundary. A range of grey levels that has the highest probability of the pupil is found by moving a circular window. The window that contains a grey level peak with a minimum standard deviation of x- and y- coordinates is selected as the pupil region. This technique may fail for images containing other objects such as eyelashes, eyebrows, hair, and possibly the black-frame of glasses. Similarly, Khan *et al.* [146] used histogram-based thresholding along with eccentricity to extract the pupillary boundary. Their technique lacks the ability to handle multiple objects in a given binary image. Use of eccentricity on its own could be misleading if a small object (other than the pupil) has smaller eccentricity in an image. To overcome these issues, Jan *et al.* [166] proposed a technique in which a common eye position is generalised by using integral image projection functions. Then it was binarized by using an adaptive threshold via a histogram bisection method. It is controlled by a parameter vector that is recorded by eccentricity, image statistics, and the two-dimensional (2D) object geometry. Then the limbic boundary is identified in the horizontal direction by using gradient information. Lastly, the iris boundaries are localised by using a mixture of radial gradients. This method gives effective results for iris databases but with certain limitations. First, the dark locations (e.g. pupil, etc.) were highlighted because of the property of integral image projection. If other low-intensity regions (e.g., eyebrows, eyelashes, hair, and the frame of glasses) block the dark regions (e.g. pupil, etc.), then the integral image projection may fail to highlight it because of the coordination failure of the horizontal and vertical projection functions. Secondly, the useful combination of area and eccentricity of 2D objects are used to locate the pupil in a binary image but, as was discussed earlier, the results are better for a perfectly round pupil, but method may not provide the desired results for a distorted pupil, for example, CASIA- IrisV3-Interval and CASIA-IrisV4-Thousand iris databases.

Although in literature, many different iris segmentation methods have been proposed, many of them are either computationally or present relatively high or unacceptable segmentation error rates. The iris image segmentation algorithm proposed in this chapter consists of two major modules, namely pupil detection, and limbic boundary localization including eyelid/eyelash detection. The implemented algorithms avoid unnecessary processing over image regions that do not contain relevant information for iris image segmentation, and consequently iris recognition.

5.4 Proposed method

We have proposed a two-stage method for iris localization. In stage 1, the pupil (iris inner boundary) is localised, an important step in iris segmentation for two reasons:

1. In the iris images, if the pupil is wrongly localised, then there are often errors in detecting the limbic boundary, as the iris's outer boundary localization methods use the pupil circle parameters as inputs [146, 166–168].

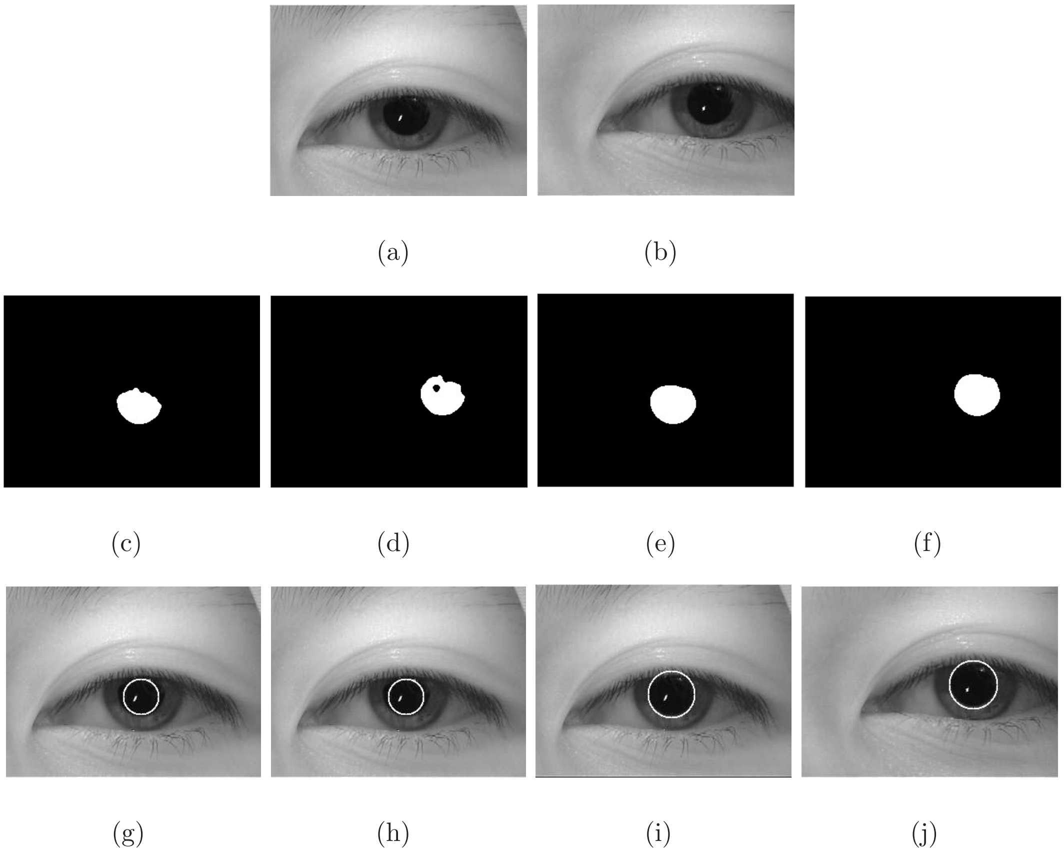
2. The time consumed in pupil localization is much more than for the limbic boundary localization because the whole eye (iris) image is processed in pupil localization, whereas a sub-image can be processed for the limbic boundary localization [167,168].

In stage 2, the limbic boundary is localised. The details of the proposed method are given in the next subsections.

5.4.1 Pupillary boundary localization

There are many methods reported in the literature for detecting circular objects in an image. Intensity-based thresholding can be treated as blob detection, in which the pupil is treated as a circular black region on a bright background. Such techniques assume that the gray-level intensity of the pupil in an eye image is less than for any other region (e.g., iris, sclera, and skin parts). Some researchers use this assumption to localise coarse pixels in the pupil region using a histogram or threshold and then use gradient-based techniques to segment the boundaries of the iris [146,148,169]. These techniques may not work for an eye image containing other low-intensity regions. When the pupil is shaded by eyelashes, thresholding fails to locate the true centre and radius of the pupil region. For example, Fig. 5.3(e) and 5.3(f) clearly show that [148] fails to get a proper pupil region because of noise. To overcome this, [170] proposed some solutions that make the implementation computationally expensive. Another disadvantage of [170] is the use of iterative processes that limit its real-time implementation.

To provide a robust, computationally light and non-iterative solution for finding the disc, we adopted a scheme that combines both a shape-based detector and an intensity-based threshold. We treated the pupil as a blob and tune a Laplacian of Gaussian (LoG) [171] operator to detect the edges or blobs at a particular scale. It is based on filtering an image with a Gaussian of particular standard deviation σ, also known as the scale factor.



(a) (b)

(c) (d) (e) (f)

(g) (h) (i) (j)

Figure 5.3: *Comparison of [146, 148] with proposed method. (a) and (b) are two noisy images from the MMU v1 database. (c) and (d) show the pupil extracted by [148]. (e) and (f) show the pupil extracted by the proposed method. (g) and (h) show the normalised pupil using [148]. (i) and (j) show the normalised pupil by the proposed method.*

The 2-D LoG function with Gaussian standard deviation σ has the form:

$$h\left(x, y; \sigma\right) = -\frac{1}{\pi\sigma^4}\left[1 - \frac{x^2+y^2}{2\sigma^2}\right] e^{-\frac{\left(x^2+y^2\right)}{2\sigma^2}} \tag{5.1}$$

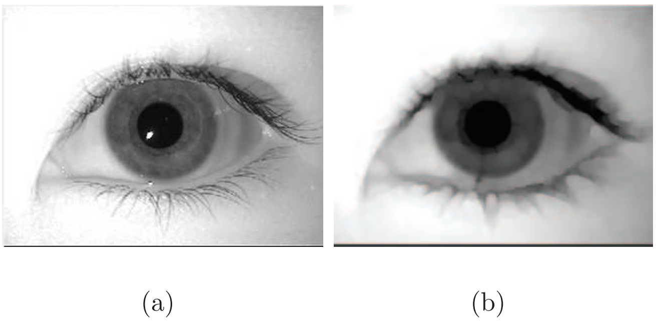
The scale normalised version of the LoG filtered defined in Eq. 5.1 is modified as [171]:

$$\begin{aligned} h_{SN}\left(x, y; \sigma\right) &= \sigma^2 \times h\left(x, y; \sigma\right) \\ &= -\tfrac{1}{\pi\sigma^2}\left[1 - \tfrac{x^2+y^2}{2\sigma^2}\right] e^{-\frac{\left(x^2+y^2\right)}{2\sigma^2}} \end{aligned} \tag{5.2}$$

The selection of σ depends on the blob size. A Gaussian has several advantages that facilitate blob detection. First, the Gaussian is separable; that makes it computationally efficient. Second, the Gaussian is smooth and localised in both spatial and frequency domains. This smoothing provides a good compromise in terms of suppressing false edges. The LoG acts as a bandpass filter because of its differential and smoothing behaviour. As the Laplacian is a linear operator, Gaussian filtering followed by differential is the same as filtering with a Laplacian of Gaussian.

To provide a robust and computationally light solution for finding the disc, we adopt the following strategy. An iris image $I(x, y)$ from the MMU-V1 database is taken as sample, as shown in Fig. 5.4(a). The image contains a black disc in the centre of the eye with small clumps of undesirable foreground pixels, e.g. salt noise. Though a median-based operator can be used to tackle such high-frequency noise, it can be computationally expensive. Therefore, it can be replaced with a morphological opening that does a similar job with fewer resources. The smoothed image is shown in Fig. 5.4(b).

The next processing step is to apply the LoG filter. The LoG filter is an anisotropic filter that has been used effectively in the past to detect blobs [172]. For iris segmentation, the pupil circular region can be taken as a black blob on a white background. To facilitate the LoG application, the iris image is first converted to a tri-level image. In the tri-level image, the pupil is represented by black intensity, the outer circular region around the pupil with white intensity, and the rest of the image as grey intensity. This particular

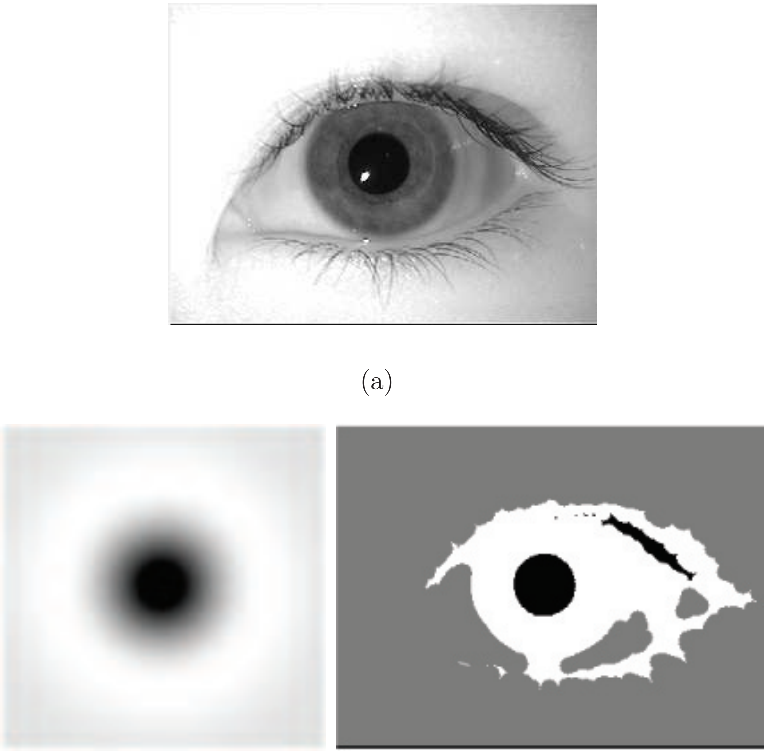


(a) (b)

Figure 5.4: *Preprocessing: (a) Sample input image. (b) Image filtered with morphological* opening *with a disc structuring element of radius 5 pixels.*

swapping of the white and grey level regions facilitates the application of the LoG filter, by giving a larger contrast between the inner and outer circles as shown in Fig. 5.5(b).

Conversion to a tri-level image requires two thresholds: T1 and T2. The first threshold T1 is chosen as a level below which we have a large confidence of picking the pupil. The second threshold T2 is an intensity level beyond which we have a high confidence to get the rest of the image. The result is therefore given by



(a)

(b)

(c)

Figure 5.5: *Median Filtering: (a) Sample input image. (b) 2D representation of a LoG filter. (c) image (a) converted to a tri-level image*

$$Tri_{img} = \begin{cases} 0 & if \;\; I_{smooth} < T1 \\ 1 & if \;\; T1 < I_{smooth} < T2 \\ 0.5 & if \;\;\; I_{smooth} > T2 \end{cases} \tag{5.3}$$

The problem of choosing the thresholds has been facilitated by preprocessing the image with morphological operator opening. This greatly reduces the stress of finding accurate thresholds T_1 and T_2. For the sake of experiments, we choose T1=0.2 and T2=0.5 for a scaled IRIS image in the [0-1] range. The tri-level intensity converted image is displayed in Fig. 5.5(c). The figure shows the effectiveness of tri-level scaling.

A LoG filter given in Eq. 5.1 is applied on Tri_{img} as

$$I_{LoG} = Tri_{img} * h\left(x, y, \sigma\right), \tag{5.4}$$

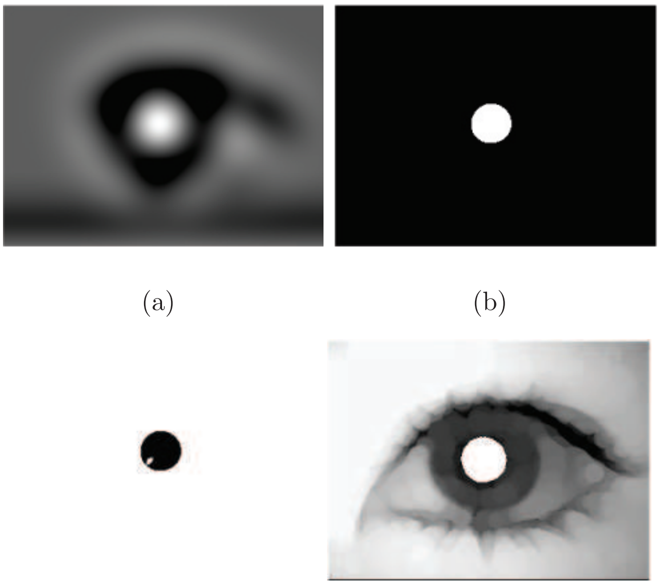
where $\sigma = R$ is the average pupil radius of a particular database. By applying this LoG with such a coarse scale, the output images I_{LoG} possesses strong contours due to the heavy smoothing, as shown in Fig. 5.6(a). The LoG filtering provides the maximum response in the pupil region. Now a mask is created that corresponds to the pixel of I_{LoG} with maximum response as

$$Ig_{mask} = \begin{cases} 1, & if \quad I_{LoG} > \lambda_a \\ 0, & otherwise \end{cases}. \tag{5.5}$$

where λ_a is set to some value above the mid-grey level, such as 0.6 for I_{LoG} scaled in [0-1] range. Fig. 5.6(b) shows the mask image that is used for generating the first seed image. This mask image Ig_{mask} is then multiplied pixel-wise by the I_{smooth} image to get the first seed image as

$$Ig_{seed} = I_{smooth}. * Ig_{mask} \tag{5.6}$$

Fig. 5.6(c) shows that this seed image contains the pupil. The next task is to choose an appropriate seed point among this seed image using centroid of the seed image. This point is used as a seed point for the region growing method [94]. Region growing is a segmentation strategy that starts with a pool of only one initial seed point and then adds more pixels to the pool that are 8-connected neighbours with similar intensity to



(a) (b)

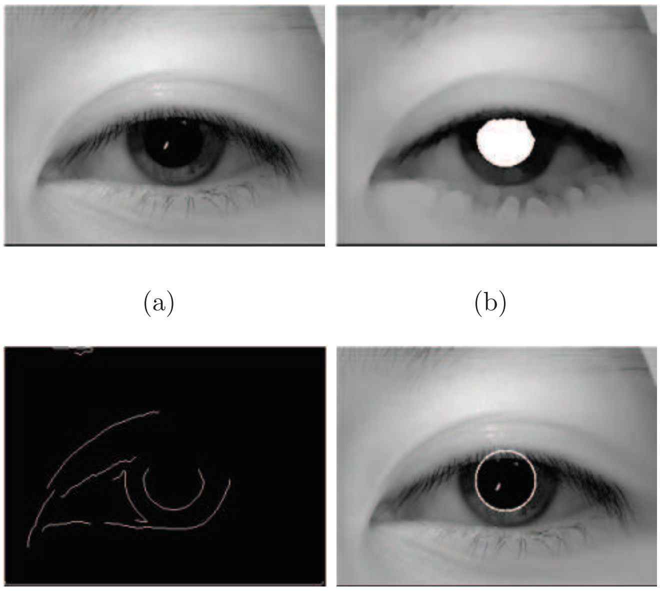
(c) (d)

Figure 5.6: *LoG filtering: (a) LoG filtered image. (b) Mask created by thresholding the LoG filtered image. (c) seed image created from threshold image. (d) Segmented pupil by proposed method*

that of the earlier seed points. The tri-level image loses important texture information, therefore, the LoG-filtered image is used only for finding seed point where the region growing is performed on the I_{smooth}. The initial seed point is grown to a target pool of pixels using a similarity measure where the intensity of the seed point is compared with 8 neighbours using a 5 percent rule. Fig. 5.6(d) shows a segmented pupil using the proposed method. By using LoG filtering along with region growing, most of the problems reported in [146, 148, 169] for pupil segmentation are addressed.

From Fig. 5.3(e) and 5.3(f) it can be observed that the pupil region reconstructed by the proposed method is an improvement on [146, 148, 169]. But in extremely noisy conditions, this still can give some error in finding the accurate centre and radius of the pupil, Fig. 5.7(b) provides evidence of this situation. Therefore, to further strengthen the proposed method, zero-crossings of the LoG-filtered image are also obtained. The zero-crossings give the true edges of the pupillary boundary. This can help the proposed method to find more accurately the centre and radius of the pupil.

The behaviour of the LoG zero-crossings edge detector is largely governed by the standard deviation of the Gaussian used in the LoG filter. It is common to see several spurious edges detected away from any obvious edges. To deal with spurious edges the first order differential information of the image is required. This information will provide The gradient magnitude at the zero crossing of the LoG-filtered image. Discarding the zero crossings with a magnitude lower than a threshold will retain only the stronger edges. To implement the zero-crossings of the LoG filter, the morphological filtered image I_{smooth} is convolved with a LoG filter with $\sigma = 2$, having filter size $n = \lfloor (3\sigma) \times 2 + 1 \rfloor$. Then the zero-crossings of this LoG filtered image are obtained with a threshold of $\lambda_c = 0.15$ for the MMU v1 database. Fig. 5.8(b) shows the result of zero-crossings. The images of the database are preprocessed with Gaussian smoothing before calculating their first order differential strength measure. A single value of λ_c is appropriate to suppress spurious edges related to insignificant zero crossing points. The value of λ_c is chosen after performing several experiments on the MMU-v1 database. It is observed that if the value of λ_c is increased then it gives fewer edges, and in some cases it affects the pupil and iris boundary edges. On the other hand, if a smaller value is chosen then noise in the zero-crossing image increases and makes it harder to locate the pupil. From Fig. 5.8(b) it can be seen that the zero-crossings image has unwanted noise that needs to be cleaned up. Connected components of the zero-crossing image are found and those with fewer fifty pixels are removed. This removes small unwanted regions without affecting the pupil and iris boundaries, as shown in Fig. 5.8(c). The seed point calculated for the region growing is used as a reference point to extract the true circle from the zero-crossings. From this reference point, the boundary of the pupil is scanned radially in the zero crossing image. From Fig. 5.7(c) it can be seen that the circle is broken because of eyelashes and eyelids. The zero-crossings help to find the true broken region. By using interpolation, the remaining part of the circle can easily be predicted. Fig. 5.7(d) clearly shows that, by using zero-crossings, the proposed method can extract the true centre and radius of the pupil from a noisy iris image.



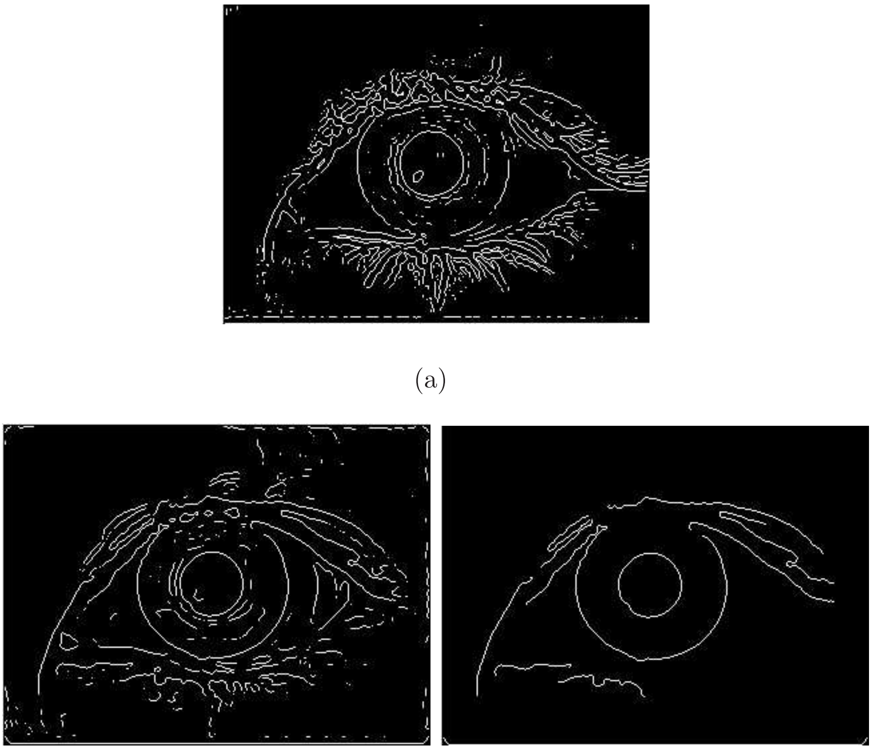
(a) (b)

(c) (d)

Figure 5.7: *LoG filtering on noisy image: (a) Sample noisy image of MMU v1 database. (b) Detected pupil by proposed method. (c) Zero-crossing of LoG filtered image. (d) Result of proposed zero-crossings on (a).*

5.4.2 Limbic boundary localization

Limbic boundary extraction is also difficult for the following reasons: first, the eyelids and eyelashes may partially occlude the iris outer boundary. Second, the contrast between the iris and sclera regions is usually low. Lastly, as the pupil always exists within the iris region, the pupil and limbic boundaries could be assumed as two nonconcentric circles; however this assumption is not always true [163]. Basit [169] picked a horizontal line



(a)

(b)

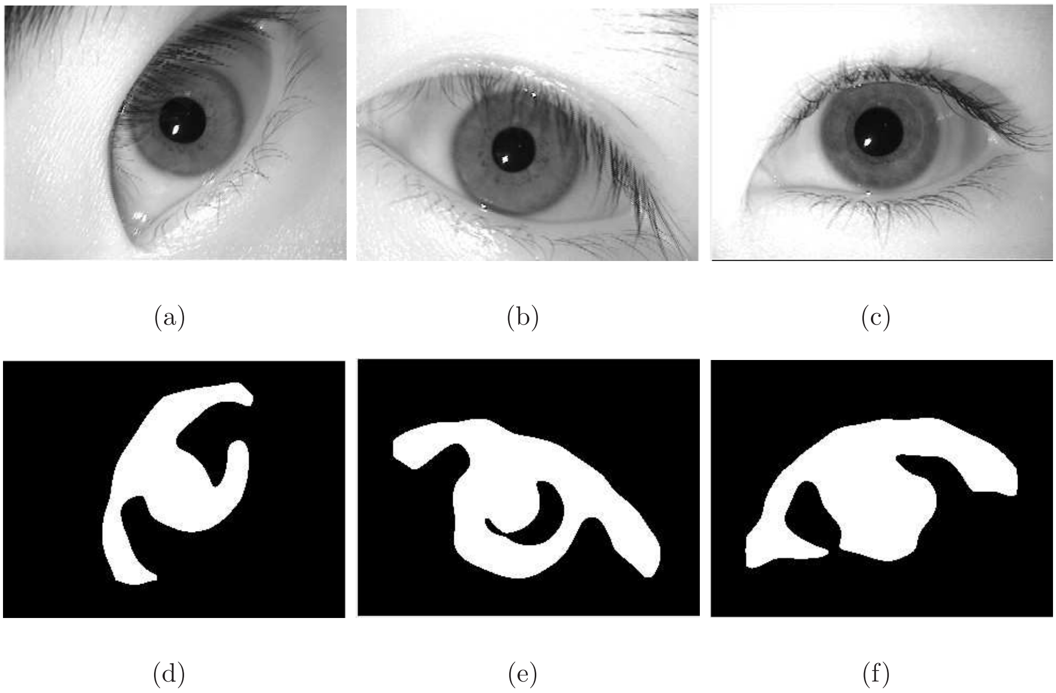
(c)

Figure 5.8: *LoG zero crossing: (a) Zero crossing without morphological filtering. (b) Zero crossing with morphological filtering. (c) Zero crossing after cleaning with area property.*

from the centre of the pupil and used a gradient to find the edges. Based on these edges, the pupil centre and the radius are calculated. [146, 148, 173] used a similar technique in which two secure regions are defined. Then their gradients are computed, followed by excluding wrong boundary points using a distance error transform. One disadvantage of these techniques is that they bias their localization of the limbic boundary in the horizontal direction. Also, these techniques may not work for an eye image having low-intensity regions.

To resolve these issues, we adopted a scheme that first finds the true orientation of the eye in the image. First, I_{smooth} is filtered using a LoG filter of $\sigma = R$. This is then thresholded to select the top 70% of positive values. The largest connected component that overlaps the detected pupil is then selected, as shown in Fig. 5.9. The detected eye image is approximately ellipse shaped with major axis almost double the minor axis. The orientation of the eye is found from the orientation of the major axis of the ellipse. There are several advantages to finding the true orientation of the eye. First, if the major axis is treated as the x-axis [146,148,173] then the chances of getting noise in the stable zone will be very low. This will certainly increase the accuracy of these algorithms. It also makes it easier to find the area affected by the eyelashes and eyelid (occlusion zone). Second, the orientation also facilitates the iris normalisation and matching process.

To extract the outer boundary of the iris, the eye image is divided into four regions: left stable zone, right stable zone, upper occlusion zone, and lower occlusion zone, as shown in Fig. 5.10(a). Usually, the upper occlusion zone affects the iris region more than lower occlusion because of the eyelashes and eyelid. The detection of these zones not only plays a vital role in limbic boundary localization but also in iris normalisation. In this chapter, this issue is addressed by first detecting the true outer boundary of the iris in the secure regions and then in the occlusion zones. To accurately detect the outer boundary of the iris, the cleaned zero-crossing image is used. Both secured zones are radially scanned in a similar manner as in [146, 173]. Fig. 5.11 shows the iris outer boundary extracted from the secured region. The average distance of the outer boundary in the stable zones from the centre of the pupil is calculated and is designated as the iris radius I_{radius}. Using this radius, the search is extended into both occlusion zones. Any discontinuity shows that the region is affected by eyelids/eyelashes, as shown in Fig. 5.11(b). Therefore, using these boundary points, the affected region in the iris can easily be marked, as shown in Fig. 5.10(c). Finally, the disconnected outer circle is interpolated to give the true centre and radius of the limbic boundary, as shown in Fig. 5.11(c).



(a) (b) (c)

(d) (e) (f)

Figure 5.9: *Binary Mask used to find out the true orientation of eye image. (a), (b), and (c) are three sample eye images of MMU database. (d), (e), and (f) are the binary masks of (a), (b), and (c) obtain by using LoG filtering.*

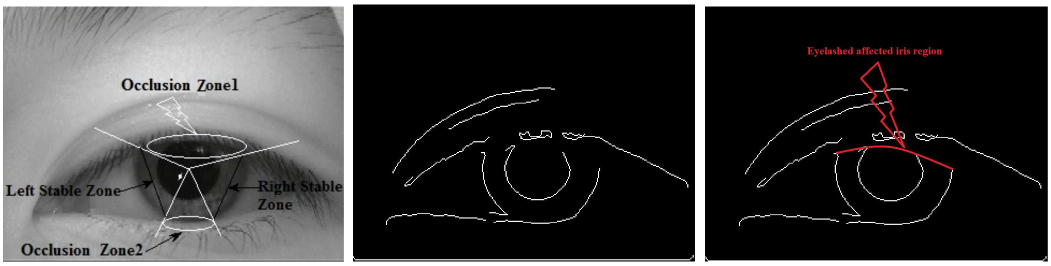
5.5 Experimental results

The validity of the proposed method is evaluated on three public databases, namely: MMU version 1.0 database [174], CASIA-IrisV1 database [175] and CASIA-IrisV3-Lamp database [175]. The accuracy rate is used to measure the performance of the proposed method. The accuracy rate (A_r) is based on the accuracy error A_e, which is defined as

$$A_e = \frac{|N_a - N_{\text{det}}|}{N_{total}} \times 100, \tag{5.7}$$

where N_{det} and N_a are the numbers of detected and actual iris pixels, respectively. The actual iris pixels are calculated manually as suggested in [148]. If A_e is less than 10%, then the detected iris is considered to be the true iris. A_r is defined as

$$A_r = \frac{N_{success}}{N_{total}} \times 100, \tag{5.8}$$



(a) (b) (c)

Figure 5.10: *(a) Iris image divided into different zones, (b) Zero-crossings of LoG filtered image. (c) Detected eyelashes in iris and pupil region.*

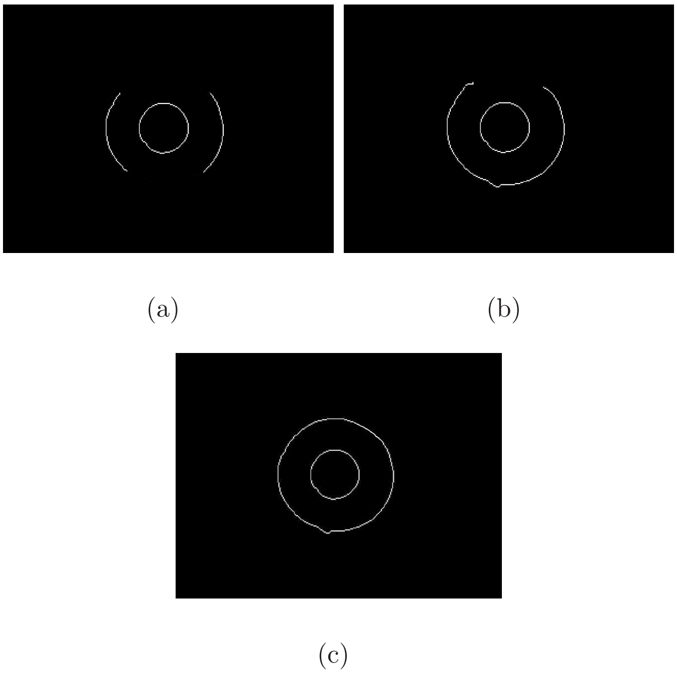
where N_{total} is the total number of images in the database and $N_{success}$ is the total number of eye images in which the iris has been successfully localized. The following sections describe the details of the experimental results.

5.5.1 Experimental setup 1

In this experiment, results are collected using data from the MMU version 1.0 database. This database contains 450 images of 45 subjects, i.e., 10 images per subject. The resolution of each image is 320×240 pixels. The proposed method has been tested on the whole database. An accuracy rate of 100% is achieved on both the inner and the outer boundary of the iris. Fig. 5.12 shows the results of the proposed method on some of the randomly selected images from this. Table 5.1 compares the accuracy of the proposed method with several existing methods on the MMU version 1.0 database.

5.5.2 Experimental setup 2

The second experiment is performed on the CASIA Ver 1.0 iris database. This database contains 756 eye images of 108 subjects, 7 images per individual. The resolution of each image is 320×280 pixels. Using the mixture of LoG filtering and zero-crossings of

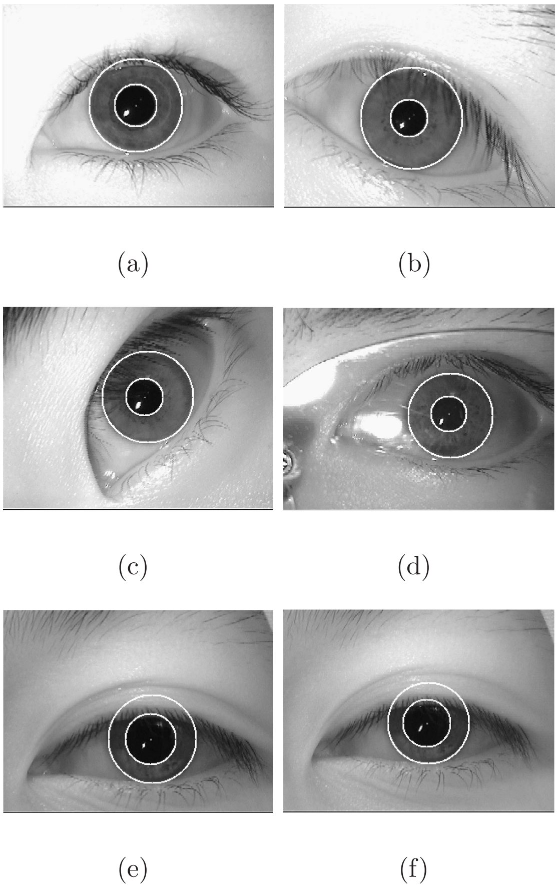


(a) (b) (c)

Figure 5.11: *(a) Extracted secure region. (b) Extended search of boundaries in turbulence zones. (c) Interpolated iris.*

Table 5.1: *Comparison of some recent segmentation algorithms applied to the MMU database (Results are taken from [165]).*

Method	Accuracy
Masek [176]	83.9%
Daugman [162]	85.6%
Ma et. al. [153]	91%
Daugman [154]	98.2%
Somanth et. al. [165]	98.4%
Proposed	100%



(a) (b)

(c) (d)

(e) (f)

Figure 5.12: *Results of proposed method on some noisy images of the MMU version 1 database.*

LoG filter we achieve 100% accuracy on this database for both the inner and the outer boundary of the iris. Fig. 5.13 shows the results of the proposed method on some of the randomly selected images from CASIA-IrisV1. Table 5.2 compares the proposed method with existing methods using CASIA-IrisV1.

5.5.3 Experimental setup 3

In this experiment, results are collected using data from the CASIA-IrisV3-Lamp iris database. This database contains 16440 images of 441 subjects. Each subject contributed 40 images, 20 images of the each eye with a resolution of 640×480 pixels. The proposed

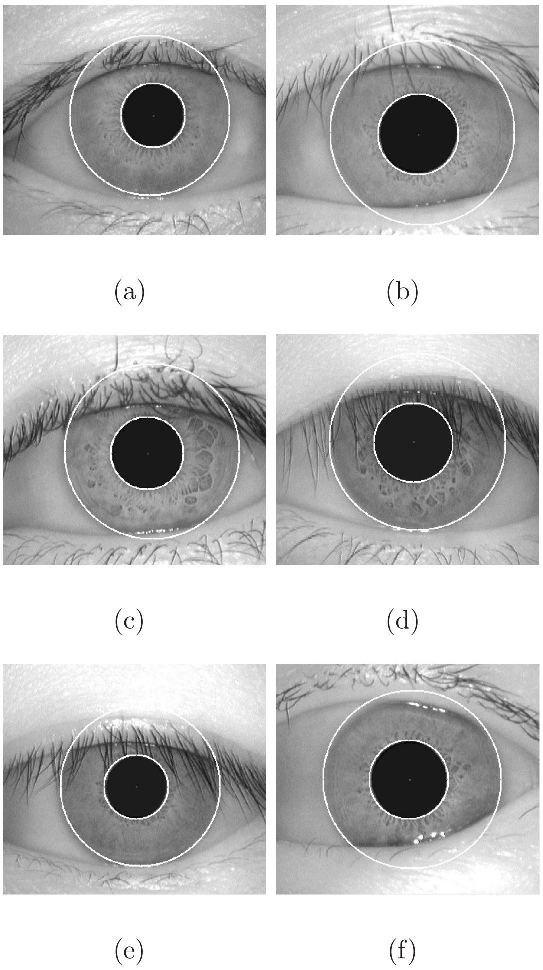
Table 5.2: *Comparison of some recent segmentation algorithms over the CASIA 1.0 database (Results taken from the published work).*

Method	Accuracy
Mateo [177]	95%
Yuan,W [178]	99.45%
Wildes [152]	99.9%
Cui [179]	99.34%
Daugman [151]	98.6%
A.Basit [169]	99.6%
Proposed	100%

method was tested on images from the first 102 subjects. The accuracy rate using the CASIA-IrisV3-Lamp database is 99.68%. Table 5.3 shows the accuracy rate of the proposed method on CASIA-IrisV3-Lamp database. Fig. 5.14 shows the results on some randomly selected images from the CASIA-IrisV3-Lamp database.

Table 5.3: *Comparison of some recent segmentation algorithms using the CASIA-IrisV3-Lamp (Results taken from [148]).*

Method	Accuracy
Masek [176]	79.02%
Ibrahim [148]	98.28%
Proposed	99.55%



(a) (b)

(c) (d)

(e) (f)

Figure 5.13: *Results of proposed method on some noisy images from the CASIA Ver 1.0 database.*

5.6 Computational cost and limitations

The average computational cost is computed for 100 randomly selected eye images from the MMU version 1.0, CASIA Ver 1.0, and CASIA-IrisV3-Lamp databases. MATLAB built-in facility is utilised to obtain the optimal results. Table 5.4 shows a comparison on the computational cost of the proposed method and three similar states of the art method.

In the presence of dense eyelashes and eyebrows, as shown in Fig. 5.15, the proposed

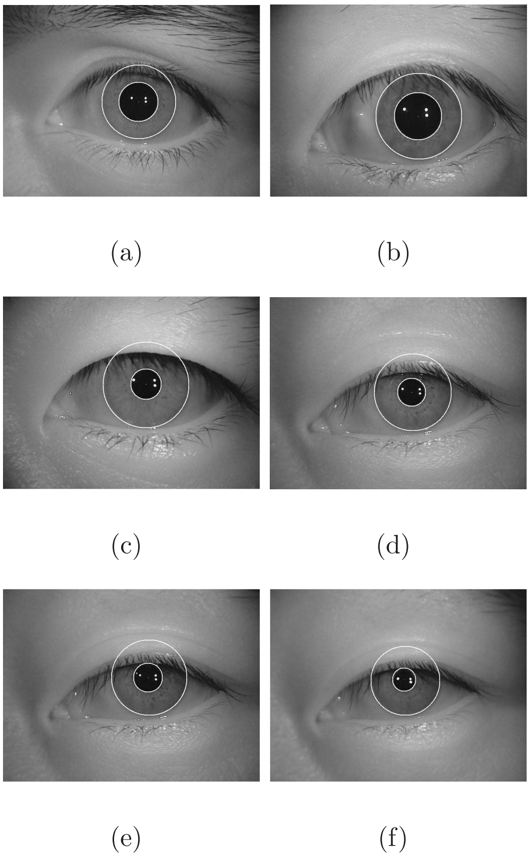


Figure 5.14: *Results of proposed method on some noisy images from the CASIA-IrisV3-Lamp database.*

method fails to locate the true pupillary boundary. Such dense eyebrows and eyelashes effect the LoG filtering along with the region growing. Although, this can be handled by setting a different σ for LoG filter and tuning the region growing, for the generic parameters that are set for the whole database, it fails to locate true pupillary boundary. Similarly, if the iris region is affected with dense eyelashes then a single value of λ_c fail to suppress the effect of eyelashes in the iris region.

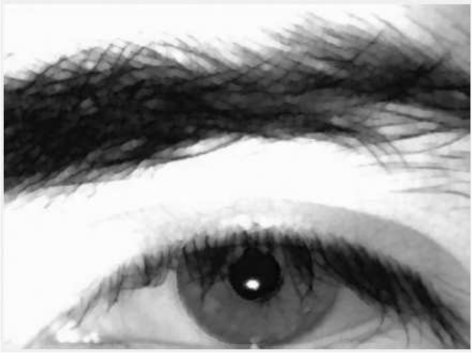


Figure 5.15: *Iris image severally occluded by the eyelashes and eyebrows.*

Table 5.4: *Processing speed (in seconds) comparison of the proposed with some of the existing methods*

Method	MMU version 1.0	CASIA Ver 1	CASIA-IrisV3-Lamp
Ibrahim [148]	0.95	1.1	2.2
Jan [166]	1.5	1.7	3.0
Labati [180]	2.6	3.0	Not available
Proposed	0.44	0.5	1.4

5.7 Conclusion

This chapter presents a fast and an accurate iris segmentation technique for iris biometrics. There are four major contributions. First, we develop a fast and novel method for pupil segmentation that is based on a shape detector and an intensity-based threshold. The use of a LoG filter followed by region growing gives an estimate of the pupil centre and radius.

The true pupillary boundary is refined using the zero-crossings of a second LoG filter. Next, the true orientation of the eye in the image is estimated using a third LoG filter. The orientation facilitates the process of locating the true limbic boundary of the iris and eyelids. Using the zero-crossings of the LoG, the search is initially started from the stable zones and then extended to the occlusion zones. The discontinuities are located, which give indications of eyelids in the iris region. Finally, using the interpolation the iris outer boundary as well as eyelid arcs are estimated. The proposed method also works well in estimating the inner and outer boundary of the iris in the case of partially opened eye images and scattered eyelashes.

The extensive experimental results on three iris databases show that the proposed method is computationally less expensive in achieving the state-of-the-art iris segmentation accuracy.

Chapter 6

Efficient Hardware Implementation Strategy for Local Normalisation of Fingerprint Images[1]

This chapter is an adapted version of a journal article. The section headings from the journal article have been retained. Figures, equations, tables, and references have been re-numbered and are in line with the thesis format. This chapter presents a hardware-based local normalisation technique that can handle the background noise amplification. The proposed normalisation is efficiently implemented in hardware and its efficiency is compared with other state of the art hardware normalisation techniques. With respect to our proposed design, this chapter falls in the fingerprint subsystem. It covers only one module of the fingerprint subsystem, as highlighted in Fig. 6.1.

[1] Published as: Tariq M. Khan, Donald G. Bailey, Mohammad A. U. Khan and Yinan Kong ,"Efficient Hardware Implementation Strategy for Local Normalisation of Fingerprint Images," *Journal of Real-Time Image Processing* , DOI 10.1007/s11554-016-0625-8

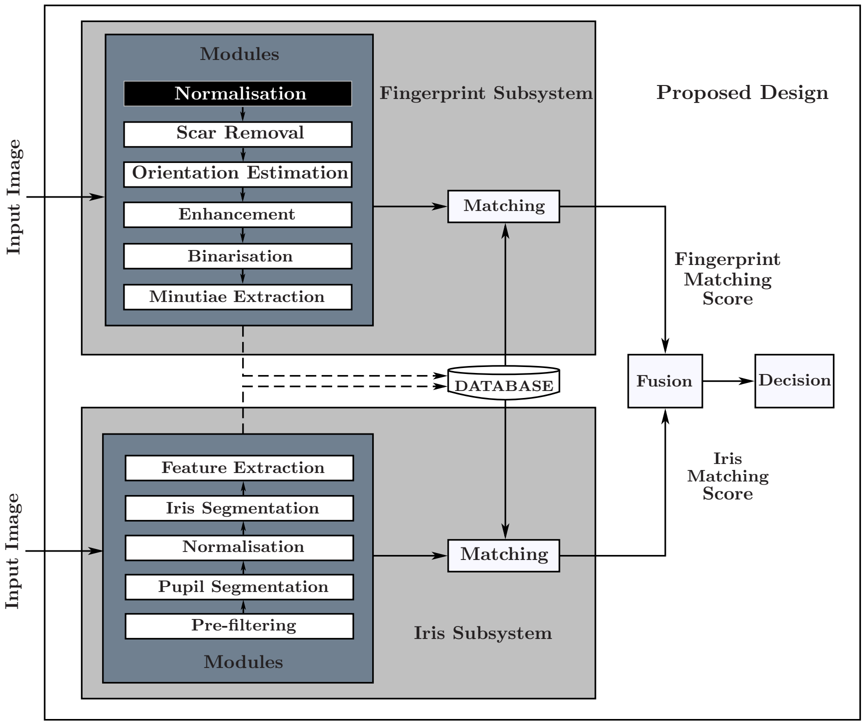


Figure 6.1: *Modules of the proposed design under study*

6.1 Abstract

Global techniques do not produce satisfying and definitive results for fingerprint image normalisation due to the non-stationary nature of the image contents. Local normalisation techniques are employed, which are a better alternative to deal with local image statistics. Conventional local normalisation techniques involve pixel-wise division by the local variance and thus has the potential to amplify unwanted noise structures, especially in low activity background regions. To counter the background noise amplification, the research work presented here introduces a correction factor that, once multiplied with the output of the conventional normalisation algorithm, will enhance only the feature region of the image while avoiding the background area entirely. In essence, its task is to provide the job of foreground segmentation. A mod-

ified local normalisation has been proposed along with its efficient hardware structure. On the way to achieve real-time hardware implementation, certain important computationally efficient approximations are deployed. Test results show an improved speed for the hardware architecture while sustaining reasonable enhancement benchmarks.

6.2 Introduction

In the research work presented, fingerprints are chosen as a test case for the proposed modified local normalisation. Fingerprint image enhancement is held as an important pre-processing step for automatic fingerprint identification systems (AFIS) [181]. This step can directly impact the overall efficiency of a given AFIS. The scanned fingerprint images often show important contrast variations, poor brightness in some areas and gaps in ridge/valley regions. These occur due to the very process of scanning a finger. The finger's middle surface is thicker than the surrounding region. This results in uneven pressure across the finger. The fingerprint scanner registers this phenomenon as a varying contrast across the fingerprint image [119]. Consequently, background variation will add bias for different regions of the same image to disturb the ridge/valley contrast. Since the ridge/valley pattern is identified and classified by its gray-level profile, this effect may worsen the performance of image enhancement algorithms. Since the end goal of a fingerprint image identification system is to convert the gray-level image to binary image, the ridge/valley intensity difference should be as wide as possible to achieve an efficient result. From classification theory point-of-view, binarization is easier if the pixel value distributions of the ridges and valleys are well separated (large inter-class difference) and homogenous within themselves (small intra-class variance). This can only be achieved when the image is properly normalised.

The fingerprint image normalisation process needs an analysis of the various image degradation sources. Generally, a fingerprint image may have narrow dynamic range due to the poor quality of the scanning device or uneven finger pressure at the time of acquisition. Contrast enhancement is a commonly used fingerprint normalisation strategy. In the context of fingerprint image enhancement, the contrast may be defined as the smallest intensity difference between a pixel belonging to ridge region with a nearby pixel of a valley region [182]. Another related terminology used for fingerprint images is their mean pixel intensity (or brightness). The image brightness can be altered by a straightforward method of either adding or subtracting a constant value to all the pixels with almost no effect on the contrast of the image. Since contrast enhancement is a more complicated process, therefore, the focus of the research work presented here is to devise an effective algorithm for improving the local contrast of fingerprint images. This will directly affect the quality of the ridge extraction process using thresholds.

In the literature, different techniques have been used to cope with contrast enhancement. Range normalisation may contain gamma intensity correction, histogram manipulation and normal distribution [183]. Some of the range normalisation techniques do not produce satisfying and definitive results for fingerprint image normalisation. However, histogram manipulation, contrast stretching, and Wiener filtering [183] have been shown to be moderately effective as an initial processing step to enhance fingerprints.

The process of contrast improvement can be illustrated by running some popular fingerprint image normalisation algorithms on a sample image taken from the FVC2004 database [58] as shown in Fig. 6.2(a). The acquired fingerprint image shows considerable local variability, as a result, some ridges appear to be washed away. Further, there is a dark patch right at the centre of the image, making it difficult to threshold the image directly.

The image was processed first with the widely used Hong method [60]. The method was later adopted by [184], [185], and [186] for their hardware implementation for its computationally simple structure. Fig. 6.2(b) depicts the result of Hong method. The contrast at different parts of the image is equalised. The pixel values are now distributed over the whole display range. The dark patch, however, is still persistent, suggesting that Hong method is unable to deal with local strong variations.

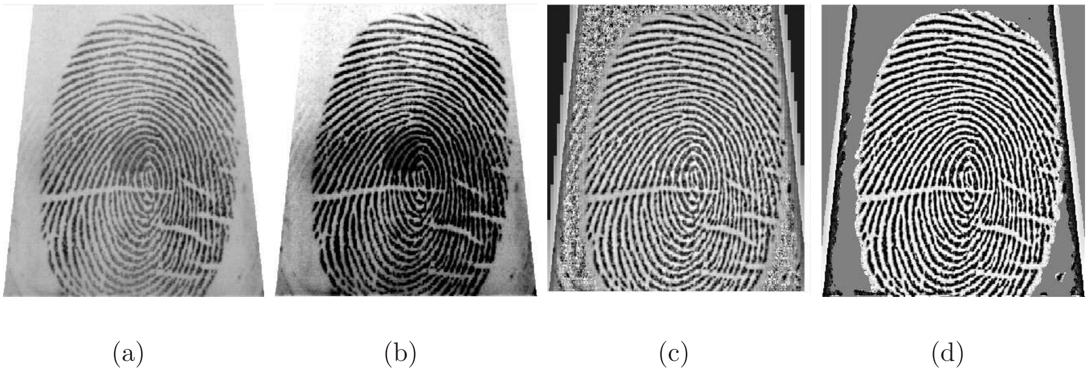

(a) (b) (c) (d)

Figure 6.2: *a) Sample image FVC2004\Dbs\DB2_A\33_3. b) Image enhanced by global normalisation [60]. c) Image enhanced by block normalisation [187]. d) Image enhanced by the proposed method.*

Later it was suggested by [188], and [189] that local image statistics, such as mean and variance in a small neighbourhood, have to be incorporated in the contrast improvement strategy. The local normalisation method comprises of first dividing the image into appropriate small neighbourhoods, and then normalising these neighbourhoods with respect to their local mean and variance. This will result in shaping these neighbourhoods to have a ridge/valley pattern with better contrast. Mathematically, it can be represented as

$$g(x,y) = \frac{I(x,y) - m_f(x,f)}{\sigma_f(x,y)} \tag{6.1}$$

where $I(x,y)$ is the input image, $m_f(x,y)$ is an estimation of a local mean of $I(x,y)$ and $\sigma_f(x,y)$ is an estimation of the local contrast (such as the standard deviation). In line with this suggestion, a block normalisation algorithm is implemented in [187]. The result of the method with 17 × 17 large window and 5 × 5 small window is shown in Fig. 6.2(c). Although the contrast is restored with no black patch at the centre, the amplitude of the granular noise in the background is significantly lifted. This happens due to the fact that the background area has almost zero local variance [190], thus resulting in division a small number, which amplifies the noise structure.

In this chapter, a modified local normalisation procedure is proposed that enhances the contrast of the foreground ridge/valley area uniformly with almost no normalisation for the background region. To do so, local image statistics are involved for segmentation of the fingerprint image into foreground and background regions. However, rather than use a binary segmentation, a gradual process in the form of a continuous function to avoid undesirable edge effects. To estimate the local mean and variance low-pass Gaussian filtering is employed. Generally, using a Gaussian filter gives the best results because the smooth transition can minimise artefacts (it has good stop-band performance in the frequency domain). To circumvent the amplification of background noise, this chapter proposes a correction factor in the form of a monotonically increasing function of local variance values.

The rest of the chapter is organised as follows. Related work is presented in Section 6.3. In Section 6.4 the proposed image normalisation algorithm is described. Section 6.5 presents the hardware adaptation of the proposed algorithm. In Section 6.6 experimental results are discussed and finally the chapter is concluded in Section 6.7.

6.3 Related Work

Local normalisation methods that have been used for fingerprints include local histogram equalisation [183], and local histogram matching [191]. [192] proposed a local normalisation technique for fingerprint image normalisation based on block processing. The mean and variance for normalisation are calculated on a block-by-block basis. Block processing is significantly faster than normal local normalisation, as all of the pixels within a block are processed using the same mean and same variance. [187] also presented a block-based local normalisation algorithm in which an entire block of pixels of size $W \times W$ is involved to compute the local normalisation parameters.

From a hardware implementation perspective, applying a large number of filters commonly requires a significant amount of computing resources. The best way to achieve good real-time performance is to utilise the parallel processing capabilities of hardware. Although several FPGA implementations have been presented in the literature for fingerprint image normalisation, most of them use the global mean and variance. [184] implemented a fingerprint image normalisation similar to the one proposed by [60]. In this implementation, global and local means as well as global and local variances are linearly mixed to process a 16×16 block. [186] used an 8×8 block instead of 16×16. In this technique, parallel processing is used, which increases the overall speed of the system as compared to [184]. [185] also used Hong's technique for image normalisation. He divided the image into four sub-images for parallel processing. Vitabile, Qin and Fons implementations all require two processing passes for the whole image.

In this chapter a local normalisation technique based on Gaussian filtering is used to estimate the local mean and variance on a pixel-by-pixel basis. This is shown to be effective and makes its hardware implementation easier than the existing local normalisation techniques. Filtering and normalisation are pipelined so that only one pass through the whole image is required, which significantly speeds up the proposed technique compared to the existing implementations.

6.4 Modified Local Normalisation

The acquired fingerprint image from the sensor usually faces the problem of a non-uniform background and spatially varying contrast. The non-uniform background can be attributed due to many sources arising in the fingerprint sensors: ageing filaments, faulty reference voltages, contaminated apertures, or non-uniform support film fabrication [193]. Removal of the non-uniform background is important for later processing stages such as image registration based on correlation and segmentation based on intensity thresholds.

The conventional normalisation model is based on two phases; removing the non-uniform background and then restoring the local contrast, as depicted in Fig. 6.3. The first phase is to estimate the non-uniform background for an acquired image using a low-pass Gaussian filter with smoothing parameter σ_1 and then subtracting this Gaussian-weighted average from each pixel of the original image. The parameter σ_1 can be set by utilising the fact that the filtered image should contain only the background changes (low-frequency content) and with almost no trace of features (which are high-frequency content). Therefore, σ_1 should be comparable to the size of the largest feature of interest present in the acquired image. A smaller σ_1 will incorporate some of the fingerprint ridge details into the background while a much larger σ_1 will not capture some of the background variations across the image as shown in Fig. 6.4. Since a fingerprint image is characterised by its locally periodic ridge/valley structure, the value of σ_1 can be set equal to the width of the average ridge. Specifically, for the experiments conducted for the FVC2004 fingerprint images, the size of σ_1 was set to 5.

Although, background subtraction is successful in removing variation in the background, it does not address the variation in local contrast. The second phase is to normalise the contrast as shown in Fig. 6.3. In the second phase, the local variance of the image is computed as an estimate of the local contrast. The subtracted image is divided pixel-wise by the standard deviation of its spatial neighbours to normalise the contrast.

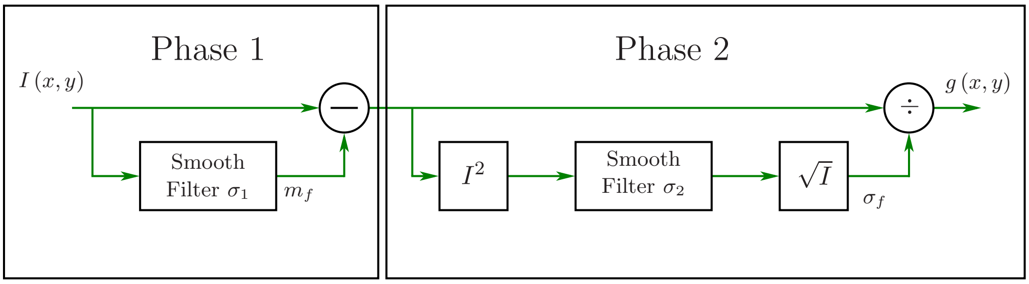


Figure 6.3: *Conceptual model for image local normalisation.*

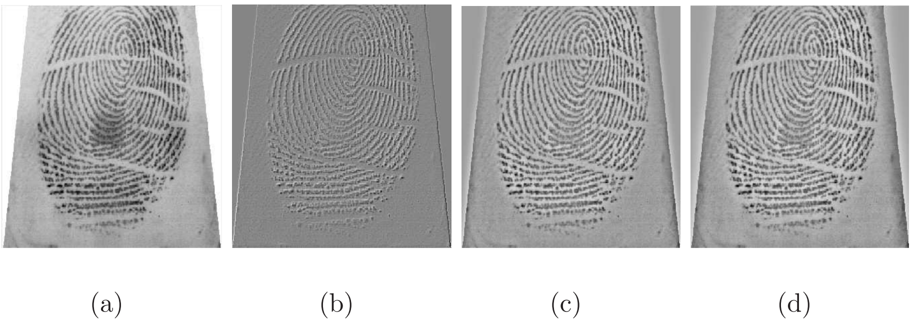


Figure 6.4: *Impact of σ_1 on fingerprint background removal: a) Sample image from FVC2004\Dbs\DB2_A database. b) $\sigma_1 = 1$. c) $\sigma_1 = 5$. d) $\sigma_1 = 10$.*

Again, the size of the local variance filter depends on the size of the texture elements. Large texture elements require a larger window [189]. After background removal, squaring the pixel value will double the base frequency. So σ_2 is commonly taken to be $\sigma_1/2$ [194].

Areas of no ridge/valley structure have low values for the local variance leading to the amplification of the low-amplitude noise structure as evident in Fig. 6.5. Subsequent fingerprint image enhancement such as anisotropic diffusion, commonly used for enhancing elongated ridge-valley structure [195], [106] may also enhance the noise creating artificial linear features in the background as shown in Fig. 6.6.

Therefore, it is necessary to suppress the background noise while keeping the foreground feature enhancement at a maximum. One approach is to apply the division selectively to only those regions where there is ridge/valley structure. This requires classifying image pixels into two categories: an active region of interest and a non-active smooth background region. To do so, a metric is needed that produces low values for a quiet background and high values for active regions. This can be derived from the local variance, which has already been calculated. We propose a function of local variance which is used as a multiplying factor for the outcome of the second phase image. The factor is defined as

$$M = 1 - \exp\left(-\frac{\sigma_f^2}{2C^2}\right) \tag{6.2}$$

where σ_f^2 is the local variance and C is a user-defined parameter to regulate the noise suppression power in background areas. The value of C is in the range 0-1, however, in our experiments, the value of 0.3 was adequate in all cases. The graph of the factor with increasing values of local variance is shown in Fig. 6.7. The factor is small for low contrast, and then it rises at a rate controlled by the constant C, to reach an ultimate value of 1. This correction factor can be introduced as a third processing phase, as shown in Fig. 6.8. Fig. 6.9 compares the effect of normalisation with and without the correction factor on three noisy images.

The whole process of modified local normalisation can be illustrated with three example images shown in Fig. 6.5. Row 1 shows three sample image taken from FVC2004 DB1_A. Row 2 shows the Gaussian filtered images having $\sigma_1 = 5$. Row 3 shows the mean

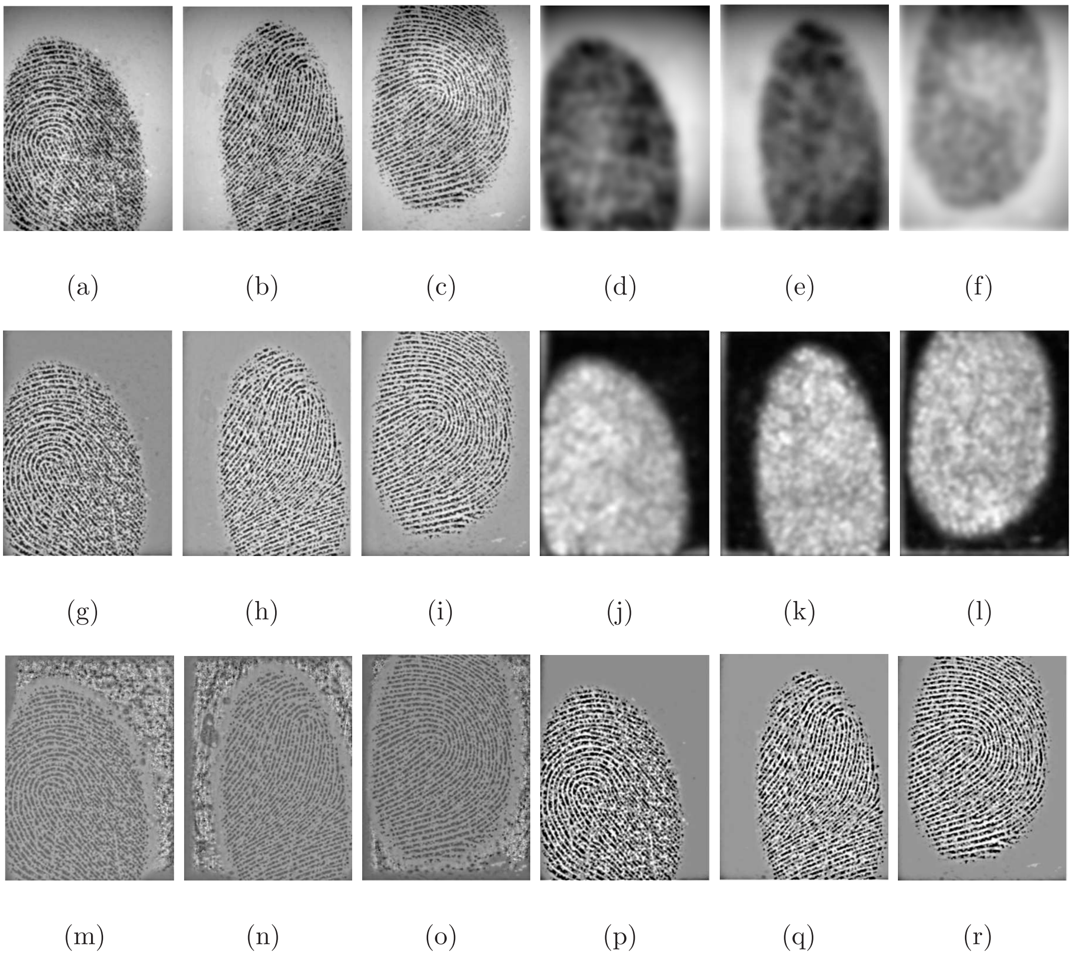
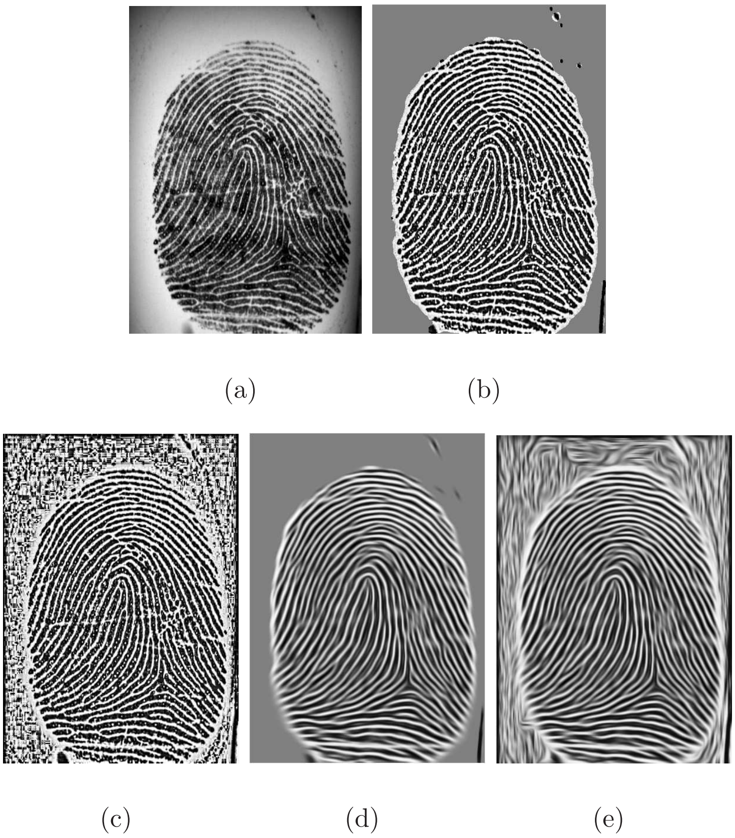


Figure 6.5: *Modified local normalisation illustration. Starting row-wise, the original images are shown in Fig. (a)-(c). Fig. (d)-(f) depicts the estimated non-uniform background, and Fig. (g)-(i) shows the uniform background images obtained by subtraction. Next the local variance image is computed, and displayed in Fig. (j)-(l). The division of the uniform background image by the local variance image is shown in Fig. (m)-(o). We observe a large noise amplification phenomenon. To mitigate it, a correction factor is introduced, and its impact is shown in Fig. (p)-(r).*



(a) (b)

(c) (d) (e)

Figure 6.6: *The impact of diffusion process on normalisation: a) Sample image. b) Image normalised with factor. c) Image normalised without factor. d) Diffused image after normalising with factor. e) Diffused image after normalising without factor.*

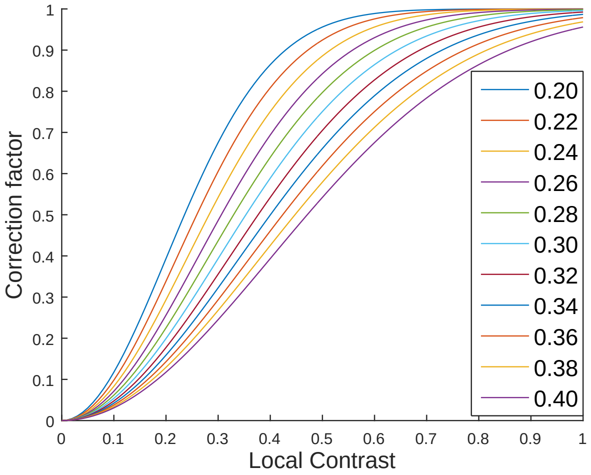


Figure 6.7: *The effect of local contrast σ_f on the correction factor as parameterized curve of C.*

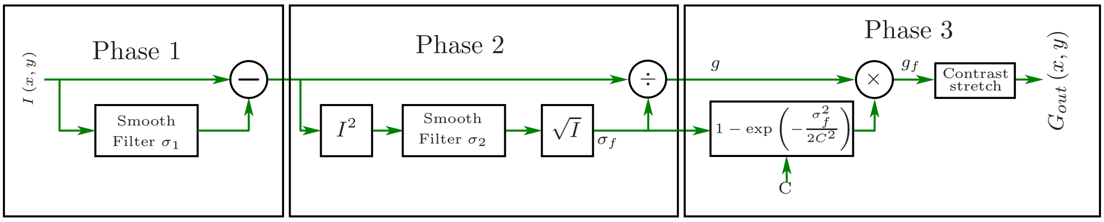
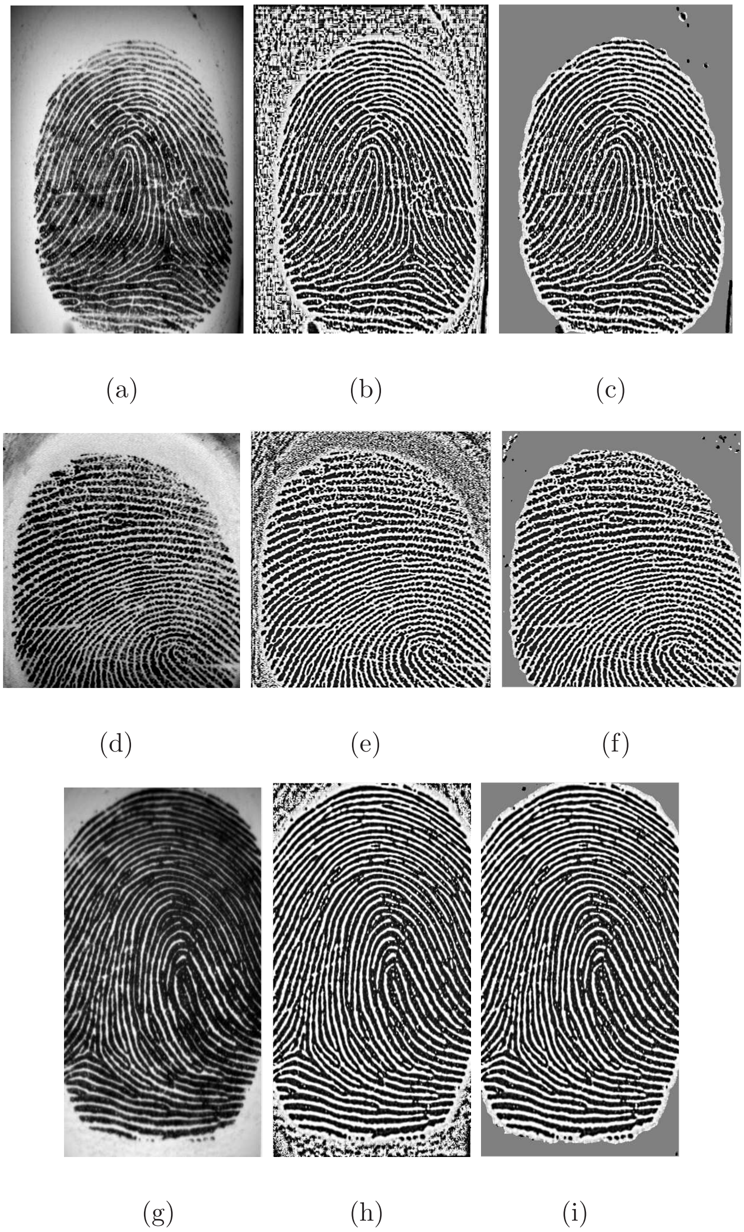


Figure 6.8: *Proposed model for image local normalisation.*



(a) (b) (c)

(d) (e) (f)

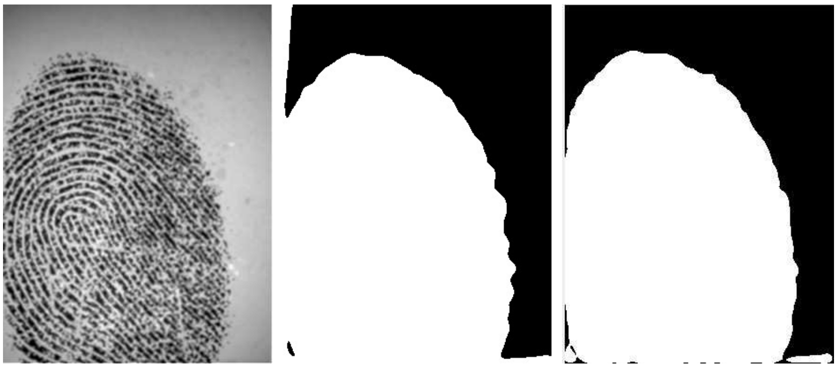
(g) (h) (i)

Figure 6.9: *Impact of correction factor M. Column 1 shows sample images. In column 2, image are normalised without multiplication by M. In column 3, images are normalised including M.*

subtracted images. Row 4 shows the variance images, with $\sigma_2 = \frac{\sigma_1}{2}$. Row 5 shows the normalised image without using the proposed multiplying factor. Row 6 shows the normalised images of the proposed method.

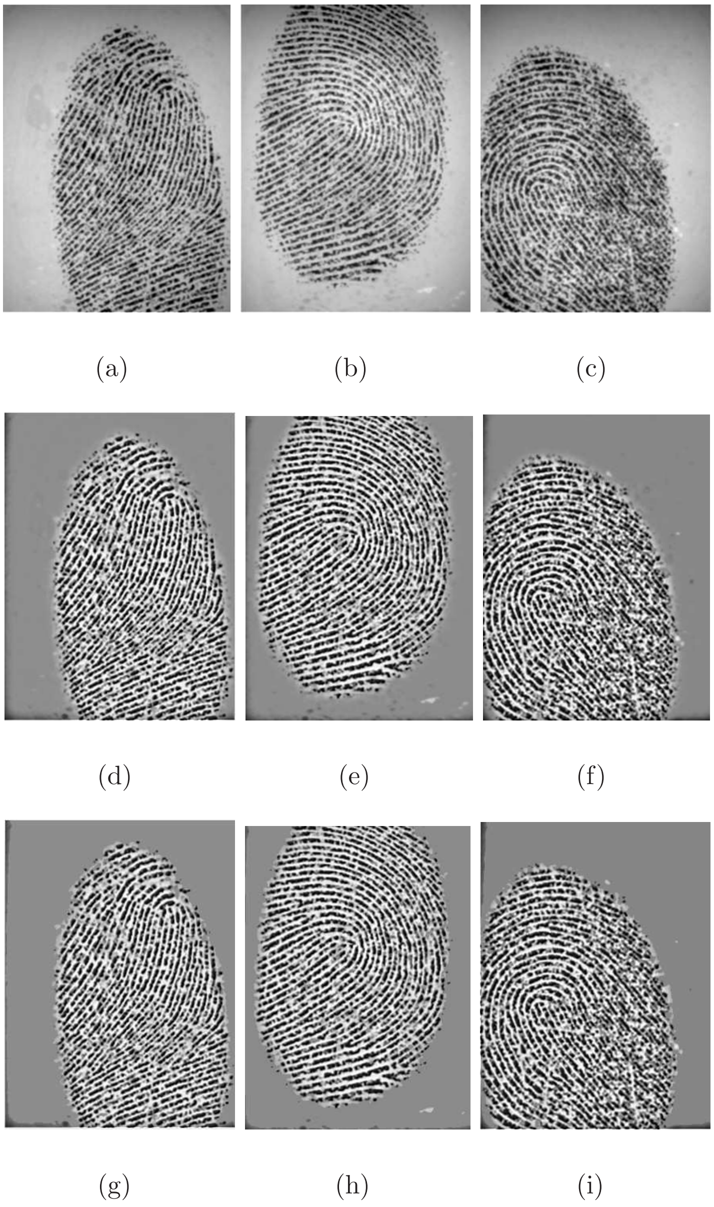
It can be argued that the use of a correction factor is doing the job of background segmentation. Usually, the local variance is used to find a contiguous block of the active region by thresholding [196], [88]. This results in a binary mask as shown in Fig. 6.10. By comparing the binary mask with that of the correction factor impact, we find that the suggested correction factor is an increasing function based on local variance values, and provides better control over designating the ridge/valley regular structure versus the background, as shown in Fig. 6.11.

Although this technique provides good results, when it comes to its hardware implementation, it has a few drawbacks. Calculating the local variance a requires square root which in hardware becomes costly. Another problem with the use of standard deviation as an estimate of the local contrast is that it tends to be more sensitive to larger values. An alternative approach to estimating the local contrast is to take the average absolute value.



(a) (b) (c)

Figure 6.10: *a) Sample Image. b) Binary mask created using local variance based thresholding. c) mask created by multiplicative factor M.*



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 6.11: *Row 1 shows three sample noisy images. Row 2 shows the results of the proposed normalisation method. In Row 3 the images are first segmented using local variance measure and then normalised without multiplying the correction factor M.*

6.5 Image Normalisation Hardware Structure

The aim of this work is to design an efficient algorithm for fingerprint image local normalisation that best suits a hardware implementation. For this purpose, a new method for image normalisation is proposed which is efficiently implemented in hardware. Our suggested hardware architecture for this modified local normalisation process is shown in Fig. 6.12. First, a Gaussian filter with of $\sigma_1 = 5$ is applied to the input image I. Then, the difference D between the image I and the background estimation m_f is calculated for every pixel:

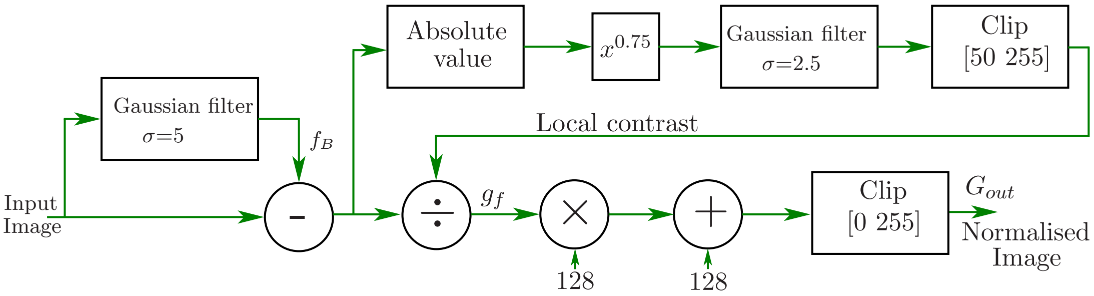


Figure 6.12: *Proposed model for image local normalisation for hardware implementation.*

$$D(x, y) = I(x, y) - m_f(x, y) \tag{6.3}$$

The next step is to estimate the local contrast within the image. Rather than use the standard deviation, the average absolute value is used instead. The magnitude is obtained by applying the absolute operator to the image pixels. Through experiments, it was noticed that better results were obtained when using dynamic range compression after the absolute value. For this purpose, the power-law transformation with $\gamma = 0.75$ is employed here to compress the high contrast pixels relative to those with low contrast. Fig. 6.13 shows the effect of the power-law transformation with different γ values on the absolute difference image. This is then averaged locally with another Gaussian filter, with

$\sigma_2 = 2.5$, to estimate the local contrast. Rather than implementing the correction factor M on hardware, a similar effect can be obtained by clipping the local contrast into the range [50-255]. This avoids over enhancing noise by low contrast values while retaining the relative strength of already high contrast regions. The resultant image is divided by the local contrast, and the output is scaled by a factor of 128, and the offset by 128 to get an unsigned image suitable for display:

$$G_{out} = \begin{cases} 0, \quad if\, g_f < -1 \\ 255, \quad if\;\; g_f > 1 \\ 128 \times g_f + 128, \quad otherwise \end{cases} \tag{6.4}$$

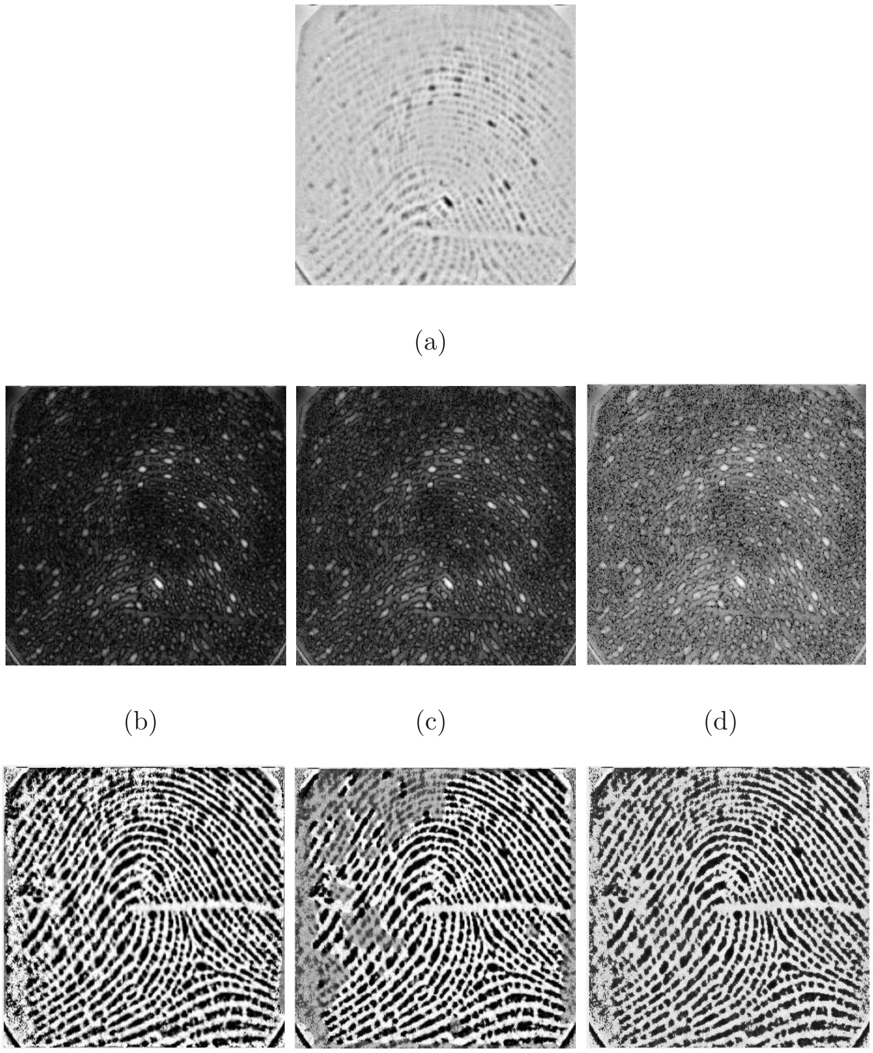
6.5.1 Gaussian Filter Implementation

In two dimensions, the Gaussian function is

$$G\left(x,y\right) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \times \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{y^2}{2\sigma^2}} \tag{6.5}$$

For large σ, the size of the filter needs to be large, which makes the hardware implementation expensive. Although a smaller rectangular filter can be used to achieve the same level of smoothing, its poor frequency response can give artefacts especially for regular patterns such as those within fingerprints. For this reason, a Gaussian filter is used. For the hardware implementation, a size of $\sigma = 5$ is used with a 19×19 window. Truncating the window limits the stop-band of the Gaussian filter at high frequencies to -30 dB.

As the Gaussian is separable, this allows the filter to be implemented as a cascade of one-dimensional Gaussian filters (1×19 and 19×1). Fig. 6.14 shows the implementation of 1×19 and 19×1 Gaussian filters with $\sigma = 5$. Although the filter can be decomposed to only use adders [197], the need of such decomposition is less important on modern FPGAs



(a)

(b) (c) (d)

(e) (f) (g)

Figure 6.13: *Impact of proposed power-law transformation on absolute difference image: a) Sample noisy image. b) Power-law transformation with $\gamma = 0.75$. c) Power-law transformation with $\gamma = 0.5$. d) Power-law transformation with $\gamma = 0.25$. e) Normalised image with $\gamma = 0.75$. f) Normalised image with $\gamma = 0.5$. g) Normalised image with $\gamma = 0.25$*

where high-speed pipelined multipliers are plentiful. The optimised hardware multipliers are hard to out-perform with the relatively slow adder logic of the FPGA fabric [198].

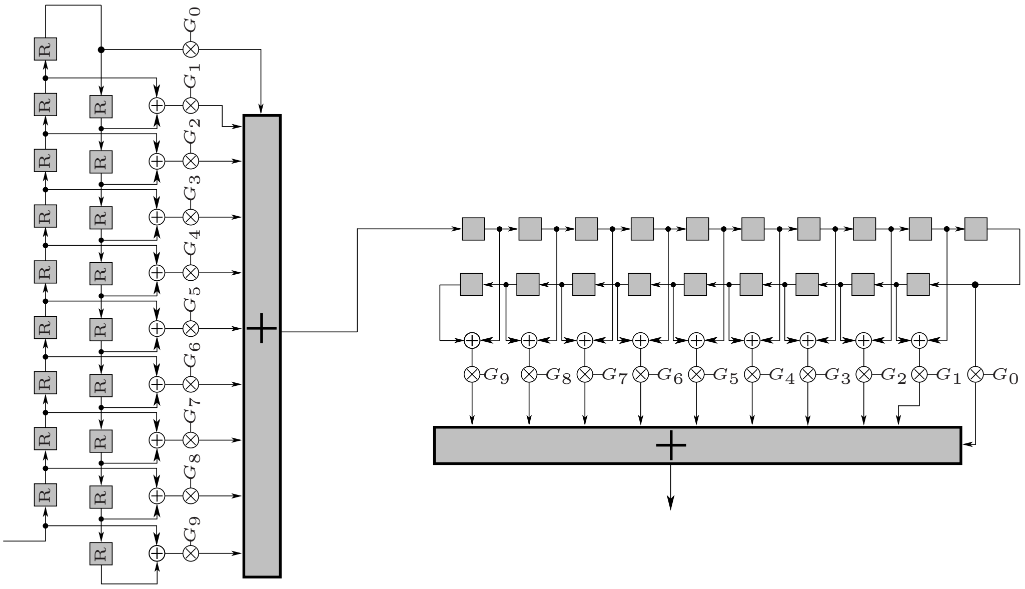


Figure 6.14: *Hardware implementation of a (*1×19 *and* 19×1*) Gaussian filter with* $\sigma = 5$*. The* G_x *are filter coefficients. For the vertical filter, the boxes represent row buffers.*

6.5.2 Remaining Operations

The power law transformation is easiest to implement as a lookup table. The input is the difference image from stage 1, with the lookup table also performing the absolute values operation. The remaining operations: clipping, scaling and division are relatively straight forward to implement.

6.6 Experimental Results and Discussion

The experiments were performed on the FVC2004 (Fingerprint Verification Competition) database which consists of four sub-bases, in which images are captured with four different sensors. DB1_A, DB2_A, DB3_A, DB4_A. Each sub-base includes 800 fingerprints, of which 8 are of the same person. Each sub-base has a different image size. The detail of the sub-base sizes is given below:

- In sub-bases DB1_A the size of the fingerprint image is 480 × 640
- In sub-bases DB2_A the size of the fingerprint image is 364 × 328
- In sub-bases DB3_A the size of the fingerprint image is 480 × 300
- In sub-bases DB4_A the size of the fingerprint image is 384 × 288

For simulation, the program was written in MATLAB and run on a 3.40 GHz Core i7 processor with 16 Gb memory. Image normalisation is used as a pre-processing step for fingerprint image enhancement. For image enhancement, coherence enhanced diffusion is used [57]. This is a well-known algorithm which is used to smooth the ridge structure in fingerprint images. The diffusion process ends after 40 iterations (steps=40) with a step size of 0.25.

For real-time implementation, VHDL is used for fingerprint normalisation, and the design is tested on a Cyclone III FPGA and simulated with ModelSim.

To assess the efficiency of the verification system the FAR (false acceptance rate) and FRR (false rejection rate) are calculated as:

$$FAR = \frac{Number\,of\,rejected\,genuine\,claims}{Total\,Number\,of\,genuine\,accesses} \times 100 \tag{6.6}$$

$$FRR = \frac{Number\,of\,accepted\,imposter\,claims}{Total\,Number\,of\,imposter\,accesses} \times 100 \tag{6.7}$$

Finally, the EER (equal error rate) was used as a success rate indicator, marking the point where FRR and FAR are equal.

$$EER = \frac{FAR + FRR}{2}, \; if \;\; FAR = FRR \tag{6.8}$$

Another metric is the ZeroFMR which is the lowest value of FRR, where false mismatch does not occur anymore.

Table 6.1 compares the equal error rate EER (%) and ZeroFMR (%) assessment of the proposed algorithm with the existing algorithms. It can be observed that Kocevar block-local normalisation is more accurate than the global normalisation method proposed by Hong. Kocevar normalised the image with 3 different block sizes, 2 × 2, 4 × 4 and 8 × 8. His 2 × 2 normalisation gives better results than the other two. However, using a small window affects the processing speed of the system. The authors claim that 8 × 8 block-local normalisation best suits real-time application. Regarding performance, the proposed normalisation method is more accurate than existing methods. If we compare our method with 2 × 2 block-local normalisation then our method gives better accuracy. One reason is the use of a correction factor to suppress the noise amplification in the background area that enhances the overall performance of the proposed method. The hardware realisation gave slightly worse performance than the software algorithm, because of the simplifications and approximations made to give an efficient implementation. Its performance was still similar to the best of the competing state-of-the-art methods.

6.6.1 Comparison to Other Hardware Architectures

In Table 6.2 the proposed FPGA-based structure is compared with existing FPGA-based implementations in terms of processing speed and hardware resources. For the comparison, a random fingerprint image of size 512 × 256 is used. It can be observed that the proposed method is significantly faster than all the existing methods. The existing hard-

Table 6.1: *Equal error rate EER (%) and ZeroFMR (%) assessment of proposed algorithm with existing algorithms, average over all the images from DB1_A and DB1_B of FVC2004 database.*

FVC2004	Compared Algorithm	EER (%)	ZeroFMR (%)
DB1_A	No enhancement	16.07	64
	Oriented linear diffusion without normalisation	21.29	63.71
	Oriented linear diffusion + [184] normalisation	12.02	54
	Oriented linear diffusion + [186] normalisation	9.74	38.1
	Oriented linear diffusion + [187] 2 × 2 block-local normalisation	9.08	30.36
	Oriented linear diffusion + [187] 4 × 4 block-local normalisation	9.93	39.93
	Oriented linear diffusion + [187] 8 × 8 block-local normalisation	11.53	51.54
	Oriented linear diffusion + Proposed Method MATLAB	8.57	29.5
	Oriented linear diffusion + Proposed Method FPGA	8.91	30.7

ware architectures require two passes for image normalisation. One disadvantage of a two-pass process is that it requires the complete image to be stored during processing i.e. large images cannot be processed completely on an FPGA due to a lack of sufficient on-chip memory. The data transfer overhead in transferring the full image to and from external SRAM makes the Fons method slower. The use of a single pass with streamed processing makes the proposed architecture significantly faster than the Fons algorithm. On the other hand, one disadvantage of using a Gaussian filter is that it requires more memory for the row buffers and also multipliers to implement it on hardware, which can be observed in Table 6.2. This directly affects the overall resource usage. If we compare the resource usage with Fons, the proposed method require more memory and embedded multipliers than the Fons method because of the use of a Gaussian filter twice in the normalisation. We consume fewer registers and logic cells than the Fons method, although a direct comparison is a little misleading because our design makes use of embedded multipliers which reduces our logic requirements. All of these factors directly affect the maximum operating frequency of the design. Our design operates at 128.3 MHz, while the Fons design operates at 50 MHz. In total, the proposed method consumes resources comparable to the Fons method with almost 20 times faster image processing time and much better performance in terms of image normalisation. A detailed compilation report of the proposed method on the different image sizes of the FVC2004 database is presented in Table 6.3; a compilation report of existing methods is not available on this databases for comparison. The only change for the different database is the size of the memory for the row buffers.

Fig. 6.15 compares the impact of the proposed and Hong normalisation algorithm which is used by most of the existing hardware methods. For this test, three sample images are taken from the FVC2004 database. For thresholding Otsu's thresholding method is used [96]. Thresholding plays a vital role in minutiae extraction. It can be seen that the proposed algorithm normalises the image better than Hong. Because of better binarization, the stress on the subsequent process, image enhancement as well as thresholding, is less than other methods. It is found that with the proposed normalisation method a simple thresholding method can give much better results, which is not possible with the image statistics based normalisation methods.

6.7 Conclusion

The research presented here deals with lifting the contrast of poor quality fingerprint images. The images often suffer due to the local variations, which can be corrected by subtracting local mean and then normalising the local contrast. However, this amplifies noise in the background. A correction factor is introduced that effectively segments the foreground and compensates for the division in background regions.

The software normalisation system introduced here is translated into an efficient hardware architecture, whose efficiency is compared with other state of the art hardware normalisation systems. Our hardware structure for fingerprint image normalisation managed to efficiently speed up the image processing time by a factor of 18 and improve some of the resource utilisation of the FPGA. For real-time applications, the speed of fingerprint image processing is very important, especially with databases that include a high number of fingerprints. One of the main advantages of using FPGAs for the implementation of image processing applications is that their structure can exploit spatial and temporal parallelism. The proposed structure for fingerprint image normalisation speeds up the processing time on FPGA-based implementations and also achieves better enhancement results on a public FVC2004 with local normalisation and oriented linear diffusion than existing methods. Although the structure of ridges and valleys does not change in the nor-

Table 6.2: *Comparison of the proposed method with some existing FPGA based normalisation methods on a sample image of size* 256×512

Technique	Processing time (ms)	Registers	Logic cells	Memory bits	Embedded multipliers	Maximum operating frequency [MHz]
[184]	85	N/A	N/A	N/A	N/A	N/A
[199]	25	1384	4729	38912	0	50
[185]	21	N/A	N/A	N/A	N/A	20.9
Proposed Method	1.16	1016	1901	58484	18	128.3

Table 6.3: *Compilation report of the proposed algorithm on a low cost Cyclone III FPGA*

Resources	Available	FVC_DB1_A	FVC_DB2_A	FVC_DB3_A	FVC_DB4_A
		480 × 640	364 × 328	480 × 300	384 × 288
Logic Elements	15408	1901(12%)	1901 (12%)	1901(12%)	1901 (12%)
Combinational function	15408	1750 (11%)	1750 (11%)	1750 (11%)	1750 (11%)
Dedicated logic register	15408	1016(7%)	1016 (7%)	1016 (7%)	1016(7%)
Memory bits	516k	106k (21%)	81k (16%)	106k (21%)	86k (17%)
Embedded Multipliers	112	18 (16%)	18 (16%)	18 (16%)	18 (16%)

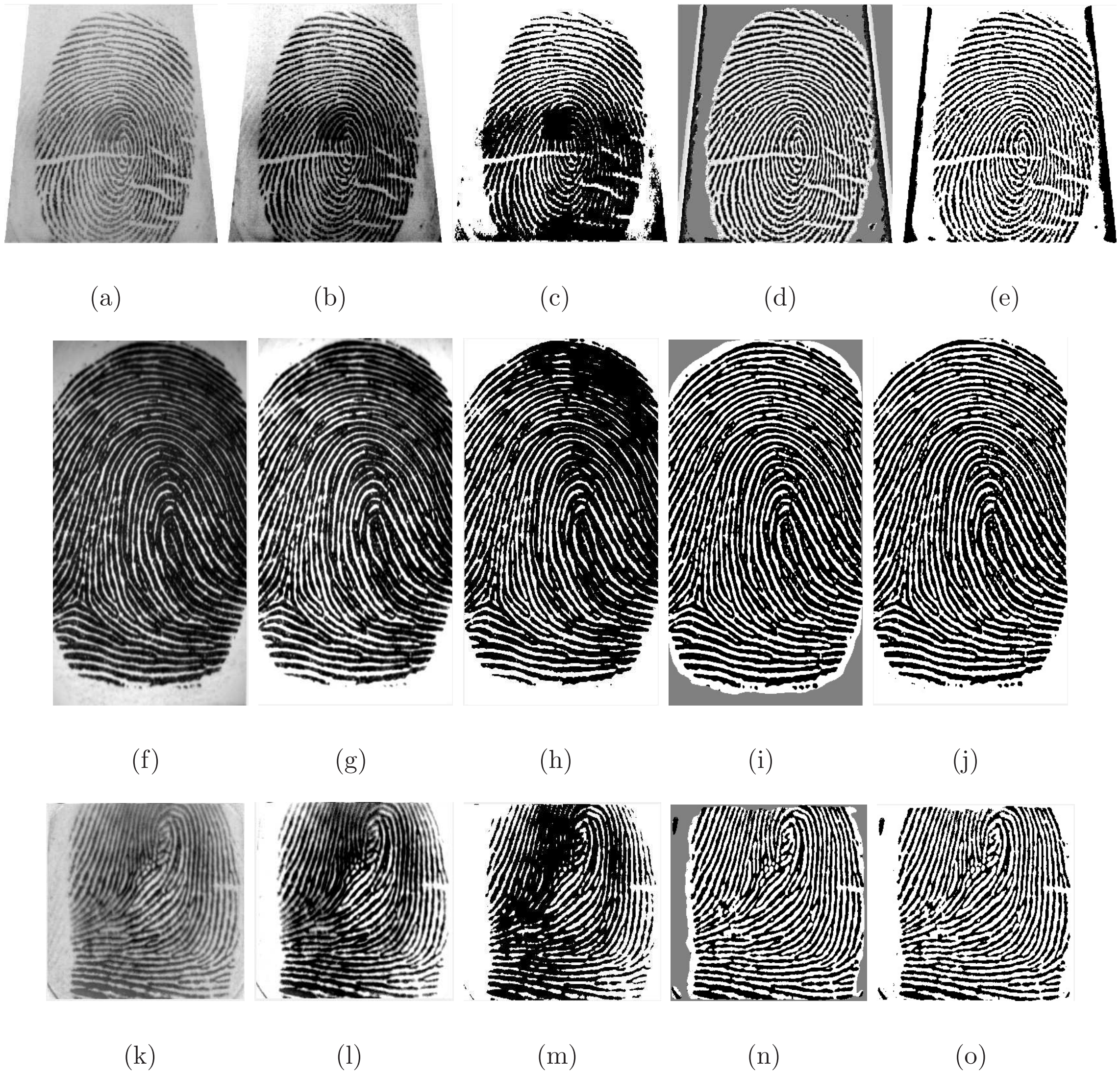


Figure 6.15: *The impact of proposed local normalisation method on low contrast fingerprint images. Column 1 shows three sample images taken from FVC2004 database. Column 2 shows the output of Hong normalisation method applied on columns 1 images. Column 3 shows threshold images after normalised by Hong method. Column 4 shows the output of proposed normalisation method applied on columns 1 images. Column 5 shows threshold images after normalised by proposed method.*

malisation process, normalisation can influence the structure in further fingerprint image enhancement procedures with contextual filters.

Chapter 7

Efficient Hardware Implementation For Fingerprint Image Enhancement Using Anisotropic Gaussian Filter[1]

This chapter is an adapted version of a journal article. The section headings from the journal article have been retained. Figures, equations, tables, and references have been re-numbered and are in line with the thesis format. This chapter presents a fast and efficient hardware implementation of fingerprint image enhancement. In the proposed method, the traditional Gabor filter is modified and decomposed it into an isotropic and an anisotropic filter. By this decomposition, we manage to efficiently speed up the image-processing time and improve the resource utilisation of the FPGA. With respect to our proposed design, this chapter falls in the fingerprint subsystem. It covers four modules of the fingerprint subsystem, as highlighted in Fig. 7.1.

[1] Published as: Tariq M. Khan, Donald G. Bailey, Mohammad A. U. Khan and Yinan Kong,“Efficient Hardware Implementation For Fingerprint Image Enhancement Using Anisotropic Gaussian Filter,” *IEEE Transactions on Image Processing* , minor revision submitted.

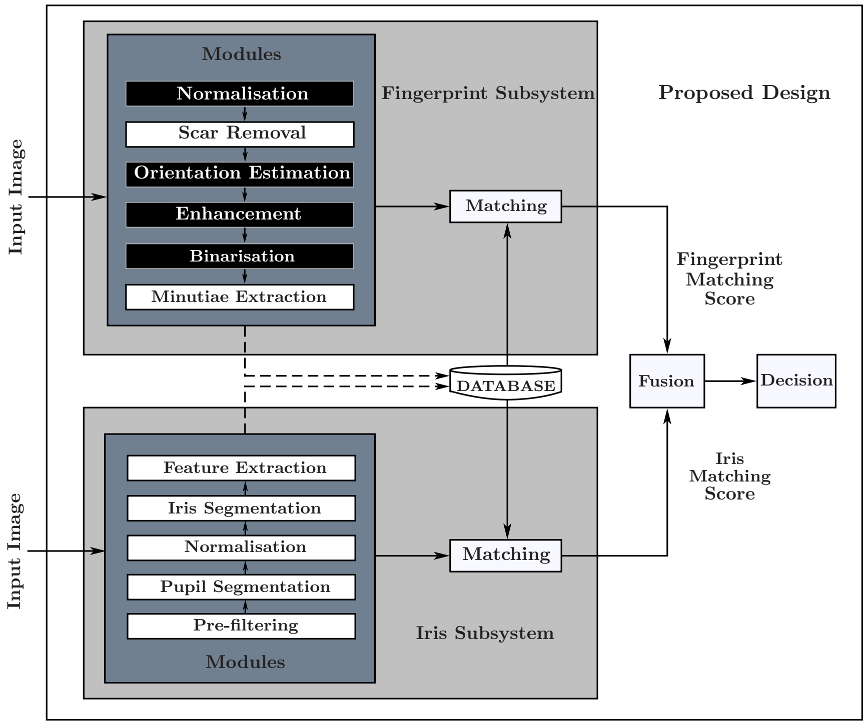


Figure 7.1: *Modules of the proposed design under study*

7.1 Abstract

A real-time image filtering technique is proposed which could result in faster implementation for fingerprint image enhancement. One major hurdle associated with fingerprint filtering techniques is the expensive nature of their hardware implementations. To circumvent this, a modified anisotropic Gaussian filter is efficiently adopted in hardware. In the proposed structure, for a middle-range reconfigurable FPGA, both parallel compute-intensive and real-time demands were achieved. Test results show an improved speed for its hardware architecture while maintaining reasonable enhancement benchmarks.

7.2 Introduction

Biometrics is a fast progressing science that deals with human identification by using their traits or characteristics. The science is especially interested in measurable, albeit distinctive, characteristics to label and to some extent describe individuals [60]. These characteristics are often categorised as behavioural versus physiological. Physiological characteristics deal with the shape of the body. Examples include the face, DNA, palm print, hand geometry, retina, iris and fingerprints. Behavioural characteristics are related to the pattern of behaviour of an individual, including but not limited to typing rhythm, gait, and voice. Among all the characteristics, a fingerprint is the most widely used.

Fingerprint identification is one of the oldest biometric techniques that has proven its worth in numerous applications [106]. Every person has a unique and immutable fingerprint that can be acquired by a scanner. A fingerprint surface consists of parallel ridges and furrows. However, at some points, one ridge splits into two (ridge bifurcations) and, at other points may even terminate and continue no more (ridge endings). These local ridge singularities (deviations from normal parallel behaviours), also known as minutiae points, are distinctive [200] and it is these that are primarily utilised for identifying individuals.

Although associating identity with a fingerprint impression can be accomplished through image correlation-based methods [201], more commonly, minutiae points are matched [119, 202]. The minutiae-based representation consists of the set of ridge endings and bifurcations along with their spatial location and direction on the fingerprint surface. Having a small template size and high accuracy, the minutia-based representation is considered favourite by many experts as compared to correlation based methods [203]. Minutiae-based fingerprint matching is widely used by both machines and human experts. This representation has now become a standard, mostly used by forensic experts, for the exchange of information between different systems across the world [204].

To make use of a minutia-based representation, it is essential that the minutiae are extracted accurately. Extraction starts with the acquisition of fingerprints by a scanner. Acquired fingerprint images often show important variations, with poor contrast in some areas and gaps in ridge and valley regions. These occur due to the very process of scanning a finger. The finger's surface is not flat. Consequently, there is more pressure on the middle of the finger than the edges, giving better contrast in the centre relative to the edges [202]. This results in background variation for different regions of the same image that disturb the ridge and valley contrast. Since the ridge and valley pattern is identified by its gray-level profile, this effect may adversely affect the performance of the subsequent fingerprint recognition algorithms.

Fingerprint image normalisation and enhancement is, therefore, essential pre-processing before minutiae extraction. General-purpose noise-reduction techniques, including local averaging, were not found to be as successful as expected. Their failure could well be attributed to the non-stationary nature [59] of a fingerprint surface. Filtering at its most abstract level can be considered as applying some prior knowledge to improve the signal-to-noise ratio. For images, a classic prior is a smoothness, which implies the application of a low-pass filter to smooth the image. However, a fingerprint has a regular texture with well-defined local orientation and frequency. To exploit this prior, techniques emerged for fingerprint enhancement that include local neighbourhood information during noise filtering. Notable among them is the Gabor filter [60], parameterized with orientation and frequency. The even symmetric two-dimensional Gabor filter kernel is given by:

$$G\left(x, y; \theta, f, \sigma_x, \sigma_y\right) = \mathrm{e}^{-\left(\frac{x_\theta^2}{2\sigma_x^2}+\frac{y_\theta^2}{2\sigma_y^2}\right)} \cos\left(2\pi f y_\theta\right) \tag{7.1}$$

where

$$x_\theta = x\cos\theta + y\sin\theta$$

$$y_\theta = y\cos\theta - x\sin\theta,$$

θ is the orientation of the ridges and f is the local ridge frequency. x_θ and y_θ respectively are the rotated coordinates with fixed σ_x and σ_y as standard deviations parallel and perpendicular to the ridges respectively. The Gabor filter is a band-pass filter with centre frequency f, and band-width defined by σ_x and σ_y.

Hong [60] introduced Gabor filtering to enhance fingerprint images and provided a systematic approach to set its parameters in a local neighbourhood. There are several works extended from [205], and most of these focused on the enhancement of performance. For example, Yang [63] introduced a modified Gabor filter (MGF) resulting in better fingerprint verification.

Unfortunately, Gabor filtering has a high computational complexity. In [206], a set of 8-fixed orientation based separable 2-D Gabor filters is introduced to reduce the computational complexity of conventional Gabor filters used in [60]. Nevertheless, their enhanced image quality was only marginally inferior to the traditional 2-D Gabor filter. Watchareeruetai [205] proposed a generalised separable Gabor filter for any orientation based on Hong's work, resulting in lower computational complexity, better enhancement, and less memory space for the Gabor filtering process. They kept the frequency constant because the frequency does not change much throughout the fingerprint image.

One key limitation of Gabor filtering is that it assumes a given ridge frequency. Although this is true for most of the image, it is not true around minutiae points. Consequently, in maintaining the regular texture of the ridges and valleys, the minutiae points are distorted, and this can affect the accuracy of matching. For fingerprint images, the exact estimation of ridge frequency is a very difficult and time-consuming task, especially in noisy regions and in regions where minutiae and singular points exist. In fact, these singularities, where the curvature is large, do not have any frequency and a wrong assumption can create spurious ridge structures.

To avoid ridge frequency dependence altogether, the prior for fingerprints becomes limited to its high value of anisotropy, which can be processed by using an anisotropic filter. The pioneering work was done by Perona and Malik [64]. They suggested employing the heat equation in a heterogeneous medium for edge enhancement. The scheme allowed both steering and scaling of an anisotropic Gaussian. However, the number of basis filters is large, and the basis filters are nonseparable, increasing the high computational cost. Geusebroek [207] proposed decomposing the anisotropic Gaussian into two Gaussian line filters in non-orthogonal directions. Choosing the axis to decompose the filter along turns out to be extremely efficient from a computing perspective. In a practical setting, not knowing the axis of orientation for each pixel poses a problem. Therefore, a large number of filters are usually applied at different scales and orientations, and the maximum response per pixel over all the filters is accumulated.

Applying a large number of filters commonly requires a significant amount of computing resources. The best way to achieve good real-time performance is to implement it in hardware utilising the parallel processing. Although several efficient FPGA implementations have been presented in the literature for separable as well as non-separable filters, research on the oriented filter implementation on an FPGA is limited. Fons [186, 208], Qin [184] and Gracia [209] used conventional Gabor filters for FPGA-based fingerprint image enhancement. Qin used a 16×16 blocks for Gabor filter implementation. Fons used a directional 7× 7 Gabor filter. For image normalisation, both Qin and Fons used a two-pass local mean and a local variance-based normalisation method.

For large windows, several decompositions can be used, for example, [210] approximates a large circularly symmetric filter by octagons. Joginipelly [211] proposed an efficient implementation of an oriented Gaussian smoother on an FPGA. They decomposed the 2-D filter into 1-D filters and then used pipelining to obtain higher throughput. Their implementation only has a single orientation; for multiple orientations, they require multiple filters in parallel. This limits the applicability of these filters for fingerprint image enhancement.

In this chapter, a decomposition of the anisotropic Gaussian is used in which the image is diffused by first applying a small two-dimensional isotropic Gaussian filter followed by a relatively large anisotropic Gaussian line filter aligned to the ridge direction. We extended the work of Joginipelly by being able to change the orientation of the final line Gaussian on a pixel by pixel basis. To implement a real-time system, these algorithms are efficiently enhanced for fixed-point representation and optimised for memory and computational capacity. The significance, or the innovation, of this work, is a novel architecture to steer the orientation of an anisotropic Gaussian filter on a pixel by pixel basis. We also implement a fingerprint image-enhancement algorithm on an FPGA, which is well suited for real-time systems. The proposed implementation is faster and consumes fewer resources than existing methods.

The rest of the chapter is organized as follows. The proposed algorithm is described in Section 7.3. In Section 7.4, the FPGA implementation of the proposed structure is presented. The experimental results, and performance with several datasets, of the proposed method are illustrated in Section 7.5. In Section 7.6, we make some conclusions and suggest future work.

7.3 Proposed Method

7.3.1 Image Normalisation

In fingerprints, the input images that are obtained from sensors may have imperfections or poor quality due to non-uniformity. The accuracy of fingerprint recognition can be improved considerably by normalising the image for background variations and contrast before filtering. In the literature, most researchers implement an adaptive normalisation algorithm based on local mean and variance [184,186]. This type of normalisation requires two processing passes of the whole image [186]. Therefore, a frame buffer is required to perform this task. To eliminate multiple image passes, a new algorithm is proposed, which is not only well suited for hardware implementation but also gives much better results than the local-property-based normalisation. The block diagram of the proposed algorithm is shown in Fig. 7.2.

A Gaussian filter with a window size of 3.5σ is used, where the chosen $\sigma = 5$ is the average ridge width of the fingerprint. The application of the filter results in blending the dominating structure with the background and results in a blurred image that contains the slowly-varying illumination pattern. Then, the difference D between the image f and the background estimation image f_B is calculated for every pixel

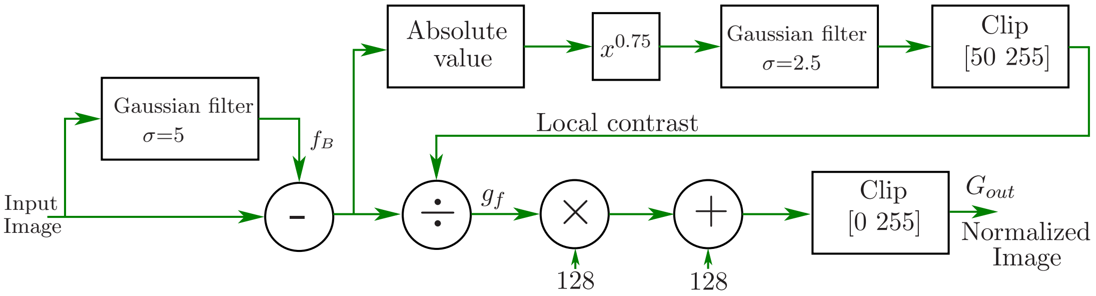


Figure 7.2: *Block diagram of the proposed local-normalisation algorithm.*

$$D(x, y) = f(x, y) - f_B(x, y) \tag{7.2}$$

The background subtraction process is effectively a high-pass filter that allows the structure of interest to pass. Though the filter provides us with a uniform background image, its contrast can vary significantly throughout the image. Therefore, contrast enhancement has to be performed next to normalise object intensities about the background. The magnitude is used to estimate the local contrast. Then a power-law transformation (with $\gamma = 0.75$) compresses the high-contrast pixels about those with low contrast. Then another Gaussian filter, with $\sigma = 2.5$, is applied on the power-law transformed image to average that locally. The resultant image is clipped in between [50-255] to avoid over-enhancing noise and to retain the relative strength of already high-contrast regions. This provides a measure of the local contrast within the image. The difference image is normalised by this local contrast. The result of this division is scaled by a factor of 128 and offset by 128 to enable negative values to be elevated for accommodation in the dynamic range of the monitor. In the end, the image is clipped to the allowed pixel range [0-255] to give the normalised output image.

7.3.2 Orientation Estimation

The patterns present in a fingerprint image demonstrate strong local directionality that has to be taken into account while filtering the image for enhancement. The local ridge orientations $\theta_{x,y}$ are to be estimated from the neighbourhood. In the literature, many researchers propose different techniques to get robust orientation estimation [70, 81, 106]. Although these techniques give excellent orientation estimation they are iterative, making them less suited to hardware. In this chapter, the procedure outlined in [212] was adopted for this purpose. First, discrete derivatives G_x and G_y in x and y directions are calculated by employing a Gaussian smoothed kernel, with a small standard deviation to mitigate noise. Then, covariance matrix data for the fingerprint image was calculated for each pixel as $G_{xx} = G_x^2$, $G_{xy} = G_x G_y$, and $G_{yy} = G_y^2$. The covariance matrix elements were further smoothed with a Gaussian having $\sigma = 1$. Since the gradient vectors on each side of a ridge are opposite to each other, if we smooth the orientation by taking the average of gradient angles directly in a local block, the opposite gradients would cancel each other. To solve this problem, Kass and Witkin [118] proposed a simple and clever idea of doubling the gradient angles before averaging. In practice, 2θ are the angles of squared gradient vectors and are related to the covariance matrix elements by [212]:

$$sin\left(2\theta\right) = \frac{2G_{xy}}{\sqrt{4G_{xy}^2 + \left(G_{xx} - G_{yy}\right)^2}} \tag{7.3}$$

$$cos\left(2\theta\right) = \frac{G_{xx} - G_{yy}}{\sqrt{4G_{xy}^2 + \left(G_{xx} - G_{yy}\right)^2}} \tag{7.4}$$

These doubled angles are smoothed with a Gaussian of $\sigma = 7$. Finally, the orientation is estimated by

$$\theta = \frac{\pi}{2} + \frac{\arctan\left(\frac{sin(2\theta)}{cos(2\theta)}\right)}{2} \tag{7.5}$$

The visual inspection of the estimated orientations provide a good match with local ridge directions as depicted in Fig. 7.3.

7.3.3 Separable Gaussian filter

In this chapter, an anisotropic directional Gaussian filter is used to enhance the ridge structure and reduce noise. Its kernel is given by

$$G_{dir}\left(x, y; \theta, f, \sigma_x, \sigma_y\right) = e^{-\left(\frac{x_\theta^2}{2\sigma_x^2} + \frac{y_\theta^2}{2\sigma_y^2}\right)} \tag{7.6}$$

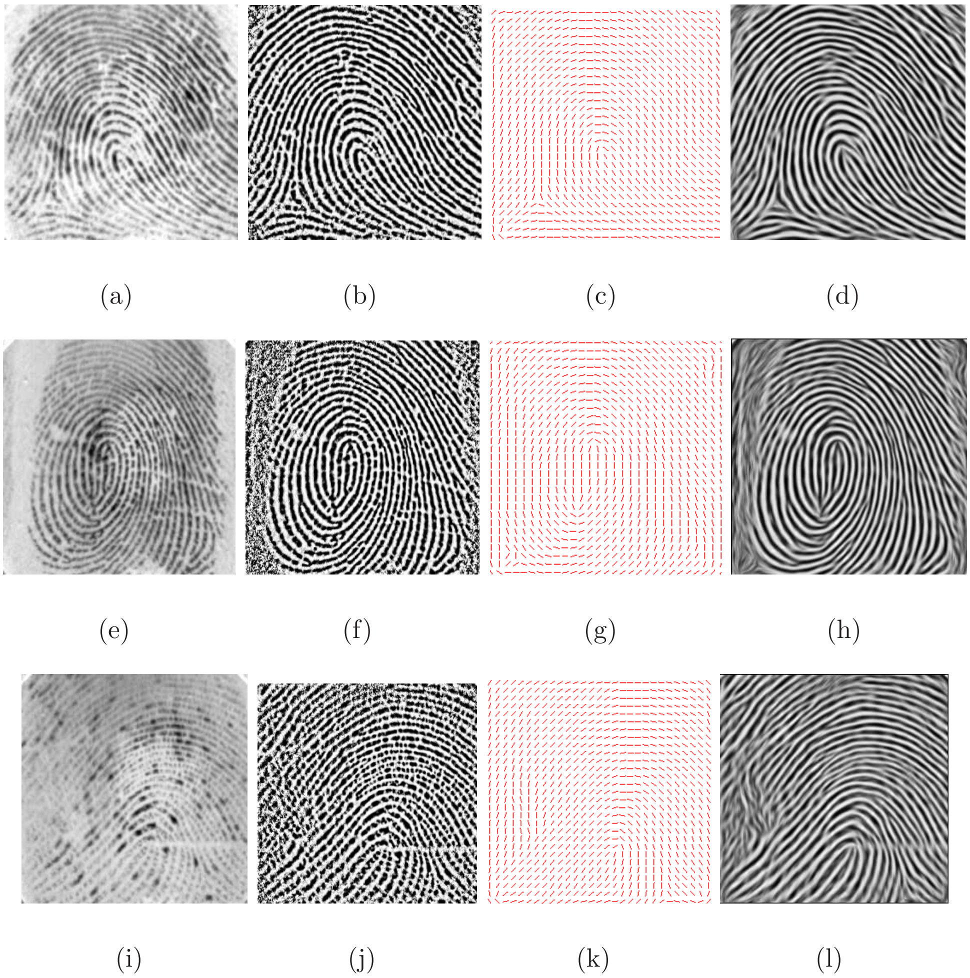
To make the implementation process simpler, G_{dir} is decomposed into two filters. Since $\sigma_y^2 \ll \sigma_x^2$, the filter can be decomposed into a small isotropic filter

$$G_{iso}\left(y; \sigma_y\right) = e^{-\frac{x^2+y^2}{2\sigma_y^2}} \tag{7.7}$$

and an anisotropic filter

$$G_{ani}\left(x_\theta; \theta, \sigma_\theta\right) = e^{-\frac{x_\theta^2}{2\sigma_\theta^2}} \tag{7.8}$$

where $\sigma_\theta^2 = \sigma_x^2 - \sigma_y^2 \approx \sigma_x^2$ and $x_\theta = x\cos\theta + y\sin\theta$. The oriented Gaussian filter is implemented by first convolving by an isotropic 2-D Gaussian filter of size σ_y. The resulting image is then convolved with a 1-D Gaussian in the θ direction. Allowing θ to vary from



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

Figure 7.3: *Impact of proposed proposed method on low contrast fingerprint images. Column 1 shows three fingerprint images . Columns 2 show the output of proposed local normalisation on those images. Column 3 show the ridge orientation map and column 4 show the result of guided line Gaussian filter on noisy images.*

pixel to pixel gives a steerable Gaussian filter, which yields an anisotropically smoothed image.

7.4 FPGA Implementation

Real-time image processing systems are hard to design using a software. The reason is that a large data set is required to represent an image, and complex operations are needed to perform a certain task. Consider that a video rate of 60 frames per second, a single operation performed on every pixel of a 640×480 colour image (VGA) equates to 18.4 million operations per second. Thus, the alternative is to make use of hardware design, by prototyping it on an FPGA. FPGAs offer a compromise between the flexibility of general-purpose processors and the hardware-based speed of an integrated-circuit design. One of the main advantages of using FPGAs for the implementation of image processing applications is that their structure can exploit spatial and temporal parallelism. FPGA implementations have the potential to be parallel using a mixture of these two forms. For example, the FPGA could be configured to partition the image and distribute the resulting sections to multiple pipelines, all of which could process data concurrently. Such parallelization is subject to the processing mode and hardware constraints of the system. Converting the software design to an efficient hardware design is one of the most difficult steps in embedded system design.

In software, usually one operation is performed at a time and its result is stored in RAM for the next operation. This is the reason why it takes a longer time to perform certain tasks which comprise multiple sequential operations, while in hardware these components can be combined to create parallel computing structures [197]. Almost all image-processing algorithms contain operations that execute in sequence. This is a form of temporal parallelism [197]. Hence, this structure is ideal to have a separate processor for each operation, as a pipelined architecture, as shown in Fig. 7.4. When processing images, data can usually begin to be output from an operation long before the complete image has been processed by that operation. The time between when data is first input to an operation and the corresponding output is available is the latency of that operation. When each operation only uses input pixel values from a small, local neighbourhood then its latency is lowest. This is because each output only requires data from a few input pixel values. Operation pipelining can give significant performance improvements when all of the operations have low latency because a downstream processor may begin performing its operation before the upstream processors have completed.

The proposed algorithm for fingerprint image enhancement contains different operations that execute in sequence. Temporal parallelism creates a different processor for each operation. In general, each processor performs an operation on an M×N window. Each clock cycle, a new pixel is input and processing begins for the current window position. When processing is complete, after some latency, an output pixel is produced each clock cycle. In this way, scanning a window through the image is equivalent to streaming the image through a processor, one pixel per clock cycle. Processing a window requires more than one-pixel input. Pixels from the previous M-1 columns and N-1 rows are cached using row buffers as described in [197]. One problem with filtering is managing what happens when the window is not completely within the input image [197]. Very few papers

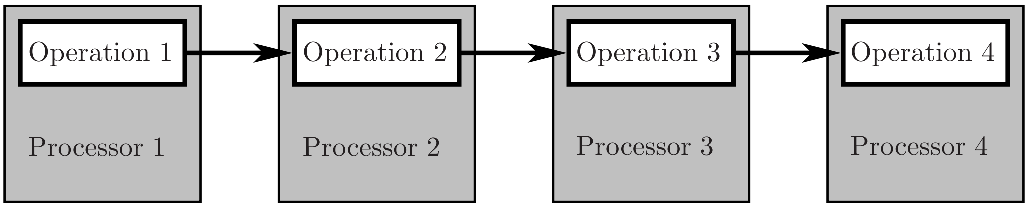


Figure 7.4: *Temporal parallelism exploited by using a processor pipeline.*

consider these boundary conditions since the design to handle them properly can take more effort than to manage the normal case where all of the data is available. In this chapter, to tackle these boundary conditions border pixel duplication is used.

Fig. 7.5 shows the block diagram of the proposed hardware structure, where an input image stream is normalised and enhanced directly as it is streamed to the output. In the proposed structure, each processor or block is designed carefully so that the minimum latency can be achieved and the overall system can get full advantage of the pipelined parallel computing structures.

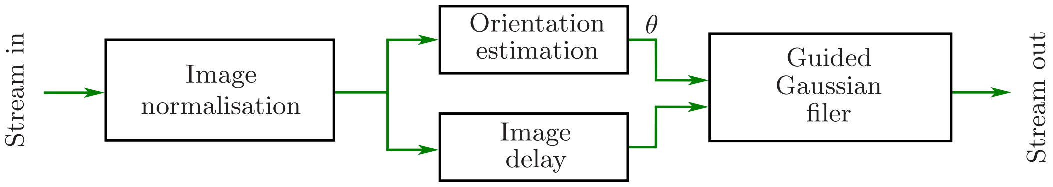


Figure 7.5: *Block diagram of the proposed hardware for fingerprint image enhancement.*

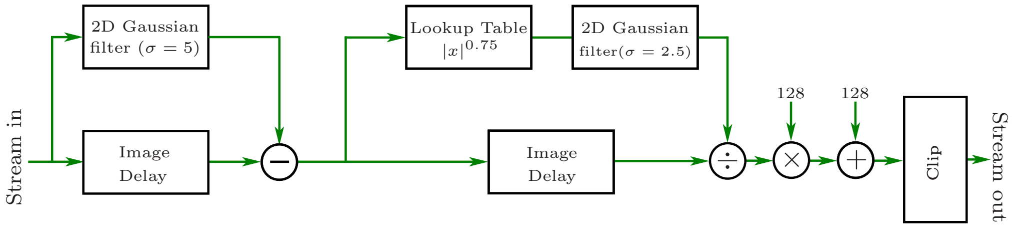

Stream in
2D Gaussian filter ($\sigma = 5$)
Image Delay
Lookup Table $|x|^{0.75}$
2D Gaussian filter($\sigma = 2.5$)
Image Delay
128
128
Clip
Stream out

Figure 7.6: *Block diagram of the proposed local normalisation algorithm for hardware implementation.*

7.4.1 Image Normalisation

This step is typically carried out to reduce the variation in the grey scale image without changing the image structure or texture information. Fig. 7.6 shows the block diagram of the proposed local normalisation algorithm hardware. The input image is first smoothed through a large 2-D Gaussian filter of $\sigma = 5$. The 2D Gaussian filter is implemented as a cascade of one-dimensional Gaussian filters (1×19 and 19×1). The details of the Gaussian filter implementation are given in the next subsection. Then the output of this Gaussian filter is subtracted from the delayed image. Appropriate X and Y delays are used with respect to the size of the Gaussian filter to get the delayed image. The power-law transformation is performed on the absolute value of the difference image. The power-law transformation is implemented in hardware by using a lookup table. This power-law transformed image is smoothed by a second Gaussian filter of $\sigma = 2.5$. The delayed image is divided by the output of the Gaussian filter. The resultant image is rescaled by a factor of 128, offset by 128, and clipped to give the normalised image.

Gaussian Filter Implementation

A Gaussian filter is used for image blurring and removing noise or high-frequency components of an image. In two dimensions, the Gaussian filter kernel is:

$$G\left(x, y; \sigma\right) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{7.9}$$

For large σ, the size of the filter increases significantly, which makes a hardware implementation expensive. For the hardware implementation, a size of $\sigma = 5$ is used with a 19×19 mask. Truncating the window limits the stopband attenuation of the Gaussian filter at high frequencies to 50 dB.

As the Gaussian is separable, this allows the filter to be implemented as a cascade of one-dimensional Gaussian filters (1×19 and 19×1. Although the filter can be decomposed to only use adders [156,197], the need for such a decomposition is less important on modern FPGAs, where high-speed pipelined multipliers are plentiful. The symmetrical nature of the Gaussian filter allows the number of multipliers to be halved by folding the data path, and performing the addition before the multiplication, as shown in Fig. 7.8

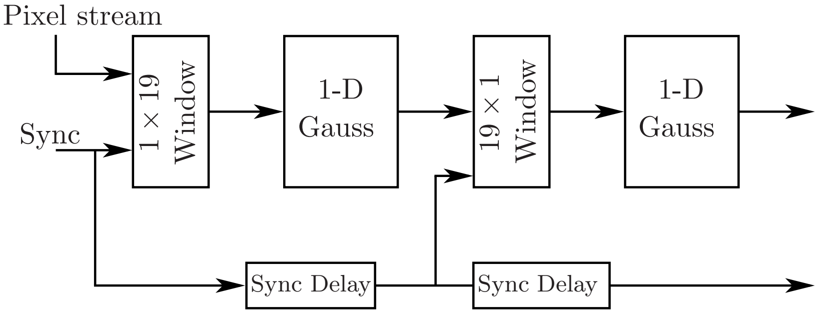


Figure 7.7: *Block diagram of a 2D Gaussian filter implementation.*

7.4.2 Orientation Estimation

Reliable orientation-field (OF) estimation plays a vital role in fingerprint image enhancement. It is one of the most important preprocessing steps. The performance of the proposed guided Gaussian filter is dependent on the orientation field as well as two additional tuning parameters: σ_x and σ_y the standard deviations of the Gaussian envelope. For this purpose, the non-iterative gradient-based method [212] described in section II B is implemented in hardware. The block diagram of the proposed hardware is shown in Fig. 7.9. The input stream data is first smoothed by a Gaussian filter of $\sigma = 1$ (with a 5×5 window) to reduce the noise before calculating the discrete derivatives. The output of the Gaussian filter is then passed through a Sobel filter. This gives the partial derivatives of intensity, G_x and G_y, as outputs, as shown in Fig. 7.10. Using these derivatives the covariance matrix data G_x^2, G_xG_y and G_y^2 is calculated. Another Gaussian filter of $\sigma = 1$ is applied on these covariance matrices and then $sin(2\theta)$ and $cos(2\theta)$ are calculated. A modified CORDIC can be used to calculate both Eq. 7.3 and Eq. 7.4 in one operation (avoiding the need for square roots and divisions), as shown in Fig. 7.11. This efficiently

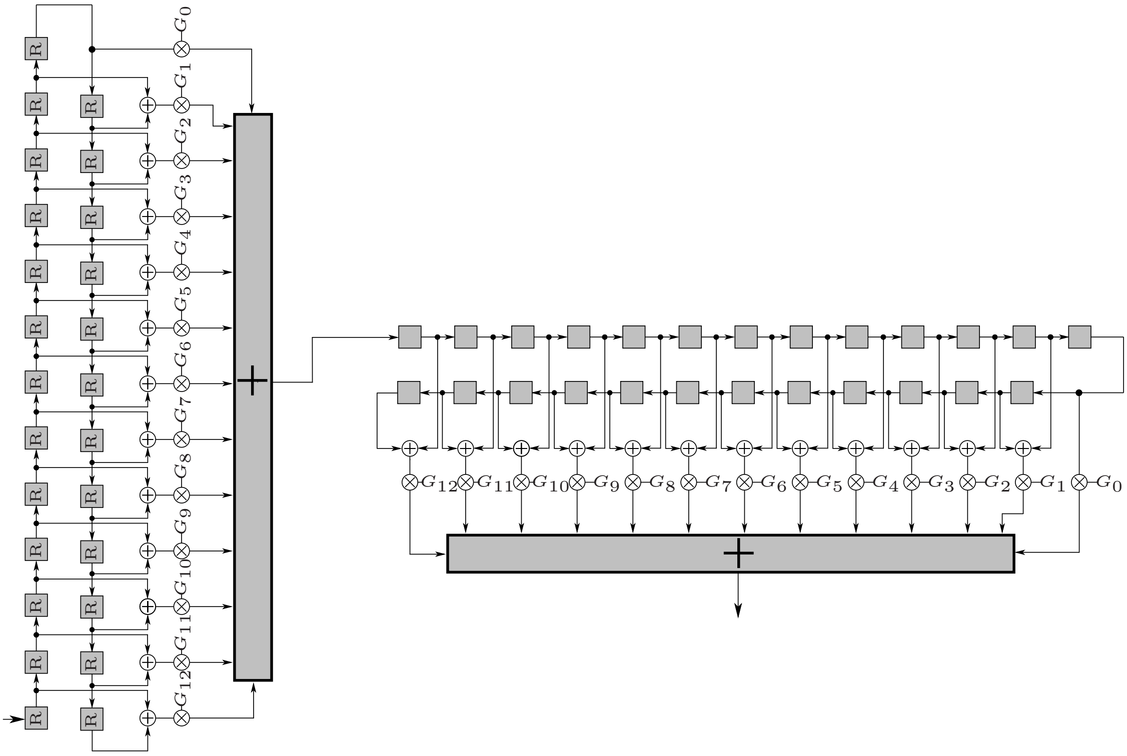


Figure 7.8: *Hardware implementation of a (1 × 25 and 25 × 1) Gaussian filter with $\sigma = 7$. The G_x are filter coefficients. For the vertical filter, the boxes represent row buffers.*

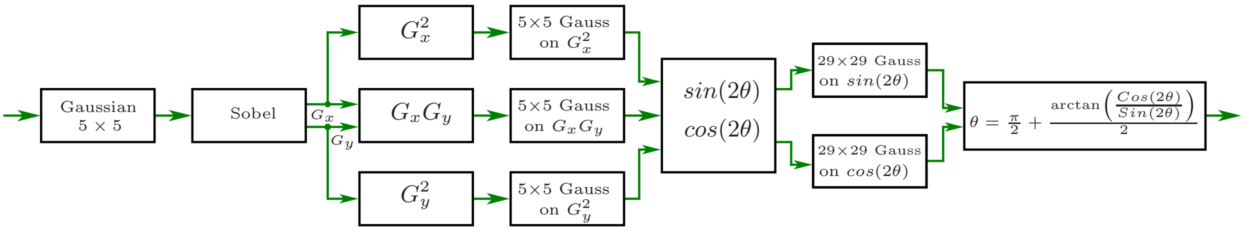
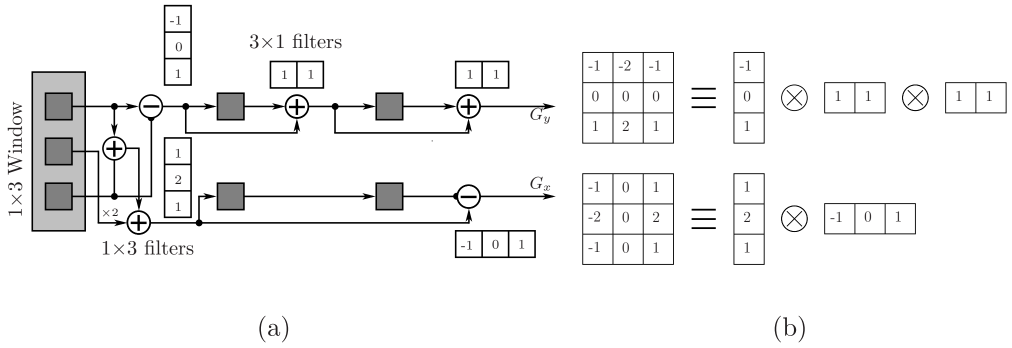

Gaussian 5 × 5
Sobel
G_x
G_y
G_x^2
G_xG_y
G_y^2
5×5 Gauss on G_x^2
5×5 Gauss on G_xG_y
5×5 Gauss on G_y^2
$sin(2\theta)$
$cos(2\theta)$
29×29 Gauss on $sin(2\theta)$
29×29 Gauss on $cos(2\theta)$
$\theta = \frac{\pi}{2} + \frac{\arctan\left(\frac{Cos(2\theta)}{Sin(2\theta)}\right)}{2}$

Figure 7.9: *Block diagram of orientation-field estimation hardware implementation.*

uses CORDIC in vectoring mode to determine 2θ combine with a second CORDIC in rotation mode to calculate $sin(2\theta)$ and $cos(2\theta)$. Since the angle determined by first the CORDIC is used to rotate the vector for the second CORDIC, both iterations can be combined, as shown in Fig. 7.12. The double angles are further smoothed by a larger Gaussian of $\sigma = 8$ (using a 29×29 window). The final orientation is estimated by arctangent of these double angles. For this purpose, a CORDIC unit using the vectoring mode calculates the arctangent. Fig. 7.13 shows the block diagram of the unrolled CORDIC iteration. The Gaussian filters are decomposed and implemented as a cascade in a similar manner to that shown in Fig. 7.7 and Fig. 7.8.



(a) (b)

Figure 7.10: *Sobel filter implementation*

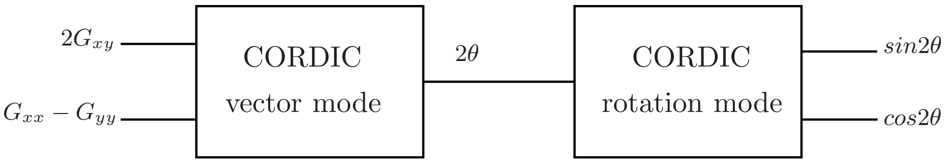


Figure 7.11: *Block diagram of the modified CORDIC for calculating $sin2\theta$ and $cos2\theta$*

.

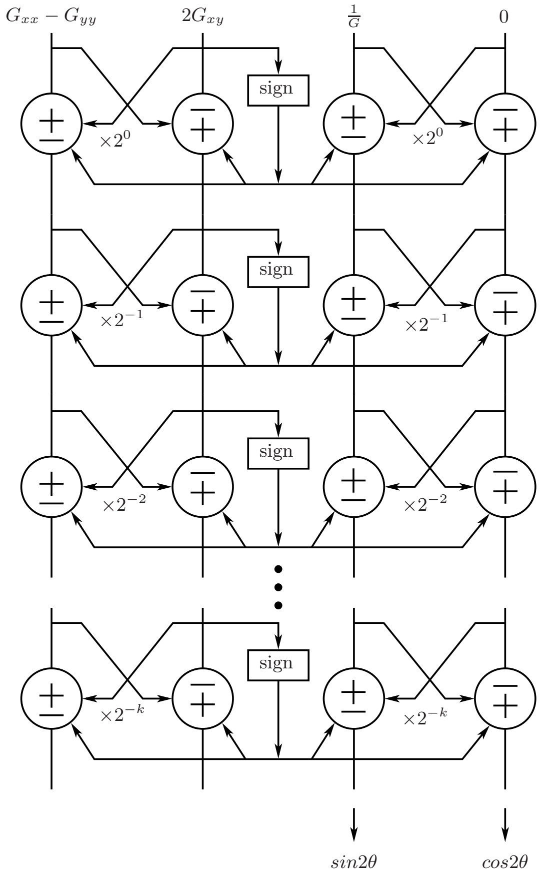


Figure 7.12: *Iterations combined CORDIC for calculating sin2θ and cos2θ.*

7.4.3 Guided-Line Gaussian filter

The aim of this work is to design an efficient algorithm for fingerprint image enhancement that best suits a real-time hardware implementation. For this purpose, the guided Gaussian function is further decomposed into a 2-D isotropic filter and a 1-D an anisotropic filter. For the 2-D filter, $\sigma_y = 0.5$ using a 5×5 window.

7.4.4 Oriented-Line Gaussian Filter Implementation

The next step is to design the hardware for the oriented Gaussian. Usually, $\sigma_x > \sigma_y$ so a larger filter size is required. Fig. 7.15 shows a line Gaussian filter directed at an

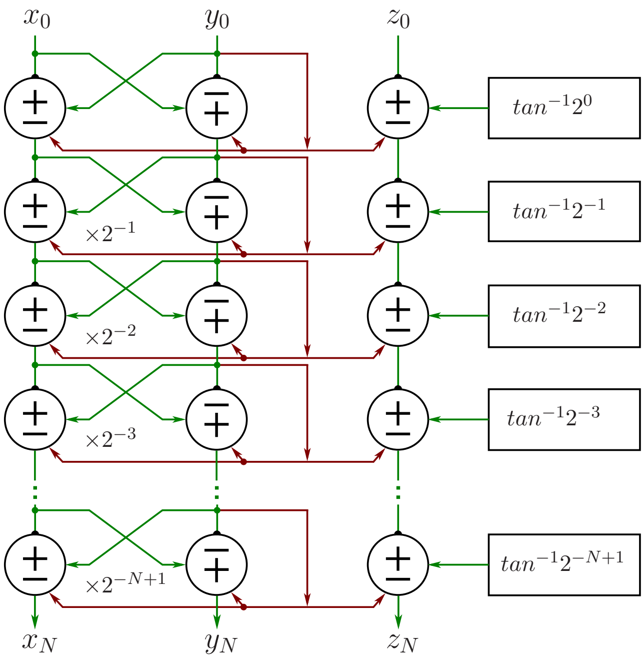


Figure 7.13: *Block diagram of the unrolled CORDIC iteration.*

angle of 45°. A direct 2-D implementation would require a 25 × 25 window where the filter coefficients change every pixel depending on θ. If we analyse Fig. 7.15, it can be observed that out of 625 pixels, only 25 pixels have non-zero values. To implement this oriented-line Gaussian filter efficiently, this filter is converted into a 1-D line Gaussian filter. It is observed that for a 25×25 window if an angle tolerance of 5.625° (180°/32) is used then it does not significantly affect the smoothing performance of the filter. A larger window may require better angle resolution. This tolerance combine with the isotropic

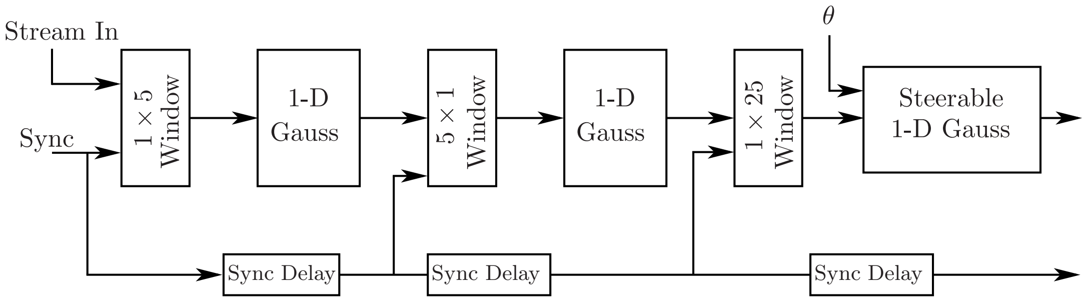


Figure 7.14: *Block diagram of Oriented Gaussian filter implementation.*

Gaussian pre-filter enables nearest-neighbor interpolation to be used to select each pixel from the window to filter. This significantly reduces the hardware complexity.

To convert the 2-D filter into 1-D, the window is divided into two sub-windows hwind and vwind, which are used to filter angles that are primarily horizontal and vertical respectively, as shown in Fig. 7.16. With nearest neighbours, interpolation angles within vwind require one pixel from each row within the window while those in hwind require one from each column. For hwind with streamed image data, the columns within the window arrive in successive clock cycles leading naturally into a transpose filter structure as shown in Fig. 7.18. Multiplexers select the appropriate row for each coefficient depending on the angle corresponding to the centre of the window. For vwind, one pixel is taken from each window row. The orientation is controlled by implementing variable delays for each window row. These allow a different delay to be selected at each clock cycle. The pixels corresponding to the required delays are selected and then multiplied by the Gaussian filter weights. These are summed to get the filter output value. The number of multipliers can be reduced to half in vwind by exploiting symmetry as shown in Fig. 7.19(c). Conceptually the variable delays are implemented using a multiplexer as shown in Fig. 7.19(a). However, to reduce the logic resources, this is implemented using a selected binary delay as shown in Fig. 7.19(b). Finally, either the output of vwind or hvind is selected based on the dominant direction of the oriented Gaussian, as shown in Fig. 7.20.

7.5 Experimental Results

The experiments were performed on the FVC2004 database [58] that consists of four sub-bases, in which images are captured with four different sensors. DB1_A, DB2_A, DB3_A, DB4_A. This database was used to enable comparison with existing algorithms. Each

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	G12
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	G11	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	G10	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	G9	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	G8	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	G7	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	G6	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	G5	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	G4	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	G3	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	G2	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	G1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	G0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	G1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	G2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	G3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	G4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	G5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	G6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	G7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	G8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	G9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	G10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	G11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 7.15: *Line Gaussian filter orientated at* 45°*.*

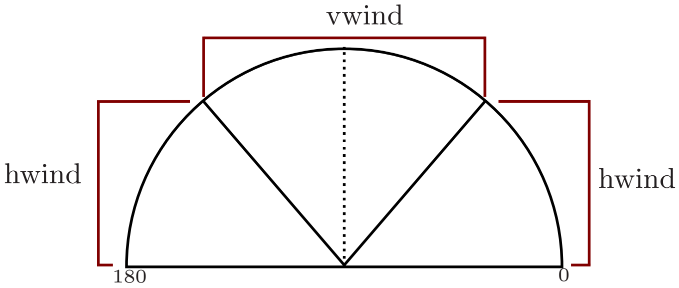
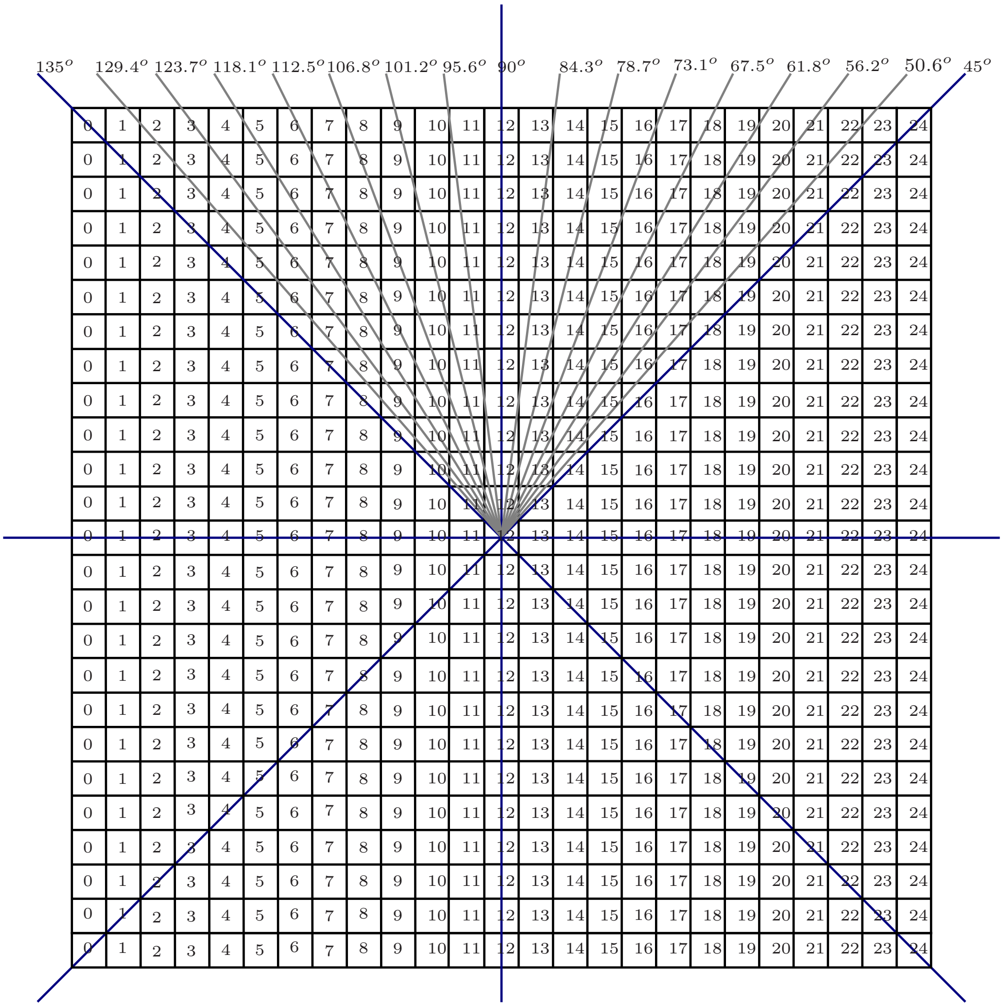


Figure 7.16: *Window selection.*



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

Figure 7.17: *List of lines with rational slopes used to calculate the variable delays of vwind.*

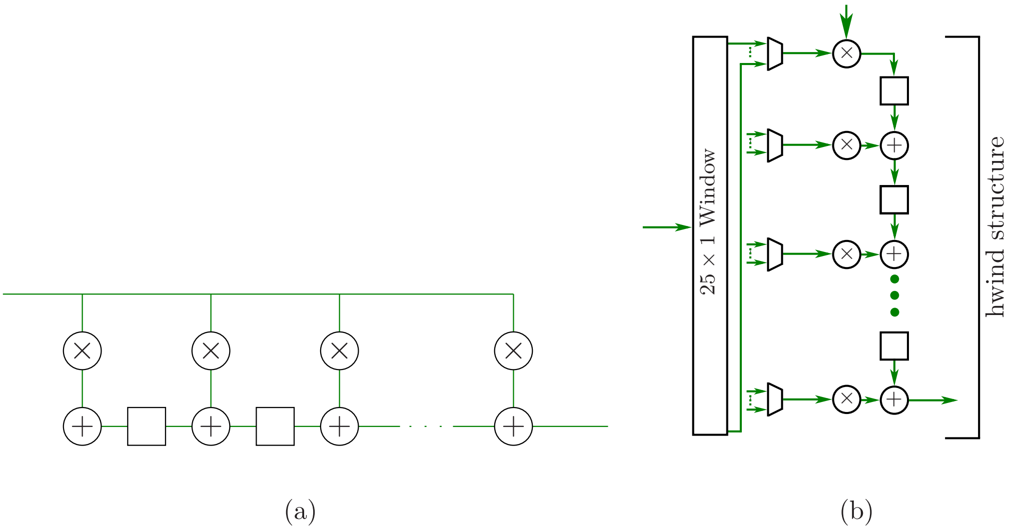


Figure 7.18: *Proposed hwind structure. a) Pipelined transpose filter structure. b) Adapted for steerable filter. The multiplexers select inputs from the appropriate rows depending on the orientation.*

sub-base includes 800 fingerprints, of which 8 are of the same person. Each sub-base has different fingerprint image sizes:

- In sub-bases DB1_A the size of the fingerprint image is 480 × 640
- In sub-bases DB2_A the size of the fingerprint image is 364 × 328
- In sub-bases DB3_A the size of the fingerprint image is 480 × 300
- In sub-bases DB4_A the size of the fingerprint image is 384 × 288

For simulation, the program was written in MATLAB and run on a 3.40 GHz Core i7 processor with 16 Gb of memory. For real-time implementation, VHDL is used and the design is tested on a Cyclone III FPGA using Quartus II, and simulated using ModelSim. To assess the efficiency of the verification system the FAR (False Acceptance Rate) and FRR (False Rejection Rate) are calculated by

$$FAR = \frac{Number\,of\,rejected\,genuine\,claims}{Total\,Number\,of\,genuine\,accesses} \tag{7.10}$$

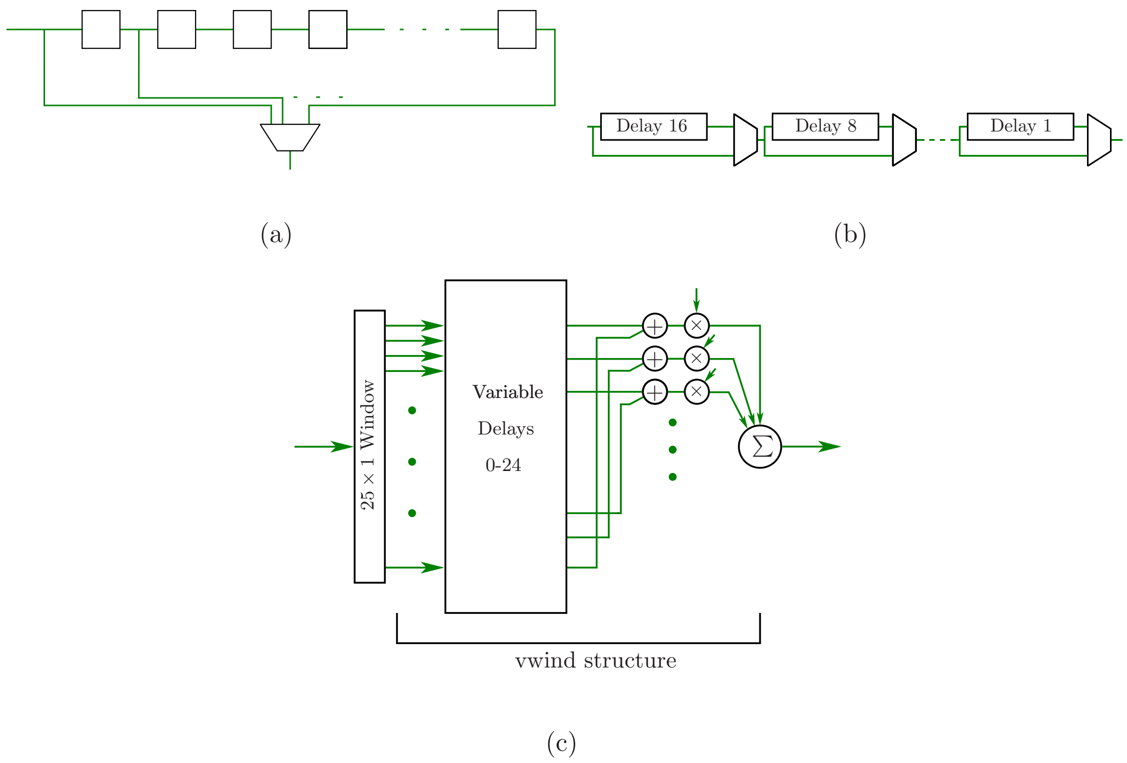


Figure 7.19: *Proposed vwind structure: a) Variable delay selected by a multiplier, b) Variable delay using a selected binary delay, c) structure of vwind using variable delays for each row, and exploiting symmetry the the multiplications.*

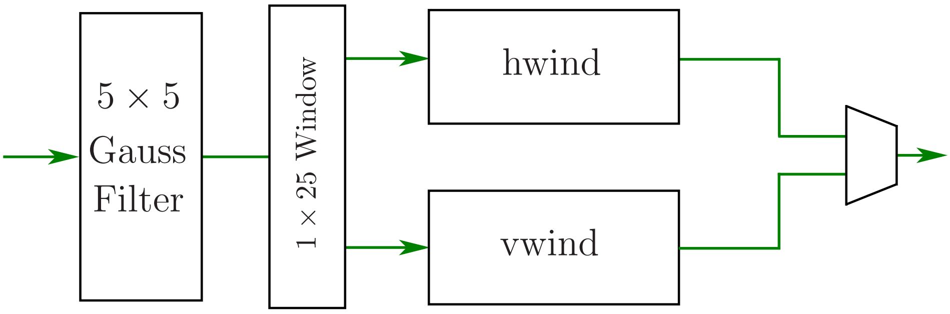


Figure 7.20: *Proposed oriented-line Gaussian filter.*

$$FRR = \frac{Number\, of\, accepted\, imposter\, claims}{Total\, Number\, of\, imposter\, accesses} \tag{7.11}$$

Finally, the EER (Equal Error Rate) was used as a success rate indicator, marking the point where FRR and FAR are equal.

$$EER = \frac{FAR + FRR}{2},\ if\ \ FAR = FRR \tag{7.12}$$

Table 7.1 shows the Equal error rate EER (%) assessment of the proposed algorithm compared with existing algorithms which are similar to the proposed algorithm. Marko [187] proposed a block-local normalisation and normalized the image with 3 different block sizes 2 × 2, 4 × 4 and 8 × 8. He claimed that 8 × 8 block-local normalisation is best suited for real-time application, only if the required accuracy of the system is not too high. For fingerprint image enhancement, both algorithms are used as a pre-processing step with the Yang algorithm [213] and oriented linear diffusion [112]. Although the EER of the proposed algorithm is slightly higher than some of the existing algorithms, it is still very good for an FPGA-based real-time system. Also in this chapter, we put more emphasis on the speed as well as the parallel compute-intensive demands of the fingerprint image-enhancement process with respect to FPGA rather than its EER.

Fig. 7.3 depicts the results of the proposed fingerprint image-enhancement algorithm on a noisy image. In column 1, three noisy fingerprint images are chosen. The proposed local normalisation algorithm is applied to these images. The second column of Fig. 7.3 clearly depicts that the nonuniformity of the poor-quality images is removed in an efficient way. Column 3 shows the orientation map of the local normalised images, calculated by using Eq. 10.3. Finally, column 4 shows the result of an oriented-line Gaussian filter on the noisy images. The second row shows that it is necessary to remove the non-fingerprint background because the noise from enhancing the contrast can result in false ridges and false features. It is also essential to accurately estimate the orientation field because errors can blur the fingerprint pattern, potentially resulting in false minutiae points. In Table

Table 7.1: *Equal error rate EER (%) assessment of proposed algorithm with existing algorithm.*

FVC2004	Compared algorithms	EER (%)
DB1_A	Oriented linear diffusion + normalisation	9.74
	Oriented linear diffusion + 2 × 2 block-local normalisation	9.08
	Oriented linear diffusion + 4 × 4 block-local normalisation	9.93
	Oriented linear diffusion + 8 × 8 block-local normalisation	11.53
	Two stage enhancement (Yang) 2×2 block-local normalisation	7.50
	Two stage enhancement (Yang) 4×4 block-local normalisation	7.17
	Two stage enhancement (Yang) 8×8 block-local normalisation	7.26
	Proposed Method	9.03
DB2_A	Oriented linear diffusion + normalisation	11.02
	Oriented linear diffusion + 2 × 2 block-local normalisation	9.99
	Oriented linear diffusion + 4 × 4 block-local normalisation	11.27
	Oriented linear diffusion + 8 × 8 block-local normalisation	13.62
	Two stage enhancement (Yang) 2×2 block-local normalisation	7.75
	Two stage enhancement (Yang) 4×4 block-local normalisation	7.70
	Two stage enhancement (Yang) 8×8 block-local normalisation	8.97
	Proposed Method	9.71

7.2, the relative processing speed of the proposed FPGA-based implementation and the proposed MATLAB-based algorithm is presented. The proposed FPGA implementation is over 300 times faster than the MATLAB-based implementation on a PC. The reason for the high speed is the efficient use of parallelism in the FPGA.

Table 7.2: *Processing speed of proposed FPGA based algorithm with proposed PC based MATLAB structure*

FVC2004	MATLAB(PC)	FPGA	FPGA vs MATLAB
FVC_DB1_A	4.78	0.0140	340×
FVC_DB2_A	1.86	0.00594	313×
FVC_DB3_A	2.24	0.00712	315×
FVC_DB4_A	1.72	0.00558	308×

7.5.1 Comparison to Other Hardware Architectures

For complete fingerprint image enhancement, the proposed method is compared with two prominent FPGA-based fingerprint image-enhancement algorithms, one proposed by Fons [186] and the other proposed by Qin [184]. In the literature, Fons was the fastest and the most efficient algorithm for fingerprint image enhancement. Table 7.3 compares the processing time of the proposed algorithm with the Fons and Qin algorithms. The proposed FPGA implementation is approximately 9 times faster than the Fons FPGA implementation. There are two reasons for this speed improvement. The first is the use of an oriented line Gaussian filter instead of a Gabor filter used by Fons. Second, our whole algorithm is single-pass while the segmentation and normalisation components of Fons are two-pass. One disadvantage of the two-pass process is that it requires the complete image to be stored off chip, large images cannot be processed completely on an FPGA due to a lack of sufficient on-chip memory. To process a single pixel a two-pass algorithm requires a minimum of 2 clock cycles. If we compare the proposed method with Fons, the proposed method requires fewer registers and logic cells. Table 7.4 gives a detailed hardware resource comparison of the proposed method with Fons [199]. A sample image of size 512×256 is used for this comparison. Again, the proposed normalisation require fewer registers as well as logical cells than the Fons method. We require more memory than the Fons method because of the use of a Gaussian filter twice in the normalisation. However, these figures are a little misleading because our architecture does not require off-chip memory which Fons does (not listed in the table). In terms of performance, the proposed normalisation performance is much better than Fons. For orientation estimation, Fons used Hong's method that does not tackle the noise present in the orientation. In the proposed method, a large Gaussian filter is used to mitigate the effects of noise present in the orientations. Although in terms of resources this is bit costly, as can be seen in Table 7.4, it gives better smoothing than Fons. For the final filtering stage, the proposed method's performance is much better than Fons. It consumes fewer memory bits and logic cells and is almost 7 times faster. This is achieved by using the oriented line Gaussian and its separability property along with efficient hardware implementation using parallelism of streamed data.

7.6 Conclusion and Future Work

In this article, we address several challenging problems for real-time fingerprint image enhancement. A new architecture for anisotropic diffusion is presented, decomposing it into an isotropic and an anisotropic filter. By this decomposition, we manage to efficiently speed up the image-processing time and improve the resource utilisation of the FPGA.

Table 7.3: *Comparison of the proposed method with some existing FPGA based normalisation methods on a sample image of size* 256×512.

Method	Processing time (ms)	Registers	Logical Cells	Memory Bits
Gracia FPGA [209]	522	N/A	N/A	N/A
Qin FPGA [184]	165	N/A	N/A	N/A
Fons FPGA [186]	60	12811	28756	289k
Proposed FPGA	6.9	9590	18160	329k

Table 7.4: *Detailed hardware resources comparison of proposed method with Fons [199] on a sample image of size* 256×512 .

Process	Method	Processing Time (ms)	Registers	Logic Cells	Memory Bit
Normalisation	Fons	25	1384	4729	38912
	Proposed	6.4	917	2286	59522
Orientation Est.	Fons	25	1513	9123	36864
	Proposed	6.6	3335	7597	164864
Filtering	Fons	35	5022	11639	65536
	Proposed	5.4	5334	8268	58464

To further improve the performance of the filter, the input image is homogenised by a new local image normalisation. The proposed normalisation method not only best suits hardware implementation but also gives much better results than the existing methods. Although the EER of the proposed algorithm is higher than some existing algorithms, our main achievement is the speed, as well as the parallel, compute-intensive demands of the fingerprint image-enhancement process with the FPGA rather than its EER. As far as the authors know, the proposed structure is the fastest, most cost-effective and most efficient one as compared to most existing FPGA-based structures. As a future work, the authors aim to develop the remaining stage of an algorithm to reach a complete embedded Automatic Fingerprint Identification System (AFIS). We also plan to investigate the further optimisation of the orientation estimation and anisotropic filtering stages of the algorithm.

Chapter 8

Hardware Implementation Of Fast Pupil Segmentation Using Region Properties[1]

This chapter is an adapted version of a journal article. The section headings from the journal article have been retained. Figures, equations, tables, and references have been renumbered and are in line with the thesis format. This chapter presents a local histogram-based and region properties based automatic pupil segmentation method. The proposed method is efficiently implemented in hardware. With respect to our proposed design, this chapter falls in the iris subsystem. It covers two modules of the iris subsystem, as highlighted in Fig. 8.1.

[1] Published as: Tariq M. Khan, Yinan Kong and Mohammad A. U. Khan“Hardware Implementation Of Fast Pupil Segmentation Using Region Properties,” *Processing of SPIE - The International Society of Optical Engineering*, vol. 9534, pp. 95340F-1-95340F-10, January 2015.

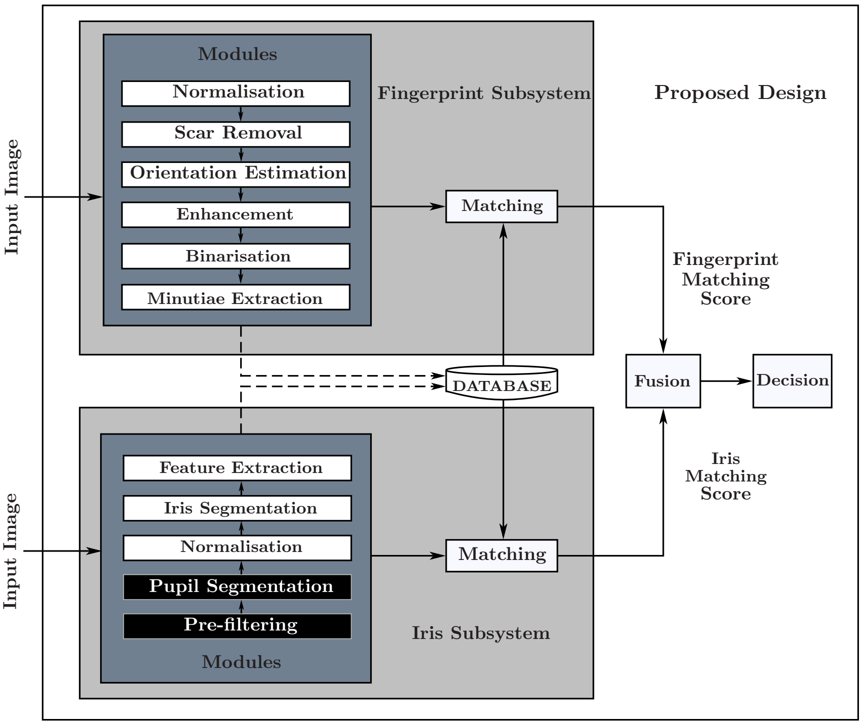


Figure 8.1: *Modules of the proposed design under study*

8.1 Abstract

This chapter presents a novel approach for automatic pupil segmentation. The proposed algorithm uses local histogram-based threshold, area and eccentricity that looks for the region that has the highest probability of having the pupil. The proposed algorithm is implemented on FPGA using a non-iterative scheme along with hardware optimised median filter and connected component logic algorithm. The proposed algorithm is tested on two public databases namely: CASIA v1.0 and MMU v1.0. Experimental results show that the proposed method has satisfying performance and good robustness against the reflection in the pupil.

8.2 Introduction

An automated real-time biometric system such as the fingerprint or iris recognition has been successfully deployed in several large-scale public applications to reduce the identity fraud and increase the reliability and convenience for users. Usually, these systems are implemented on microprocessors working at clock frequency in GHz range. The average execution time to segment an iris code is about 30ms on Intel Pentium 4 at 3.2 GHz with 1 GB RAM [214]. Such software implementations restrict biometric-based applications to specific markets, because of microprocessor cost. On the other hand, the low-cost devices available in the market are too slow for intensive computation. For example, 3162 ms are required to execute an iris recognition algorithm on an ARM922 T at 160 MHz, which is almost 80 times slower than high-performance microprocessors [215]. Field Programmable Gate Array (FPGA) platforms with a large number of embedded memory blocks and registers, parallel DSP blacks and high-speed memory and storage interface provides an attractive solution for these intensive computation based systems.

Given the importance of digital image processing and the significance of their implementations on hardware to achieve better performance, this work addresses implementation pupil segmentation algorithm on FPGA using VHDL language. To achieve high speed with less use of memory, we propose a design and implementation of parallelized pupil localization using an efficient connected component method. The main contribution is the parallel architecture design that utilises on-chip memory controller to achieve our real-time image processing goal. The image is stored in FPGA built-in memory to support the parallel architecture.

8.3 Related Work

It is observed that iris patterns have high dimensionality which makes the recognition decision very much reliable. Generally, an iris recognition system can have four parts [179]:

1. Image Acquisition

2. Segmentation

3. Normalisation and iris code generation

4. Template matching/Recognition

In many machine vision applications, such as iris recognition, pupil tracking [216], ocular torsion measurement [217], pupil size estimation [218], pupillometry [219], point of gaze extraction [220]. The inner boundary location (pupil localization) of an iris is considered as one of the most important pre-processing steps. As performance and accuracy of any pupil-based system depend on pupil localization thus, it is very important to develop an accurate and fast pupil localization method for these systems.

Pupil is nearly a circular region located in the centre of the iris of the eye. Its basic function is to control the amount of light that enters the eye [221]. Pupil absorbs most of the light that enters the eye. Due to this reason, it appears black. Most of the algorithms use two methods to locate the pupil region either by finding its edges using circular mask since pupil is nearly a circular region [148] or by using thresholding as it is the darkest region in an eye image. One of the most common methods for edge detection using circular mask is Hough transform [222]. This method requires extensive computations. Bai *et al.* [223] used a global histogram technique for binarization in order to locate the pupil region. It is an effective method to some extent, but if the grey level of the other part of eye falls below the grey level of pupil region, then it is unable to detect the pupil region correctly. Talal *et al.* [149, 224] introduced a new feedback method for pupil and iris localization which locates the pupil on the basis of adaptive thresholding that looks for the region that has the highest probability of having the pupil in an iterative manner. Khan *et al.* [146] proposed an eccentricity based histogram method to locate the pupil region which overcomes the problem that pupil should always have minimum grey level value. This method is iterative wherein each iteration two regions are constructed by bisection method, then morphological operators are applied on these constructed regions for removing the unwanted noise. This formation makes it computationally intensive approach. Another drawback of this method is in bisection process. If the algorithm picks a wrong side then it can never locate true pupil which ultimately cost more iterations than normal.

Most of the discussed algorithms are computationally expensive, because of their iterative nature. Iterative algorithms require the complete image to be stored on the off-chip memory during processing. The data transfer overhead in transferring the full image to and from off-chips memory makes these algorithms slower. FPGAs typically have an order of magnitude slower clock speed than high-end processors. For improvement, a gain of factor 10 is required. A memory-based algorithm cannot achieve this much gain factor. Also, these algorithms have been optimised for serial implementation, adapting to a parallel implementation is not trivial.

In this chapter, we propose a new method which finds the pupil by eccentricity and area of the regions belongs to three dominating peaks of the histogram of the low grey level region. In order to implement it on FPGA, a parallel technique is used in which region properties of all three regions are calculated simultaneously using an efficient connected component method. This formation makes this technique much faster and more efficient than [146]. Rest of the chapter is organised as. Section 8.4 details the proposed method. In section 8.5 hardware implementation is presented. Experimental results and quality measure are discussed in Section 8.6. Finally, Section 8.7 presents our concluding remarks.

8.4 Proposed Method

There are several issues to be handled for segmenting iris. Firstly, static threshold fails to binarize iris image for varying illumination. Secondly, iris occlusion by eyelids and eyelashes degrades the performance of localization module. Thirdly, during image acquisition, the spot of light creates specular highlights on pupil which adds noise to input and hinder localization. Lastly, the gaze of an individual may not be centred. Such images are usually acquired in a non-cooperative environment. If we use eccentricity based thresholding [146] without any pre-filtering then we cannot get a circular pupil region as shown in Fig. 8.2. In order to mitigate these effects, a non-iterative eccentricity based method is proposed. In the proposed method 7x7 median filter is used as pre-filter to cater the problem of varying illumination. As eyelids and eyelashes badly affect the structure of pupil and it is hard to locate pupil with just eccentricity therefor area and eccentricity are used together for locating the pupil region. The block diagram of the proposed method is given in the Fig. 8.3. The details of the proposed algorithm are given below:

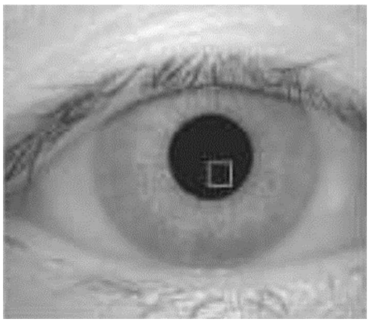


Figure 8.2: *Binarization using adaptive threshold*

Step I: To mitigate the effects of noise, a 5×5 median filter is used which facilitates the thresholding process to construct pupil region as much circular as possible.

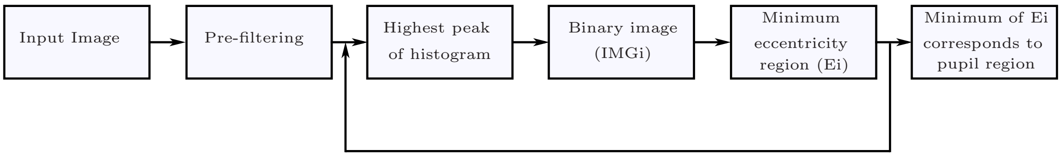

Input Image
Pre-filtering
Highest peak of histogram
Binary image (IMGi)
Minimum eccentricity region (Ei)
Minimum of Ei corresponds to pupil region

Figure 8.3: *Block digram of proposed scheme for pupil localisation*

Also set $i = 1$.

Step II: Histogram of the enhanced image is calculated. As mentioned previously, pupil is a dark region in an eye image and its grey level value always falls below 70 [146]. For further processing, the grey levels above 70 are neglected for CASIA v1.0[175] and above 50 are neglected for MMU v1.0 database [174].

Step III: Highest peak H_p of the histogram is found and a region is constructed by selecting grey levels ranging from $H_p - \varepsilon$ to $H_p + \varepsilon$, where ε is a small real value.

Step IV: Eccentricities and Areas of the entire connected components are calculated and the minimum eccentricity region E_i in pre-set area range is found.

Step V: Set highest peak value in the histogram to zero and increment i by 1.

Step VI: Repeat Step III-IV until $i < 4$.

Step VII: The minimum eccentricity region, in the predefined area bound, corresponds to the pupil region.

8.5 Hardware Implementation

The proposed pupil algorithm contains following two main blocks, as shown in Fig. 8.4:

1. Median Filter

2. Area and eccentricity calculation using connected components

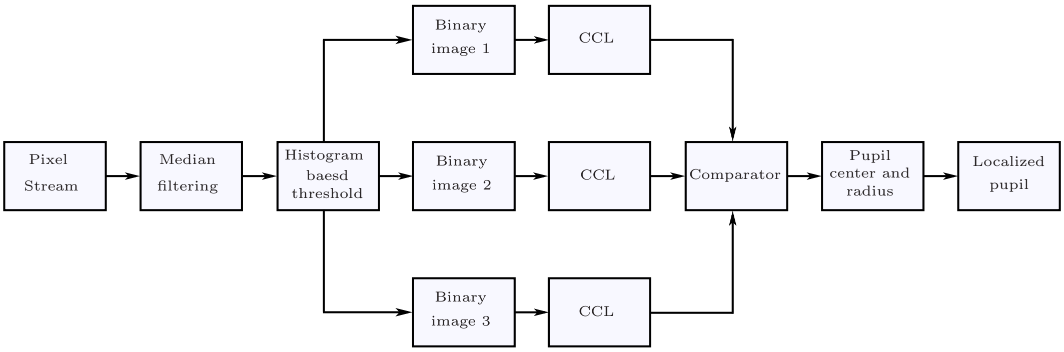


Figure 8.4: *Proposed parallel structure for hardware implementation*

8.5.1 Median Filter

The median filter of a sequence can be found by choosing the middle value of the sorted sequence. Median filtering is usually based on date sorting algorithms, like insertion sort, quick sort, and bubble sort. In literature, several authors used these techniques for implementing median filters on hardware [225–227]. If we use bubble sort to find out a median of a sequence of size $(2N + 1)$ then it requires $N(2N + 1)$ sorting units and $(2N + 1)$ registers. On the other hand, if we increase windows size, the number of compare-and-swap units increases significantly. The authors in [228] used cumulative histogram based implementation which uses 256×245-bits codes saved in 8 Block RAM's (BRAM). Although histogram based method is fast than sorting based methods but it is inefficient because of high memory use. In this chapter, a directional median filter is used [229]. This filter uses only 32×32 bit codes which reduce the BRAM utilisation and also a number of clock cycles. In image processing, to implement a filter on image sliding window is used. For a 5×5 filter, when a window slides over the image old 5 pixels are removed with new 5 pixels. To read the first window 25 clock cycles are needed and 5 clock cycles to update the window. With this formation, median filter overlap with maximum 5 pixels for each direction. For every new window, four new cumulative histograms corresponding to four directions are built. [229] proposed three methods for median filter implementation on FPGA. From Table 8.1, it can be seen that different methods have different advantages in terms slices occupied, LUTs used and the number of clocks. In real time pupil segmentation, less number of clock cycle with the minimum resource are required therefore, we use method 3 as it best suits for our application.

Table 8.1: *Median filters resource utilisation for 200x200 image on Virtex 5 FPGA*

Method	**Slices occupied**	**Slice LUTs used**	**Slice register used**	**Number of clock cycles**
Method 1	28%	12%	20%	24%
Method 2	24%	14%	8%	24%
Method 3	28%	16%	20%	8%

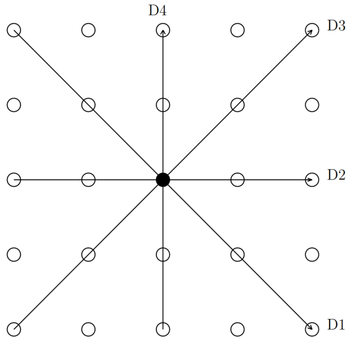


Figure 8.5: *Direction considered for directional median filter implementation*

8.5.2 Area And Eccentricity Calculation Using Connected Component Labeling

To calculate the eccentricity and area of all the regions of the threshold image, the first step is to locate all the regions in the image. For this purpose, connected components labelling algorithms are used. Classic connected components labelling algorithm [230] requires two raster-scans. In the first scan, four neighbours are examined to determine the label of the current pixel. In the second pass, a merge table is used to relabel all pixels within an image. Jablonski and Gorgon [231] have implemented a two-pass algorithm on FPGA. [232] introduced a high-speed parallel algorithm. Although parallel algorithms give considerable speed improvement over the classic one but these are very resource intensive, requiring a large number of essentially identical processor. Some resource efficient iterative algorithms have been implemented on FPGA [233,234] which uses very simple processing, but require multiple passes. D. G. Bailey [235] introduced a single pass algorithm which eliminates the need for a frame buffer and significantly reduces the latency. This removes the need for processing the labelled imaged. In this chapter, a similar single pass algorithm is used for connected component labelling. Fig. 8.6 shows the architecture of the implemented connected component labelling algorithm. The algorithm implemented requires a Line FIFO for intermediate label storage and a Lookup Table (LUT) memory. Control Unit is used to Label the connected components and stores their area and other region properties in the LUT. Finally, Decision Unit finds the minimum eccentricity region in the defined area.

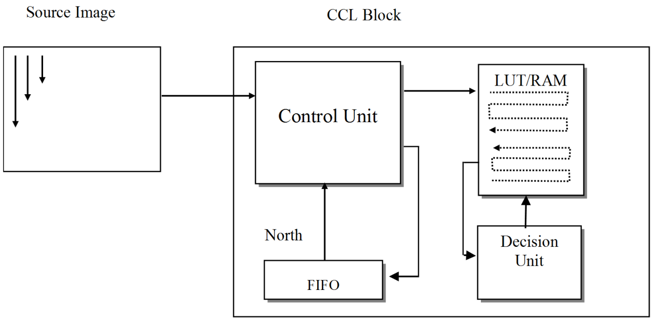


Figure 8.6: *Connected component labeling block architecture*

Control Unit

The data is fed sequentially to the Control Unit. If the input pixel is zero, the pixel label is set to zero. If the pixel value is one, the pixel should be labelled. Fig. 8.7 shows the block diagram of Control Unit. The label is decided on the base of three pixels: Current pixel (Cp), North pixel (Np) and West pixel (Wp) as mentioned in [236]. Fig. 8.8 shows the flowchart of the algorithm used in Control Unit for labelling. In this algorithm, an additional flag will be raised which indicates the need of a merging label. Decision Unit needs to update this merge part to make the implementation optimal.

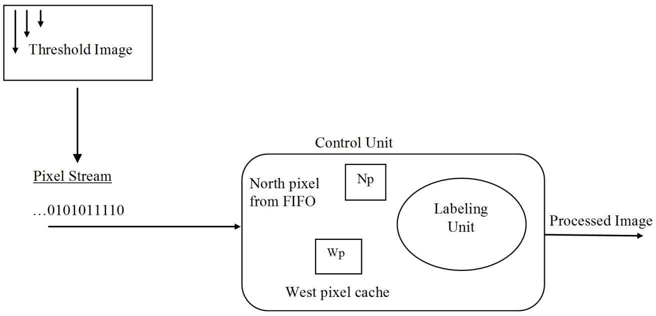


Figure 8.7: *How Control Unit Works*

Decision Unit

Decision Unit checks the LUT in a descending order. First, it checks the merge flag, if it is non-zero then it adds the area value to the appropriate label and also adjusts the Min_x, Min_y, Max_x and Max_y accordingly. If the merge flag is non-zero then it checks the area, if it is in the defined range then it calculates the eccentricity and compares it with the one stored in the cache. If the new eccentricity value is less than the old one then the new one is replaced with old one. This process continues and the final value of the cache represents the lowest eccentricity value in the defined area range. Fig. 8.9 shows the working procedure of Decision Unit.

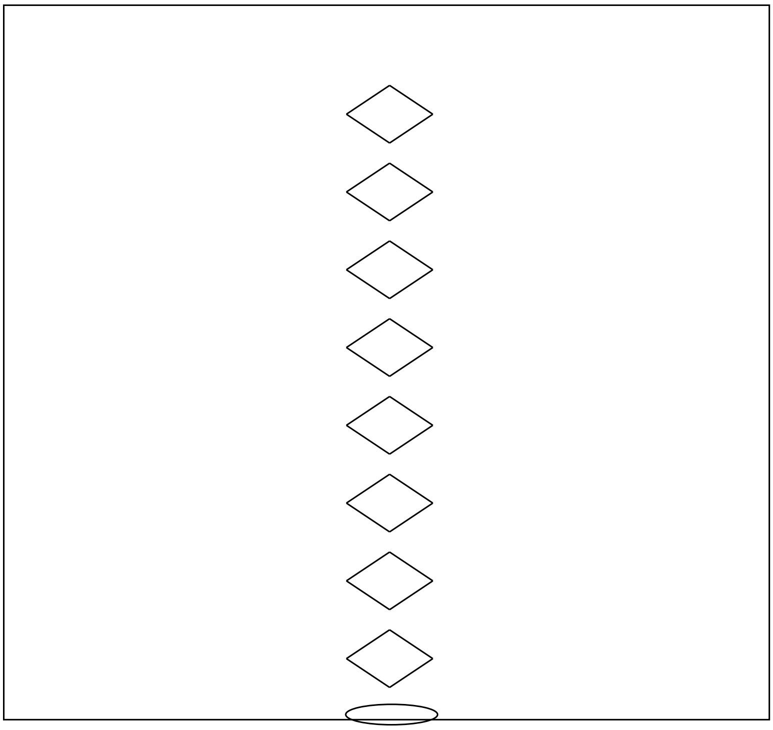
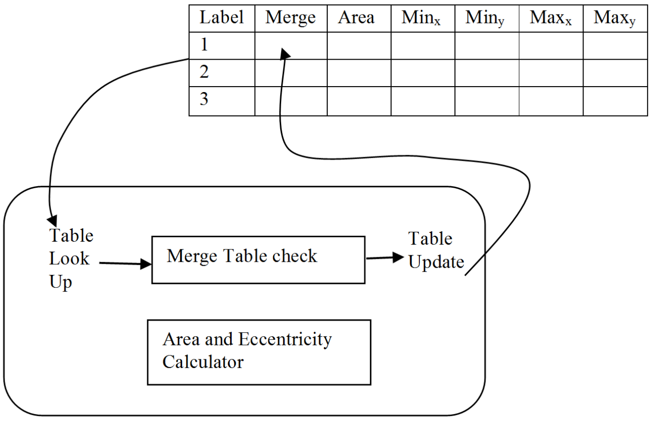


Figure 8.8: *Connected component labeling block flowchart*



Label	Merge	Area	Min_x	Min_y	Max_x	Max_y
1						
2						
3						

Figure 8.9: *How Decision Unit Works*

8.6 Experimental Results

To check the authenticity of the proposed pupil localization method, the MATLAB based algorithm is tested on two most widely used databases mainly MMU v1.0 [174] and CASIA v1.0 [175]. MMU v1.0 database contains 460 images from 46 subjects with 320×240-resolution whereas CAISA v1.0 database contains 756 images of 320×280 resolution. For experiments on CASIA v1.0 database, only those connected components are compared having an area in between 2000-6000 and eccentricity less than 0.7. On the other hand, for MMU v1.0, for area 700-3000 is used and for eccentricity 0.75 is used as a threshold. Pupil localization of MMU v1.0 database is more difficult as compared to CASIA v1.0 because of reflection present in the pupil region shown in Fig. 8.11. The accuracy of the proposed algorithm is 100% on both databases, for the analytical measure, we adapt the same procedure which is used by [224]. In a noisy environment, the bisection method [146] fails to locate the true pupil region. Fig. 8.12(a) shows a noisy image of CASIA-Iris-Twins v4.0 if we apply bisection method [146] then both upper and lower level high peaks refer to the non-pupil region. In such case, if the eccentricity of the upper level is lower than low level then this algorithm fails to locate true pupil region. On the other hand, if we use proposed method then second highest peak can easily locate the true pupil region as shown in Fig. 8.12(d). It is found that the proposed algorithm is much faster than the other region properties based algorithm proposed by [146, 148, 149, 224]. Table 8.2 shows the average speed comparison of proposed method with [146, 224].

8.7 Conclusion

In this chapter, a new method of pupil segmentation based upon local histogram and standard deviation has been proposed. Using the standard deviation and finding the region that has the highest probability of having the pupil region from the local histogram

Table 8.2: *Comparison of Average speed of proposed method with [146, 224] in seconds*

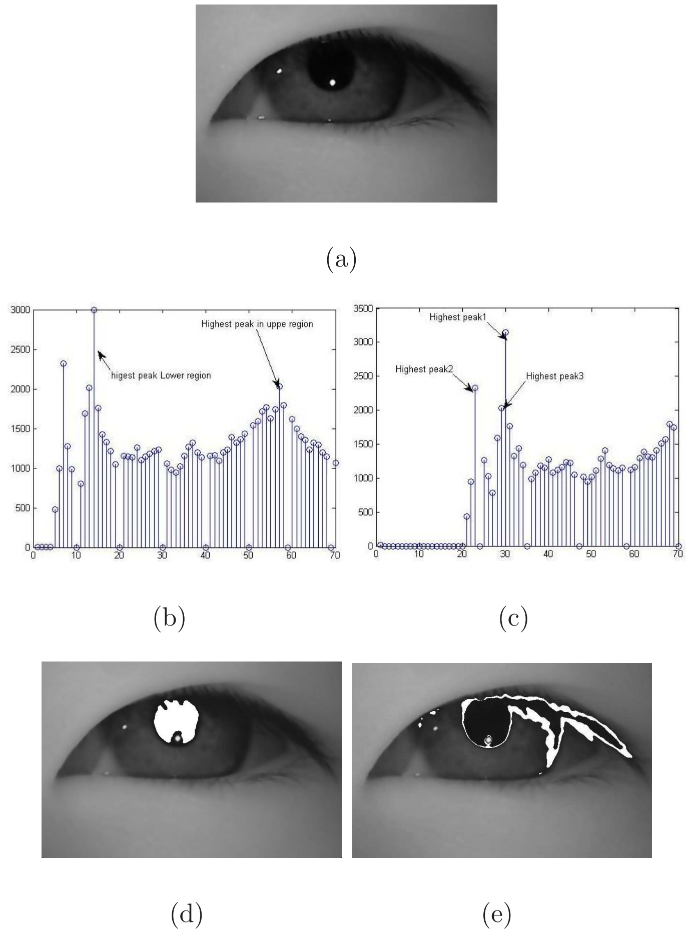
Method	CASIA v1.0	MMUv1.0
M. Talal[224]	0.46	0.43
Khan [146]	0.49	0.45
Proposed	0.145	0.13

Table 8.3: *Comparison of Accuracy Rates on MMU v1.0 database*

Masek [176] as reported in [165]	93.33%
Ma et. al. [153] as reported in [165]	97.87%
Wildes [152] as reported in [165]	98.87%
Daugman [162] as reported in [165]	99.77%
M. Talal[224]	99.77%
Proposed	100%

has overcome the drawbacks faced while locating pupil by using the global histogram. Experiments on MMU v1.0 and CASIA v1.0 iris databases show very satisfactory results. In CASIA v1.0, the pupil area is replaced by a circular region with a constant grey-level that makes it easy for the automatic pupil segmentation.

In the case of MMU v1.0, there is a reflection in the pupil that makes it difficult for the pupil segmentation. From Table 8.1 and Table 8.3, it is clear that the performance of our proposed method is not affected by the reflection in the pupil and it is still capable of segmenting the pupil from the eye image.



(a)

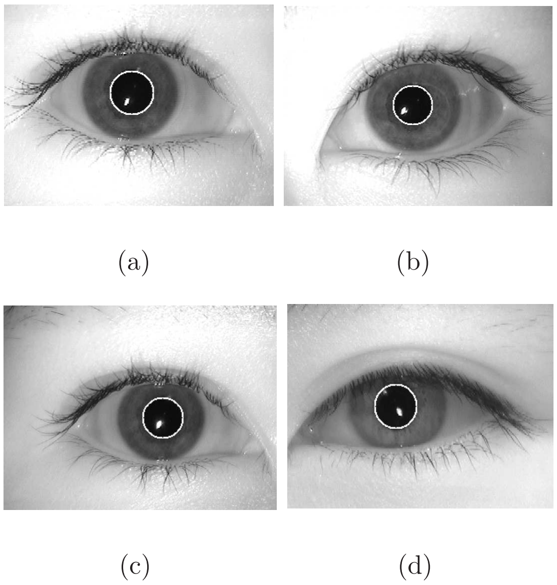
(b)

(c)

(d)

(e)

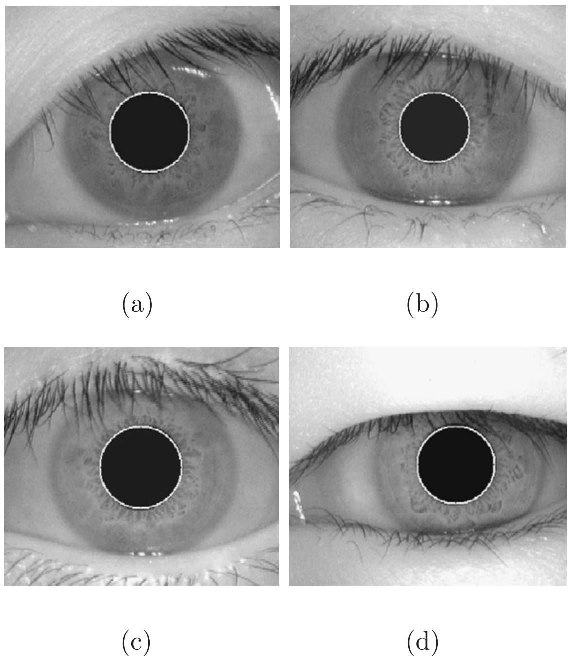
Figure 8.10: *Comparison of bisection method with proposed one on CASIA-Iris-Twins v4.0. (a) Sample Image. (b) Histogram showing two highest peaks, one in an upper region and one in a lower region. This histogram is calculated using algorithm proposed by [146] (c) Histogram of the proposed method which shows three dominating peaks (d) Region constructed by using Highest peak 2 (e) Region constructed by using Highest peak 1*



(a) (b)

(c) (d)

Figure 8.11: *Examples of MMU v1.0 database after applying the proposed method*



(a) (b)

(c) (d)

Figure 8.12: *Examples of CASIA v1.0 database after applying the proposed method*

Chapter 9

Real-time implementation of fast iris segmentation and normalisation on FPGA[1]

This chapter is an adapted version of the journal article. The section headings from the journal article have been retained. Figures, equations, tables, and references have been renumbered and are in line with the thesis format. This chapter presents a fast and efficient implementation of iris segmentation on an FPGA. With respect to our proposed design, this chapter falls in the iris subsystem. It covers four modules of the iris subsystem, as highlighted in Fig. 9.1.

[1] Published as: Tariq M. Khan, D. G. Bailey, Yinan Kong, Mohammad A. U. Khan,"Real-time implementation of fast iris segmentation and normalisation on FPGA," *Journal of Real-Time Image Processing* , submitted

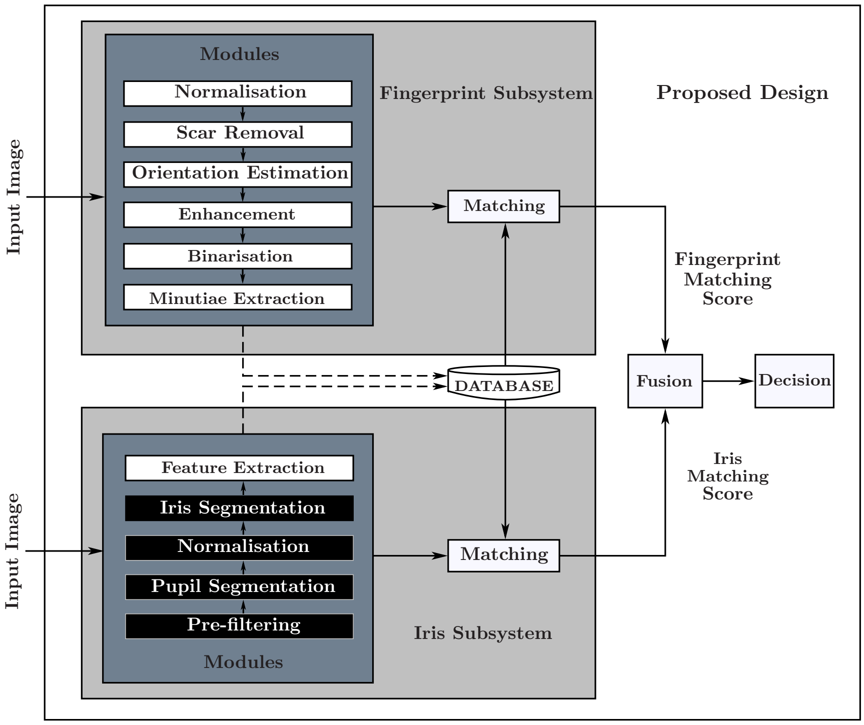


Figure 9.1: *Modules of the proposed design under study*

9.1 Abstract

This chapter presents a real-time iris segmentation technique that results in a faster implementation of iris segmentation on an FPGA. One major hurdle associated with iris segmentation techniques is the use of iterative processes that lead to expensive hardware implementations. To circumvent this, the proposed algorithm uses a threshold of the signed image obtained from the background subtracted image along with morphological operators to localise the pupil. The outer boundary is located by first normalising a selected image region that contains the iris, and then using a first-order gradient operator. The proposed algorithm is implemented on an FPGA using a non-iterative scheme. Three Iris public databases, namely: CASIA-IrisV3-Lamp, CASIA v1.0 and MMU v1.0, are used to test the proposed algorithm. Test results

show an improved speed for its hardware architecture.

9.2 Introduction

Identity fraud and terrorism pose a great threat at the present day, so security is of the utmost concern. Biometrics is a developing technology that provides a higher level of security, convenience, and efficiency than traditional password methods for user authentication. The primary advantage of biometric-based authentication is that it cannot be forgotten, stolen or misplaced. Humans have many biometric features such as a face, hand geometry, fingerprint, voice, and iris. Iris recognition is accurate and reliable due to its high degree of uniqueness and randomness, even between identical twins and remains constantly stable throughout an adult's life [146, 148].

It is observed that iris patterns have a high dimensionality that makes the recognition decision very reliable. A typical iris recognition system has, in general, common steps [179] that start with the acquisition of eye image, segmentation of the iris area, extraction of unique features, matching the features and recognition of a person. Among these essential steps, iris segmentation or boundary localisation plays a significant role in system accuracy because it is the vital step of the whole system. It has been observed that the most computationally intensive task in iris recognition is iris segmentation [146]. Iris segmentation localises two different boundaries; the first is an inner boundary that segments the pupil from the iris, known as the pupillary boundary and the second is the outer boundary of the iris, known as the limbic boundary. These boundaries generally need to be found before filtering to remove any extraneous noise, such as the eyelashes, specular reflections and eyelids in the desired iris region. Duagman proposed an edge-based detector using an integral differential operator (IDO) [162]. The operator searches over the image domain (x, y) for the maximum in the blurred partial derivatives with respect to increasing radius r, of the normalised contour integral of $I(x, y)$. The complete operator behaves as a circular edge detector, blurred at a scale set by σ, searching iteratively for the maximal contour integral. A circular Hough transform is used to detect the inner and outer boundaries of the iris. Three parameters (x_0, y_0, r) are used to define each circle, where (x_0, y_0) is the center and r is the radius of the circle. The circular Hough transform (CHT) was used by Wildes [152] to pinpoint the iris boundaries. However, most segmentation methods execute perfectly well for ideal iris images that are obtained under very controlled settings. In controlled settings, the image acquisition uses a very constrained approach, for example, the subject should use no contact lenses or glasses, and he/she should stand at a fixed distance and stare directly into the camera. On the other hand, less controlled settings lead to non-ideal data [237] in which the image is received from a potentially moving subject at a larger distance. Due to this, many non-ideal problems may arise in the captured image. For example, the image can be blurred, the contrast can be poor, the illumination can be non-uniform, with specular reflections, the image can be off axis or at an angle, and there may be other objects in the iris region, such as eyelids, eyelashes, hair, the frame of glasses, or contact lenses.

Although there has been substantial work on combining different biometrics for a variety of purposes, not much work has focused on the implementation of a biometric system for real-time systems. Because of the limited computational resource and memory space, the currently available embedded devices are not suitable for the real-time implementation of a biometric application system. To implement a real-time system, the biometric algorithms need to be efficiently enhanced within a fixed-point representation and optimised for memory and computational capacity. The significance, or the innovation, of this work, is the implementation of an iris recognition system on an FPGA, that is best suited for real-time systems. To achieve high speed with less memory, a threshold-based method is used that efficiently locates the iris boundaries by using morphological operators and region properties. Our main contribution is a real-time implementation of iris segmentation that fully utilises the parallel and pipelined architecture of an FPGA.

The rest of the chapter is organised as follows: Related worked is discussed in Section 9.3. Section 9.4 details the proposed method. In Section 9.5, a hardware implementation is presented. Experimental results and quality measures are discussed in Section 9.6. Finally, Section 9.7 presents our concluding remarks.

9.3 Related work

There are a lot of different methods being used for iris segmentation, for example edge detectors [238], histogram, active contour models [239] and taking thresholds [146, 149] etc. Methods that involve thresholding and histogram processing are comparatively speedy [146]. It is believed that the threshold and histogram based methods are better suited for FPGA implementation than the more complex methods, that involve the circular Hough transform or active contour models. Ibrahim et al. [149, 224] used histogram-based and standard deviation based adaptive thresholding to localise the pupillary boundary. A circular moving window is used to locate the pupil by finding the range of grey level that has the highest probability of the pupil. The window with the grey level peak having a minimum standard deviation of x- and y- coordinates is selected as the pupil region. Ibrahim didn't use any other considerations before compiling this decision. But the method fails if we apply a standard deviation to a binary object containing some binary objects that cause regions that are low in intensity, for example, hair, eyelashes, and glasses frame. In the same way, Khan et al. [146] used threshold, histogram, and eccentricity to identify the pupillary boundary. This is an iterative method that uses a histogram-based bisection method to construct two regions, and then morphological operators are applied to these constructed regions for removing the unwanted noise. A region with low eccentricity is selected for next iteration. This formation makes it a computationally intensive approach. Another drawback of this method is in the bisection process. If the algorithm picks a wrong side, then it can never locate the true pupil and may ultimately cost more iterations than normal. But, like Ibrahim et al. [146], they also were unsuccessful in proposing any technique to deal with multiple binary objects in a single binary image. Moreover, using eccentricity to localise it can be deceptive as well because, other than the pupil, any small object in an eye image may have a similar eccentricity value.

Most of the discussed algorithms are computationally extensive, making them harder to implement efficiently on an FPGA. In the literature, there are a few FPGA implementations of iris segmentation, but most of them do not support real-time implementation. In [240], a modified form of [146] is proposed that used a histogram-based threshold and region properties for pupil segmentation. As discussed earlier, the use of eccentricity can be deceptive, which is one drawback of this implementation. The use of connected component analysis three times also makes this implementation very costly. Lastly, the image needs to be saved twice, which is the main hurdle for a real-time implementation. When processing streamed images, the bandwidth bottleneck of reading in the image data destroys many of the benefits gained by pipelining the operations. Ngo et al.[241] proposed a real-time iris segmentation by using an edge detector. Instead of using a circular Hough transform, a modified circle detection method is proposed which has three steps: 1) forming the circles with different parameters, 2) reading the pixels that are part of the circles and 3) accumulating the values and computing a percentage match. Although this technique is very efficient and well suited for real-time implementation, it is still a computationally expensive technique.

9.4 Proposed Method

9.4.1 Pupil Segmentation

There are several issues to be handled for segmenting the pupil from an eye image. Firstly, a static threshold fails to capture a binary image under varying illumination. Secondly, occlusion by eyelids and eyelashes degrades the performance of the localisation module. Thirdly, during image acquisition, the spot of light creates specular highlights on the pupil which adds noise to the input and hinders localisation. Lastly, the gaze of an individual may not be centred. Such images are usually acquired in a non-cooperative environment. To mitigate these effects, a non-iterative eccentricity-based method is proposed.

Thresholding using mean subtraction

The acquired images often show important background variations and poor contrast in some areas. One of the reasons of this background variation is the spot of light that creates specular highlights on the pupil and adds noise to the input, as shown in Fig. 9.2(a). To reduce these imperfections, and generate images more suitable for thresholding, preprocessing comprising non-uniform background correction is carried out. The block diagram of the proposed background removal algorithm is shown in Fig. 9.3.

To mitigate the brightness variations in the image, the first step to be carried out is background image subtraction. A Gaussian filter with a window size of $3.5 \times \sigma$ is used, where the chosen $\sigma = 5$. Selection of σ is critical. It depends not only on the size of the image but also on the size of the pupil and iris. The application of the filter results in blending the dominating structure with the background and results in a blurred image that contains the slowly varying illumination pattern. Then the difference D between the image f and the mean image f_B is calculated for every pixel:

$$D\left(x,y\right) = f\left(x,y\right) - f_B\left(x,y\right) \tag{9.1}$$

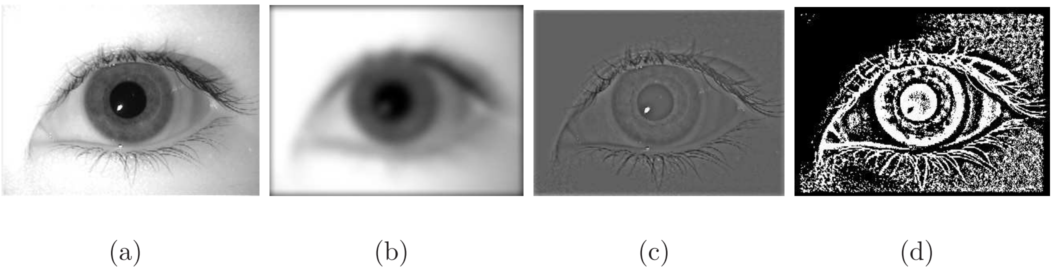


Figure 9.2: *Stages of homogenized threshold process: (a) shows the input sample image. (b) shows the image after applying a Gaussian filter of $\sigma = 5$. (c) shows the difference image. Finally, (d) shows the threshold image.*

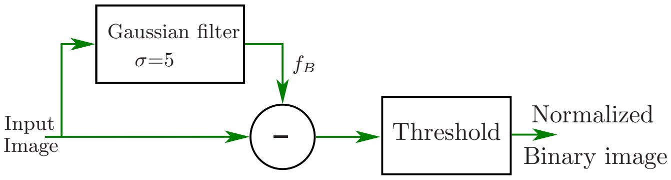


Figure 9.3: *Block diagram of the thresholding using mean substraction.*

This is a high-pass filter that allows the structure of interest to pass. The filter provides us with a uniform background image as shown in Fig.9.2(c). From this background subtracted image, the image is thresholded at 0, effectively giving a sign image, as shown in Fig.9.2(d).

Morphological Operation (Erosion and Dilation)

Morphology is a broad set of image-processing operations that process images based on shapes [242]. In morphological operations, a structuring element is applied to an input image to create an output image of the same size. The shape and size of the structuring element have a great impact on the outcome of a specific morphological operation [242]. Two morphological operators, erosion and dilation, are used to remove the small binary objects from the threshold image. As the pupil is a circular region, therefore a circular structuring element is used. First, an erosion operation is applied on the threshold followed by dilation. A disk of radius 5 is used as a structuring element for erosion whereas a disk of radius 7 is used for dilation.

Isolating pupil by analyzing connected components

This preprocessed image is binary, consisting of some regions against a background. Labelling is used to assign each connected region a unique label, enabling distinct objects to be distinguished. Then each region is processed (based on its label) to extract features of the object represented by the region. The eccentricity of the pupil is nearly equal to zero. To overcome the problem of other small objects having low eccentricity, we use area along with eccentricity to isolate the pupil from other regions. By using area information, regions smaller than 200 pixels are ignored. A region with the lowest eccentricity is selected. The eccentricity is calculated by

$$Ess = \sqrt{\frac{Maj^2 - Min^2}{Maj^2}} \tag{9.2}$$

where Maj and Min are the lengths of the major and minor axes. This ratio for a circular object is zero. For an ellipse, it is between zero and one. Fig. 9.4 (d) shows the pupil isolated using eccentricity and area.

Now the next task is to locate the centre coordinates and the radius of this isolated pupil. By using the boundary box, the boundaries of the pupil region are located. The radius R_{pupil} of the pupil is determined by

$$R_{pupil} = \frac{\max(x) - min(x)}{2}, \tag{9.3}$$

where x is the column coordinates of the detected pupil pixels. The central column C_{col} of the the pupil is calculated as

$$C_{col} = \frac{\max(x) + min(x)}{2} \tag{9.4}$$

In the case of a partially opened eye, where the upper or the lower boundary of the pupil is covered by the eyelids, it is not possible to locate the correct central row of the pupil. Instead of finding max(y) and min(y) for the central row, C_{row} is calculated as

$$C_{row} = \max(x) - R_{pupil}, \tag{9.5}$$

where y is the row coordinates of the detected pupil pixels.

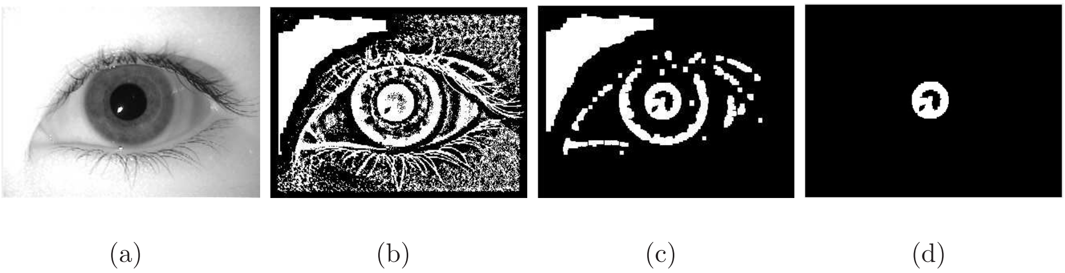


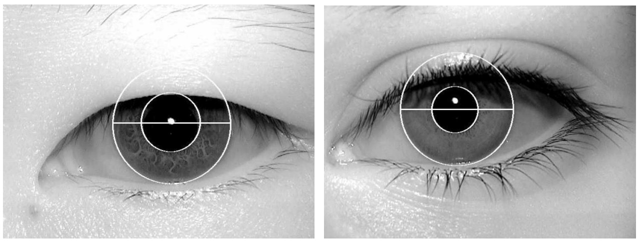
Figure 9.4: *Stages of pupil localization process: (a) the sample input image. (b) thresholded image using mean subtraction. (c) after applying morphological operations on b). (d) the isolated pupil using eccentricity and area.*

9.4.2 Transformation to Polar Coordinates

In iris recognition, iris segmentation involves either localisation of the inner and outer boundaries simultaneously (mostly using edge detection and Hough transform) or location the inner boundary followed by the outer boundary (mostly using histogram or region property based methods). In the literature, almost all researchers transform the iris to polar coordinates after iris segmentation. In region property based techniques [146, 243] for outer boundary localisation, first, a region of interest is defined and then that is converted from rectangular to polar coordinates about the centre of the pupil. Finally, a gradient is used on that polar region to locate the limbic boundary. The image is then resampled between the pupillary and limbic boundaries to give the transformed image. Such a double conversion is challenging in a streamed hardware implementation. Therefore we propose to first transform a selected image region that contains the iris, then the outer boundary of the iris is located from the in polar coordinates image. After conducting a series of experiments on available iris databases (CASIA-3 and MMU) and samples of real-time images taken from a camera, it is found that iris region always lies in a circular region whose radius is within three times the radius of the pupil [146]. All the portion of the eye image beyond this circular region can be ignored [146].

Due to the features of the iris image, the upper and lower parts of the iris image may be influenced by eyelashes and eyelids [243]. From the literature, it is found that usually the upper region is more affected than the lower one [244], as shown in Fig. 9.5. As the iris has a rich texture, just half of the iris pattern can be used for recognition purposes [245].

For transformation, the first image is cropped to a smaller rectangular region that contains the lower part of the iris, as shown in Fig. 9.6(c). Now the image is transformed by assuming the radius of the iris is three times the radius of the pupil. The image is mapped from Cartesian coordinates to polar coordinates by



(a) (b)

Figure 9.5: *Iris sample images: Both show the upper iris region badly affected by eyelashes and eyelids.*

$$I\left(r,\theta\right)=I\left(x\left(r,\theta\right),y\left(r,\theta\right)\right) \tag{9.6}$$

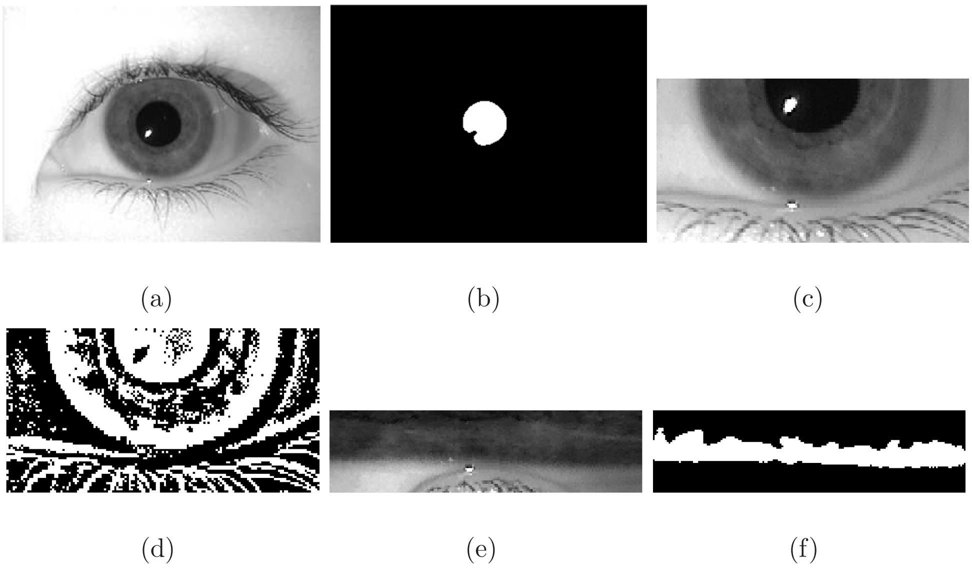


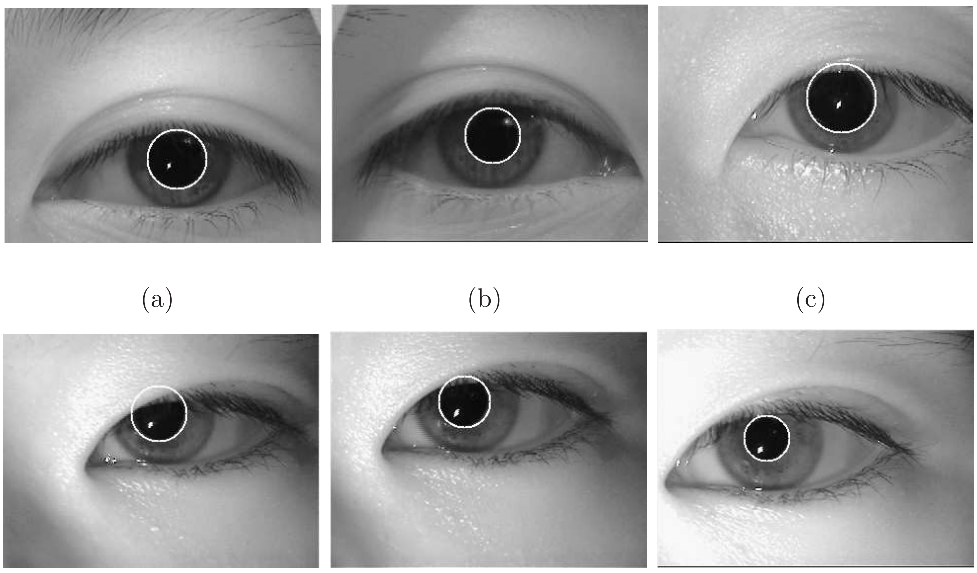
Figure 9.6: *(a) the input sample image. (b) Pupil localized image. (c) Cropped image that contains the lower part of the iris. (d) Threshold cropped image of the lower part of the iris. (e) the transformed image. (f) the thresholded transformed image.*

where $x(r,\theta) = r\cos\theta$, $y(r,\theta) = r\sin\theta$, $R_{pupil} < r < 3 \times R_{pupil}$ and $180° < \theta \leq 360°$. This reverse map converts the output coordinates to get the corresponding pixel in the source image. Bilinear interpolation is used to derive the output pixel value from the four nearest pixels, where (x_i, y_i) are the integer parts of the coordinates, an (x_f, y_f) are the fractional parts.

$$\begin{aligned} I_{yi} &= I[x_i, y_i] + x_f \left(I[x_i+1, y_i] - I[x_i, y_i]\right) \\ I_{yi+1} &= I[x_i, y_i+1] + x_f \left(I[x_i+1, y_i+1] - I[x_i, y_i+1]\right) \\ I[x,y] &= I_{yi} + y_f \left(I_{yi+1} - I_{yi}\right) \end{aligned} \tag{9.7}$$

9.4.3 Iris Segmentation

Once the image is transformed, locating a rectangular outer boundary is much easier than locating the circular boundary. The transformed image usually possesses background illumination which makes the process of finding the true outer boundary difficult. One



(a) (b) (c)

(d) (e) (f)

Figure 9.7: *Six samples of segmented pupil using proposed method*

way to tackle this kind of problem is to homogenise the range of features by a local homogenization method. The problem with local homogenization is that if it tries to enhance the foreground feature area, the background noise is also amplified. Another disadvantage of using homogenization at this stage is that it will make the system more costly. Therefore, instead of finding the boundary of the originally transformed image, the transformed threshold image is used. A first-order gradient is applied vertically to the transformed threshold image, as shown in Fig 9.8. This gradient image is summed up row-wise and the maximum value is picked to give the true outer boundary of the iris. Finally, by using this the true outer boundary original transformed image is updated, as shown in Fig. 9.8 e).

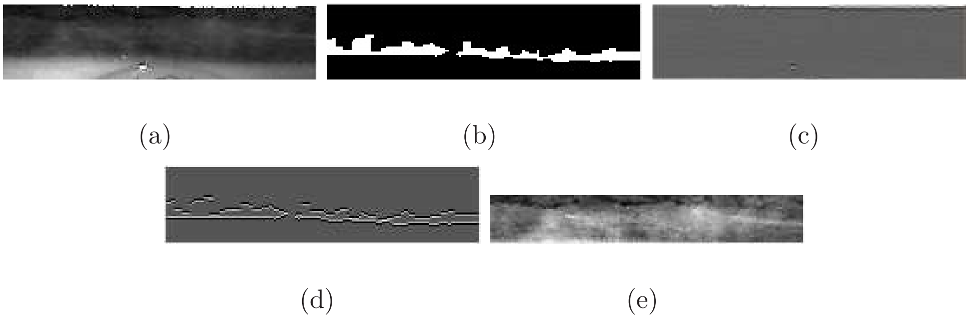


Figure 9.8: *(a) Transformed cropped image. (b) transformed threshold image. (b) Vertical gradient on a). Vertical gradient on b). (e) Localised iris using (d).*

9.5 Hardware implementation for a real-time iris segmentation system

For efficient implementation, it is desirable to minimise the resources used by an algorithm. An important resource is a memory required to buffer any intermediate image and data. While high-end FPGAs have sufficient memory on a chip to hold a complete image, such devices are relatively expensive making them less suited for commodity applications. To minimise the memory required it is, therefore, desirable to perform all of the processing on the image data as it is streamed into the FPGA. This leads naturally to a pipelined implementation, with separate modules for each operation within the chain and processing of pixel-based data at the input data rate. For real-time processing, it is also desirable to reduce the latency between the data input and the classification results output. Our goal is to implement a real-time iris segmentation system that is capable of supporting human identification based on a large database. To do this, we propose a parallel and pipelined architecture for an FPGA-based system. Fig. 9.9 shows the block diagram of the proposed method.

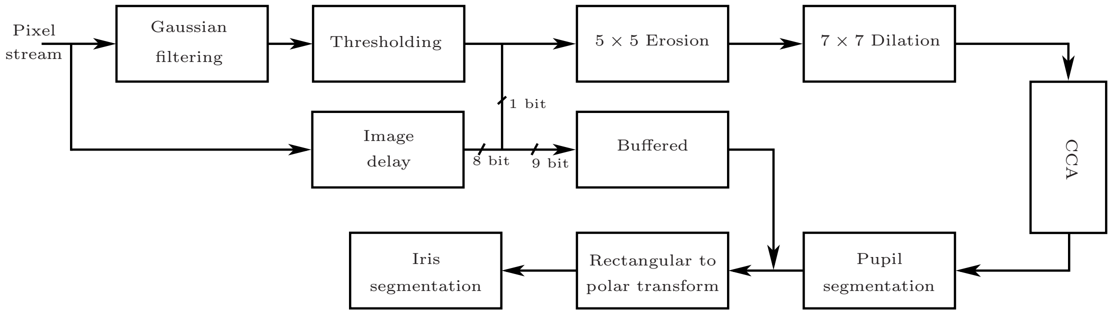


Figure 9.9: *Block diagram of the proposed real-time Iris segmentation*

9.5.1 Pupil Segmentation

We propose a method that extracts the pupil from the streamed data. The hardware structure of the proposed method is described in the following subsections.

Mean subtraction and thresholding

Subtracting the local mean typically carried out to reduce the variation in the gray-scale image without changing the image structure or texture information. Although the mean can be computed efficiently using a rectangular box filter, The poor frequency response can lead to texture artefacts. A Gaussian weighted filter, although more expensive, has better high-frequency characteristics, and overcome these problems. The input image is first smoothed through a 2-D Gaussian filter of $\sigma = 5$. The output of this Gaussian filter is subtracted from the delayed image. Appropriate X and Y delays are used to synchronise the stream with the filtered image. The Y delays can reuse the row buffers of the 1×21 Gaussian filter. The image is thresholded at 0 by keeping only the sign bit.

In two dimensions, the Gaussian function is

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{9.8}$$

For large σ, the size of the filter increases significantly, which makes a hardware implementation expensive. For $\sigma = 5$ truncating the window to 21×21 limits the stopband attenuation at high frequencies to 50 dB.

As the Gaussian is separable, this allows the filter to be implemented as a cascade of one-dimensional Gaussian filters (1×21 and 21×1). Fig. 9.10 shows the implementation of a (1×21 and 21×1) Gaussian filters. Note, symmetry is exploited to reduce the number of multipliers.

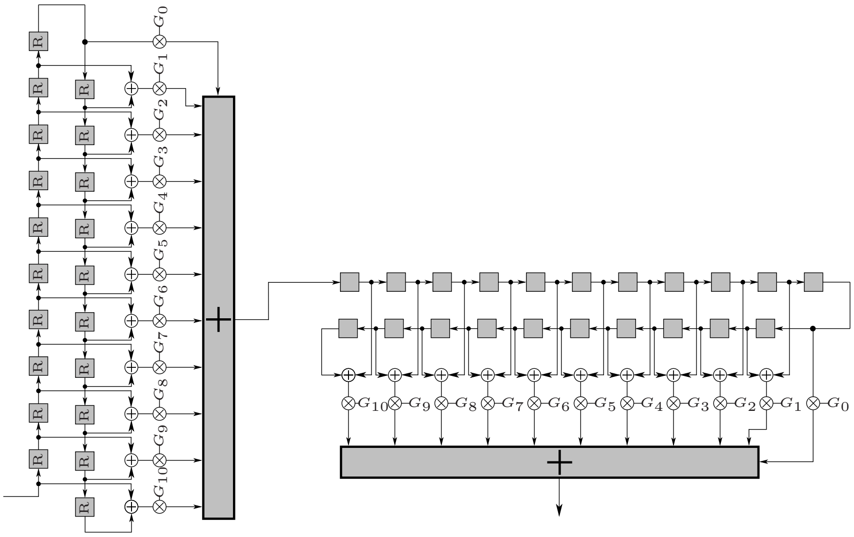


Figure 9.10: *Hardware implementation of a (*1×21 *and* 21×1*) Gaussian filter with* $\sigma = 5$*. The* G_x *are filter coefficients. For the vertical filter, the boxes represent row buffers.*

Morphological Operations (Erosion and Dilation)

Binary morphological filters require simple processing and modest storage [197]. This makes their FPGA implementations easier. In the proposed method, a disc is used as a structuring element for both erosion and dilation. Therefore, for hardware implementation, a non-rectangular window is created using a parallel decomposition [246]. The size and shape of the structuring element determine the number of parallel filters. Fig. 9.12

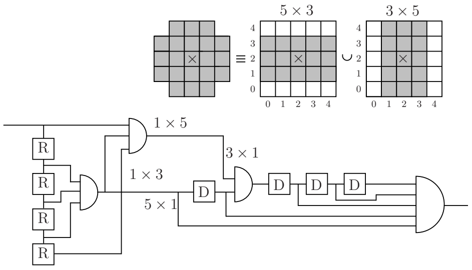


Figure 9.11: *Hardware structure for* 5×5 *circular erosion using parallel decomposition.*

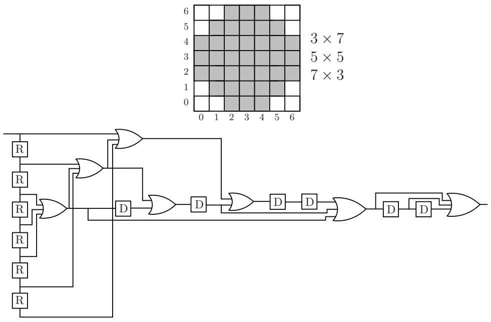


Figure 9.12: *Hardware structure for* 7×7 *circular dilation suing parallel decomposition.*

shows the hardware structure of 5 × 5 erosion with circular structuring element, where R represents a row delay. The parallel decomposition consists of a 5 × 3 and a 3 × 5 filter. To align the two rectangular structuring elements correctly, 3 × 1 and 1 × 3 must be delayed by one column or row respectively. Fig. 9.12 shows the hardware structure of 7 × 7 dilation. The parallel decomposition of this filter consists of a 3 × 7, 5 × 5, and 7 × 3 filter.

Isolating pupil by analyzing connected components

Connected-components labelling is an important step that is used to assign a unique label to a connected group of pixels. By using this unique label, it enables distinct objects to be distinguished. Unfortunately, the classical connected-components labelling algorithm [247] requires two passes through the image, requiring buffering of the intermediate image.

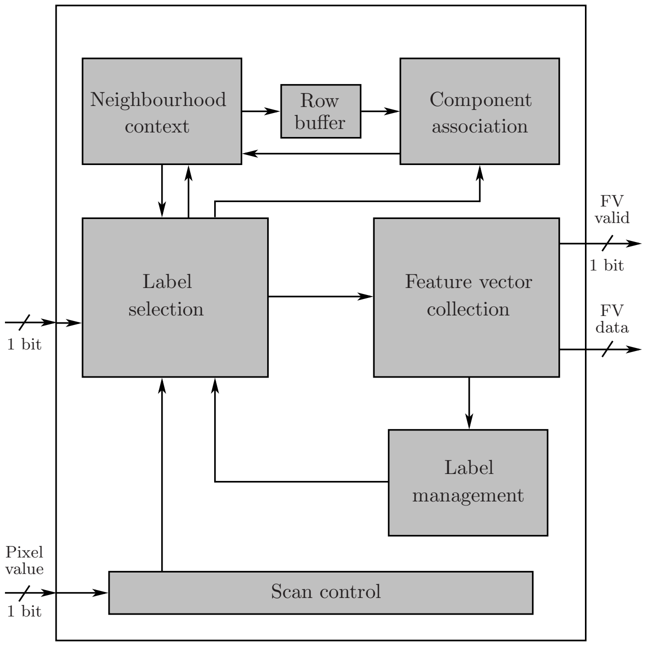


Figure 9.13: *Hardware architecture of connected component analysis*

To overcome these constraints, Klaiber et al [248] proposed a single pass connected component analysis algorithm that is well suited for FPGAs. It is implemented as a window filter, as shown in Fig. 9.13, with the pixel labels shifted along each clock cycle as the window is scanned across the image. The neighbourhood context provides the label of the four pixels adjacent to the current pixel. The labels from the previous row are cached using a row buffer. A label selection block is used that selects the label for the current pixel based on the labels of its neighbours. The eccentricity of the components is calculated based on the second moments of the connected region. Data for calculating the moments ($\sum x, \sum y, \sum x^2, \sum y^2, Area$) and bounding box are directly accumulated as the image is streamed in. The data is merged when components merge, and the final feature data for each connected component is available as soon as it is detected that the component is completed.

9.5.2 Rectangular to polar transformation

After locating the centre and radius of the pupil, the buffered data image is randomly accessed. The region that contains both pupil and the iris is cropped. This region is located by only taking those pixels that fall in the box of length and width $6 \times R$, where R is the radius of the pupil. This cropped region is buffered for the conversion from cartesian to polar coordinates. For transformation, bilinear interpolation is used as given in Eq. 9.7. The direct approach to do bilinear interpolation requires four clock cycles to access four neighbours. In this chater, all four input pixels are accessed within a single clock cycle by using a careful cache design proposed by [197]. For this purpose, the cropped region is buffered in four RAM instead of one with the data partitioned so that odd and even addresses are stored in separate banks enabling all four input pixels to be accessed simultaneously. To control the cache, the expired data is replaced with new data. The image is scanned in successive rows so that when a pixel is not used in a column it is replaced with next corresponding pixel from the frame buffer.

9.5.3 Iris Segmentation

To extract the limbic boundary, AND operation is applied to adjacent rows of the transformed threshold image to detect the edge. The output is summed row-wise and the maximum indicates the true outer boundary of the iris.

9.6 Experimental results

To check the authenticity of the proposed iris localisation method, the software-based design is tested on three widely used databases: MMU v1.0 [174], CASIA v1.0 [175] and CASIA-IrisV3-Lamp iris database. For simulation, the program was written in MATLAB and run on a 3.40 GHz Core i7 processor with 16 Gb memory. The CAISA v1.0 database contains 756 images of 320 × 280 pixel resolution and the MMU v1.0 database contains 460 images from 46 subjects with 320 × 240 pixel resolution. The CASIA-IrisV3-Lamp iris database contains 441 subjects and each subject gives 40 images with a resolution of 640 × 480. For our experiments, only the first 102 subjects are tested. For pupil localisation, on MMU v1.0 and CASIA v1.0, 100% accuracy is achieved. A detailed accuracy comparison of the proposed method with the existing methods on the MMU v1.0 CASIA v1.0 databases is given in Tables 9.1 and 9.2. The accuracy rate on the 102 subjects of the CASIA-IrisV3-Lamp database is 99.3 % which compares well with existing methods as shown in Table 9.3.

In terms of average processing speed, the proposed algorithm is much faster than the other region-properties based algorithm [146, 148, 149, 224]. Table 9.4 compares the average speed of the proposed method with [146, 224].

9.6.1 Comparison to Other Hardware Architectures

For real-time implementation, VHDL is used, and the design is tested on a Cyclone IV FPGA and simulated with ModelSim. For testing, a sample image of size 240 × 320 of MMU database is used. The processing speed of the proposed FPGA-based structure on the three databases is given in Table 9.5. It should be noted that the proposed algorithm is significantly faster on the FPGA than in software. In Table 9.6 the detailed hardware resource utilisation of the proposed fingerprint recognition system is presented. In the literature, Ngo [241] is the fastest and the most efficient algorithm for Iris segmentation. Table 9.8 compares the processing time of the proposed algorithm with the Ngo and some other existing hardware implementations of iris segmentation. The proposed FPGA implementation is 25% faster than the Ngo FPGA implementation. Table 9.7 gives a detailed hardware resource comparison of the proposed method with Ngo. The primary reason for the significant reduction is that in the proposed method, region properties

Table 9.1: *Comparison of Accuracy Rates of the proposed method with existing methods using the MMU v1.0 database*

Masek [176] as reported in [165]	93.33%
Ma et. al. [153] as reported in [165]	97.87%
Wildes [152] as reported in [165]	98.87%
Daugman [162] as reported in [165]	99.77%
Somnath *et al.* [165]	98.41%
M. Talal [224]	99.77%
Proposed	100%

Table 9.2: *Comparison of Accuracy Rates of the proposed method with existing methods using the MMU v1.0 database*

Mateo [177]	95%
Cui [179]	98.60%
Yuan [178]	99.45%
M. Talal[148]	99.90%
Proposed	100%

Table 9.3: *Comparison of Accuracy Rates of the proposed method with existing methods using the CASIA-IrisV3-Lamp database.*

Masek [176] as reported in [165]	79.02%
M. Talal [148]	98.28%
Proposed	99.3%

Table 9.4: *Average processing time comparison of the proposed method with [146, 224] (in seconds)*

Method	CASIA v1.0	MMUv1.0
M. Talal [224]	0.46	0.43
Khan [146]	0.49	0.45
Proposed	0.145	0.13

are efficiently utilised to locate the true boundary of the iris. Ngo used edge detection along with circle detection that makes the implementation costly in terms of resource utilisation. A sample image of size 256×256 is used for this comparison. The proposed method require fewer registers, logical cells and the memory bits than the Ngo method. In terms of performance, the proposed method is better than Ngo.

9.7 Conclusion

In this chapter, a new method of iris segmentation is presented that uses morphological operators along with two region properties, eccentricity and area. The pupil is segmented by thresholding the sign image obtained from the background subtracted image. Then noise is suppressed by using morphological operators, and finally, the pupil region is separated by using two region properties, area and eccentricity. This makes the implementation of the parallel and pipelined architecture of an FPGA easier and more efficient than other existing algorithms with 80% reduction in resources compared with Ngo's implementa-

Table 9.5: *Comparison on average processing time of the proposed FPGA structure with [156] (in milliseconds)*

Database	Proposed	Khan [156]
MMU v1.0	6.5	19
CASIA v1.0	10	27
CASIA-IrisV3-Lamp	26	76

Table 9.6: *Detailed Hardware resource utilisation of the proposed iris segmentation on a low-cost Cyclone IV GX FPGA*

Modules	Available	Pupil Segm.	Normalisation	Iris Segm.
Logic Elements	149760	2910	997	986
Logic register	149760	1174	735	607
Memory bits	6M	712k	364k	336k

tion. To locate the outer boundary of the iris, a different technique is proposed that first transforms a selected image region that contains the iris, to polar coordinates and then the outer boundary of the iris is located from the transformed image. Experiments on MMU v1.0, CASIA v1.0 and CASIA-IrisV3-Lamp iris databases show that the algorithm improves on the current sate of the art results.

Table 9.7: *Performance Comparison with existing implementation for iris segmentation*

Hardware	Method	Comp. Time
DSP [249]	Edge + Hough	$\sim$ 683
FPGA [215]	Edge + Hough	$\sim$ 69.8
FPGA [250]	Edge + Hough	$\sim$ 56.5
FPGA [241]	Edge + Circle	$\sim$ 7.42
Proposed FPGA	Region Property	$\sim$ 5.6

Table 9.8: *Resource Utilization comparison of proposed method with Ngo[241]*

Modules	Ngo[241]	Proposed	Reduction
Logic Elements	223013	4913	97.8%
Logic register	166196	2516	98.5%
Memory bits	7112k	1412k	80%

Chapter 10

Hardware Implementation of Multimodal Biometric using Fingerprint and Iris[1]

This chapter is an adapted version of a journal article. The section headings from the journal article have been retained. Figures, equations, tables, and references have been renumbered and are in line with the thesis format. This chapter presents a fast and efficient implementation of multimodal biometric on an FPGA. For multimodal, two biometric traits fingerprint and iris are used. Some of the optimised structures discussed in previous chapters are used in this chapter. Although this chapter covers all the blocks of our proposed design but there are four main contributions discussed in it, as highlighted in Fig. 10.1.

[1] Published as: Tariq M. Khan, D. G. Bailey, Yinan Kong and Mohammad A. U. Khan,"Hardware Implementation of Multimodal Biometric using Fingerprint and Iris," *Machine Vision and Applications* , Submitted.

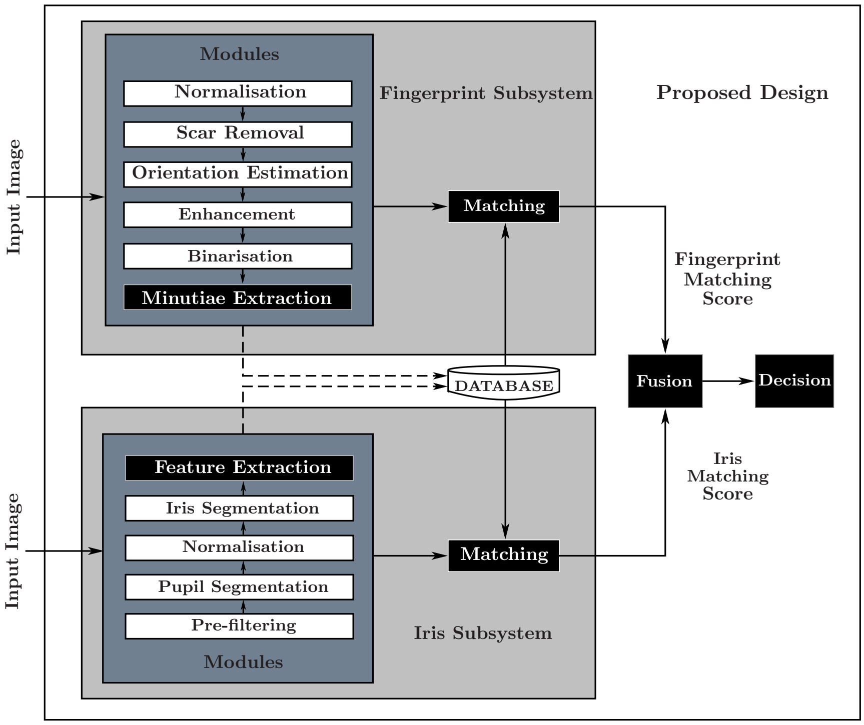


Figure 10.1: *Modules of the proposed design under study*

10.1 Abstract

In this chapter, a hardware architecture of a multimodal biometric system is presented that massively exploits the inherent parallelism. The proposed system is based on multiple biometric fusion that uses two biometric traits, fingerprint and iris. Each biometric trait is first optimised at the software level, by addressing some of the issues that directly affect the FAR and FRR. Then the hardware architectures for both biometric traits are presented, followed by a final multimodal hardware architecture. To the best of the authors knowledge, no other FPGA-based design exists that uses these two traits.

10.2 Introduction

In recent years, biometric authentication is gaining popularity because of its reliability and accuracy over possession-based (e.g. ID card) and knowledge-based (e.g. pass code) authentication methods. Biometric identifiers can not be forgotten, guessed, misplaced or easily copied. However, despite the advantages of biometric authentication, biometric traits face numerous problems. These include inter-class similarity, intra-class variation, spoofing attacks and universality of the trait. Apart from these, it also suffers from an enrolment problem because of noisy data resulting from defective sensors. Environmental variations, signal distortion, background noise, and changes in biometric features can cause inherent variations in the biometric measurements. Therefore, a single biometric trait may not be sufficiently robust.

A multimodal biometric system is introduced to overcome these problems. It uses multiple sensors to acquire biometric traits. This allows: (i) multiple units of the same biometrics (middle and index fingerprints), multiple sensors of same biometrics (Capacitive and Optical fingerprint sensor), (iii) multiple representation and matching of the same biometric (texture-based or minutiae-based fingerprint), (iv) multiple samples of the same biometrics (three templates of left index fingerprint), and (v) multiple biometrics (face and iris or fingerprint and iris). Because of this, a multimodal biometric system is less affected by noise, it overcomes the non-universality problem, it provides a storage security environment and it improves the matching accuracy. Due to these advantages, it has received a considerable amount of attention from researchers.

Most of the existing multimodal biometric systems are computer based. The authentication is performed in an insecure environment that uses the central server for template storage. This can cause a critical information leakage issue. Another disadvantage of a multimodal system is that it requires a large amount of processing as compared to a unimodal biometric system. This makes a multimodal system less suitable for real-time application although, in multimodal biometric, most of the operations are independent. Because of the serial nature of most programming languages, especially the ones used in computers, these can not be performed at the same time. The implementation of a multimodal biometric system in hardware can address these critical problems.

In this chapter, we present the hardware architecture of a multimodal biometric that comprises multiple biometrics (fingerprint and iris). The proposed architecture provides massive exploitation of the inherent parallelism. The rest of the chapter is organised as follows: Related work is discussed in Section 10.3. In Section 10.4, proposed software-based design for fingerprint feature extraction is discussed. A proposed software-based design for iris feature extraction is discussed in Section 10.5. Section 10.6 details the matching and fusion. The hardware implementation of the proposed multimodal biometric system is presented in Section 10.7. Experimental results are discussed in Section 10.8. Finally, Section 10.9 presents our concluding remarks.

10.3 Related Work

From the literature, it is found that only a few multi-modal biometric systems have been implemented as embedded systems. One reason is that a real-time embedded system in a resource-constrained environment poses great challenges, as it possesses limited computational resources and limited memory space. On the other hand, most of the existing multimodal biometric systems are computationally very expensive and are not suitable for real-time implementation. Converting a software design to hardware is one of the most difficult tasks. Therefore it is the least developed, more so with fingerprint and iris multimodal biometrics.

Sonal *et al.* [49] implement a palm-vein identification system in hardware. For hardware implementation a Blackfin ADSP-561 processor is used, whereas the C language is for the algorithms used for matching of palm veins. Template matching and principal component analysis (PCA) are used as verification algorithms for palm-veins and are integrated at a match-score level. Yoo *et al.* [50] have developed two DSP systems for face-fingerprint and iris-fingerprint recognition. In their system, the most computationally expensive tasks are implemented on an FGPA in order to increase the system speed. They used a Xilinx XC3S4000 onboard FPGA and an ARM920T DSP clocked at 400 MHz, and a 128 MB SDRAM. However, no fusion strategy was applied in the embedded biometric system.

Audrey *et al.* [51] propose a contactless multimodal biometric system that combines two modalities: face and palmprint, by using fusion at the score level. This hardware architecture has been implemented on DSP and FPGA. Wang *et al.* [52] proposed a multimodal biometric system that implements fingerprint and voiceprint matching. Matching-score level fusion was applied to voiceprint and fingerprint. They used an ARM9-Core based S3C2440A microprocessor that works at 400 MHz and the Microsoft Windows CE operating system. Moganeshwaran *et al.* [53] use finger vein and fingerprint for their multimodal biometric system. Two biometric traits, finger vein and fingerprint, are used and the whole process is implemented in an SOC FPGA. The biometric fusion strategy applies at the matching score level. Conti *et al.* [54] propose a multimodal technique for an embedded fingerprint recognizer. In this technique, fingerprint minutiae points along with fingerprint singularity points are used for robust user authentication. For biometric fusion, a matching score fusion module is used.

10.4 Fingerprint Feature Extraction

In a fingerprint recognition system, reliable extraction of the minutiae (the ridge bifurcations and terminations) from the input fingerprint image is the most critical step, that directly affects the recognition rate. The performance of minutiae an extraction algorithm heavily depends on the quality of the input image. This necessitates the use of good-quality input scans of fingerprints [119]. In reality, the acquired fingerprint images cannot be considered as good-quality scans in all circumstances. The presence of dirt or oil on the surface of the finger may result in a blurred scan. Furthermore, due to the electronic noise present in scanner electronics, the fingerprint scans may not be of clear quality. Mostly, the noise in a fingerprint image manifests itself in cuts or interrupted ridge lines. For an automatic fingerprint identification system (AFIS) to work reliably, these broken ridge lines need to be restored via an enhancement process [251].

The minutiae extraction algorithm processes the fingerprint image in several stages in order to find the singular points related to bifurcation and termination of ridges. The number of stages and the processing involved in each one differ slightly depending on the algorithm employed, being in our case five stages that are briefly described in this section.

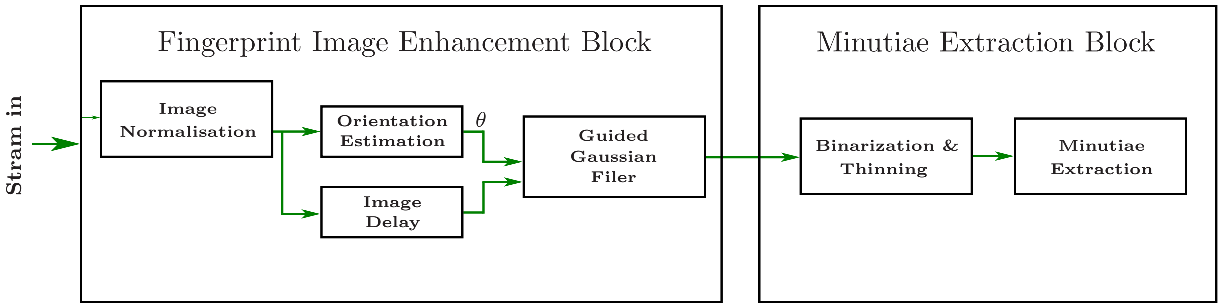


Figure 10.2: *Block diagram of fingerprint feature extraction process*

10.4.1 Image Normalisation

Normalisation is a process that changes the range of pixel intensity values. Normalisation is sometimes called contrast stretching or histogram stretching. In more general fields of data processing, such as digital signal processing, it is referred to as dynamic range expansion [94]. [188] and [189] suggested local image statistics, such as the mean and variance in a small neighbourhood, to be incorporated in the contrast improvement strategy. The local normalisation method comprises first dividing the image into appropriate small neighbourhoods and then normalising these neighbourhoods with respect to their local mean and variance. This will result in shaping these neighbourhoods to have a ridge/valley pattern with better contrast. Mathematically, it can be represented as

$$g\left(x,y\right)=\frac{I\left(x,y\right)-m_f\left(x,f\right)}{\sigma_f\left(x,y\right)} \tag{10.1}$$

where $I(x, y)$ is the input image, $m_f(x, y)$ is an estimation of a local mean of $I(x, y)$ and $\sigma_f\left(x, y\right)$ is an estimation of the local contrast (such as the standard deviation). Although the contrast is restored with no black patch at the centre, the amplitude of the granular noise in the background is significantly lifted. This happens because the background area has almost zero local variance [190], thus resulting in division by a small number, which amplifies the noise structure. We propose a function of local variance which is used as a multiplying factor for the outcome of the second phase image. The factor is defined as

$$M=1-\exp\left(-\frac{\sigma_f^2}{2C^2}\right) \tag{10.2}$$

where σ_f^2 is the local variance and C is a user-defined parameter to regulate the noise suppression power in background areas. The value of C is in the range 0-1, however, in our experiments, the value of 0.3 was adequate in all cases.

10.4.2 Orientation estimation

In this chapter, the procedure outlined in [212] was adopted for this purpose. First, discrete derivatives G_x and G_y in the x and y directions are calculated by employing a Gaussian smoothed kernel, with a small standard deviation to mitigate noise. Then, covariance matrix data for the fingerprint image was calculated for each pixel as $G_{xx} = G_x^2$,

$G_{xy} = G_x \times G_y$, and $G_{yy} = G_y^2$. The covariance matrix elements were further smoothed with a Gaussian filter having $\sigma = 1$ standard deviation. Since a ridge line has two edges, the gradient vectors at both sides of a ridge are opposite to each other. If we want to calculate θ by directly taking the average of the gradient angles in a local block, the opposite gradients at both sides of a ridge line are likely to cancel each other. To solve this problem, Kass and Witkin [118] propose a simple and clever idea of doubling the gradient angles before the averaging process. These doubled angles are smoothed with a Gaussian filter of $\sigma = 7$. Finally, the orientation is estimated by

$$\theta = \frac{\pi}{2} + \frac{\arctan\left(\frac{cos(2\theta)}{sin(2\theta)}\right)}{2} \tag{10.3}$$

10.4.3 Filtering

In this chapter, an oriented Gaussian filter is, proposed by [202], used that works similarly to an anisotropic diffusion filter. However, for a general Gaussian filter, the separable axes do not align with the image axes. While separability can be used, it is more complex to implement in this case. It is even worse for a steerable filter, where the major axis of the Gaussian kernel changes with each pixel. The oriented Gaussian filter can be expressed as

$$G_{dir}\left(x, y; \theta, f, \sigma_x, \sigma_y\right) = \exp\left\{-\frac{1}{2}\left(\frac{x_\theta^2}{\sigma_x^2} + \frac{y_\theta^2}{\sigma_y^2}\right)\right\} \tag{10.4}$$

To make the implementation process even simpler, G_{dir} is decomposed into two filters. Since $\sigma_y^2 << \sigma_x^2$ when filtering ridge patterns (such as fingerprints), the filter can be decomposed into a small isotropic filter

$$G_{iso}\left(y; \sigma_y\right) = \exp\left\{-\frac{1}{2}\left(\frac{x^2 + y^2}{\sigma_y^2}\right)\right\} \tag{10.5}$$

and an anisotropic 1D filter

$$G_{ani}\left(x_\theta; \theta, \sigma_\theta\right) = \exp\left\{-\frac{1}{2}\left(\frac{x_\theta^2}{\sigma_\theta^2}\right)\right\} \tag{10.6}$$

where $\sigma_\theta^2 = \sigma_x^2 - \sigma_y^2 \approx \sigma_x^2$ and $x_\theta = x\cos\theta + y\sin\theta$.

10.4.4 Binarisation & Thinning

In binarization, the grey scale image is converted to a binary image where the value of each pixel can be 1 or 0. A pixel set to 1 corresponds to a background/valley, whereas a pixel set to 0 is associated with a foreground/ridge. As claimed by [252], the normalisation facilitates the binarization, therefore, a simple threshold works equally well as the famous Otsu [96] thresholding method. The use of a simple threshold makes the hardware implementation much easier than the existing thresholding methods.

After binarization, the next step is thinning, performed prior to minutiae extraction. For thinning, Zhang and Suen's algorithm modified by [253] is used. In this process, 8 adjacent neighbours are evaluated to a central pixel that determines whether to delete this pixel or not. The original algorithm is modified and the process is based on the representation of the image with '1' for light (white) and '0' for dark (black), or a region point is for pixel value '0' and a background point is '1'.

10.4.5 Minutiae Extraction

After the image pre-processing step, a minutiae extraction process is applied. Minutiae point detection depends on the pixel value ('0' or '1'). This process is quite simple as it can be carried out by examining the connectivity of the thinned image. Obtaining the parameters like type and position is quite easy [254]. It depends on the position of pixel P and its connectivity. If the connectivity is 1 then it corresponds to an End Point and if the connectivity is 3, it corresponds to a Bifurcation Point (BP) of minutiae. To reduce false minutiae detected at the edge of the fingerprint image, a process to check a candidate minutiae point, whether at the edge of the image or not, is applied. Checking the existence of pixel value '0' at the right, left, top or bottom of candidate minutiae

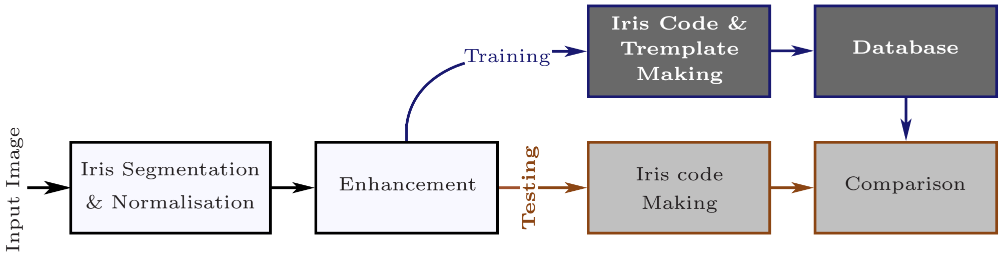


Figure 10.3: *Block diagram of iris recognition system*

points for a specific distance performs this process.

10.5 Iris Feature Extraction

The process of iris recognition can be mainly divided into three main subtasks:

- The task of extracting the iris from an already acquired image i.e. Segmentation.
- The task of straightening the extracted iris i.e. Normalisation.
- The task of extracting the feature from the normalised image i.e. Iris code making.

Fig. 10.3 shows a block diagram of the proposed iris recognition system.

10.5.1 Iris Segmentation and Normalisation

In the field of image processing, or specifically in computer vision, segmentation is defined as the partitioning of an image into smaller parts or segments which are easier to process and analyse [255]. In iris recognition, segmentation mean extraction of the pupillary and the limbic boundaries of an iris. This can be done by either localising these boundaries simultaneously (e.g. Hough transform) or first locating the pupillary and then the limbic boundary (mostly using region property based methods). Hough-transform based techniques are iterative and are not suitable for hardware implementation whereas region property-based techniques are faster and suitable for parallel processing. In region property-based techniques, the pupillary boundary is first located. Then using radial scanning (gradients) and interpolation the limbic boundary is located. Finally, the iris is normalised by using the information of the segmented iris that requires interpolation again. Such a double conversion is challenging in a streamed hardware implementation. To overcome this problem Tariq *et al.* [256] use interpolation only a single time. In this chapter, for iris segmentation and normalisation, we adapt the same technique. In this technique, the pupillary boundary of the iris is first localised using a fast region property based method. Then, instead of locating the limbic boundary, a region of interest is defined and the resultant image is converted from rectangular to polar coordinates about the centre of the pupil using

$$X = r\cos(\theta) \tag{10.7}$$

$$Y = r\sin(\theta) \tag{10.8}$$

where r represents radius of the pupil and θ is the the angle in radians. After normalisation, a gradient-based method is used to locate the true limbic boundary.

10.5.2 Feature Extraction/Iris code making

Image Enhancement

In iris code making, the normalised image is enhanced by a contrast normalisation process similar to the one proposed by [252]. This local normalisation technique deals with local image statistics in a better way. Another reason for selecting this technique is that it best suits hardware implementation. Fig. 10.4 shows a block diagram of the modified local normalisation technique. In this two phases are used: one that removes the non-uniform background and the other that restores the local contrast. In the first phase, the input image is subtracted from the Gaussian-weighted average smoothed by a low-pass Gaussian filter with σ_1. The parameter σ_1 can be set by utilising the fact that the filtered image

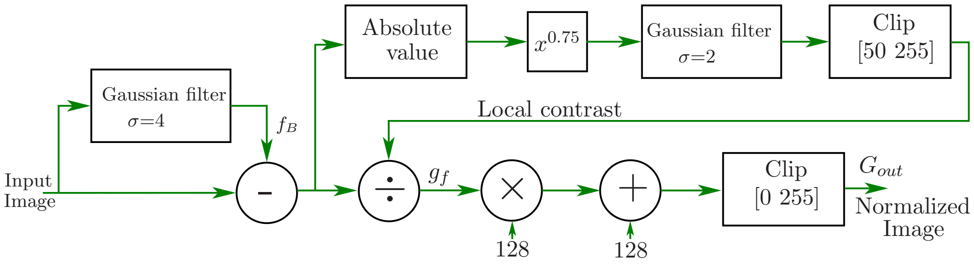


Figure 10.4: *Image enhancement using local normalisation*

should contain only the background changes (low-frequency content). In this chapter, $\sigma_1 = 4$ is used. In the second phase, the local variance of the image is computed as an estimate of the local contrast. To normalise the contrast, the resultant image of the first phase is divided pixel-wise by the standard deviation of its spatial neighbours. Again, the size of the local-variance filter depends on the size of the texture elements. As squaring the pixel value will double the base frequency, σ_2 is commonly taken to be $\sigma_1/2$. Fig. 10.5(d) shows the enhanced iris image using the proposed method.

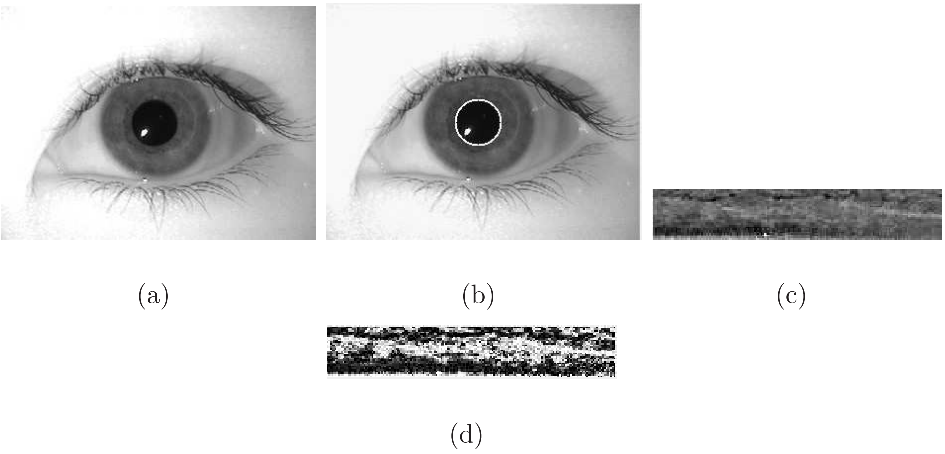


Figure 10.5: *Iris segmentation and enhancement (a) Sample eye image (b) Pupil segmentation (c) Iris normalisation and segmentation (d) Iris feature enhancement.*

Bitplane Slicing

In bitplane slicing, we first consider each grey-level value in the image, in its binary equivalent form, and then consider one of the eight bits at a time. For example, when considering bit plane 0, we check the least-significant bit of each value while forcing all other bits to zero. Now, if the least-significant bit is 1, we replace the whole number by grey-level value 1 in the image, and if it is zero we replace the whole number by grey-level value 0. In this way, bit plane zero is formed for the input image. Similarly, when considering bit plane 1, we consider the second bit from the right side of the binary sequence while forcing all others to zero. In this case, however, if the bit under consideration is 1, the original grey level value is replaced by 2 so we get the binary representation in 0 and 1 form. This process of thresholding continues in this manner for all the seven bit planes. By applying bitplane slicing on the 8 bits of the image grey-level values, we get 8 slices, as shown in Fig. 10.6. For iris code generation in our project, we have used the concept of bitplane slicing [169]. We generate 8 slices of our enhanced normalised image as mentioned above.

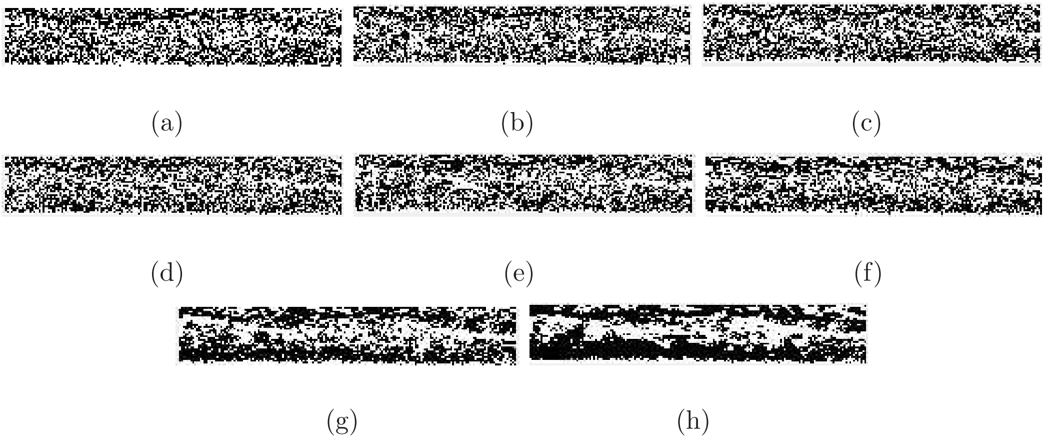

(a) (b) (c) (d) (e) (f) (g) (h)

Figure 10.6: *Bit Plane slicing: (a) Bit 0. (b) Bit 1. (c) Bit 2. (d) Bit 3 (e) Bit 4. (f) Bit 5 . (g) Bit 6. (h) Bit 7.*

In this chapter, six of the eight generated slices or planes are used to implement the bitplane slicing on the enhanced normalised image. Bit planes 0 and 7 are discarded since it is clear from the histogram of the iris image that the lowest and highest grey-level components do not fall in the iris region of the eye; mostly it consists of the middle grey-level values. The remaining bitplanes are termed a feature of type 1, 2, 3 and so on. This method of representing the features in binary codes makes the comparison process more efficient.

Iris code making

As mentioned above, we have used 6 bitplanes or slices and have neglected the bit planes 0 and 7. By considering the fact that each bit plane has two values, either 0 or a non-zero value, we have normalised all the non-zero values to 1 in all bitplanes, thus generating binary codes, so that a matching code can easily be generated. Then by using the 6 bitplanes normalised to 0's and 1's, we generate a code for size M×6, where M is the length of the 1D-transformed row vector of the normalised iris image. The final matrix is generated by simply merging the codes of the 5 bitplanes vertically. For testing hardware, 5 people of MMU-v1 iris database are used. This database has 5 left and 5 right eyes for each person and the database comprises a total of 44 persons. For iris code generation, three left and three right eye images are used while the remaining 2 left and 2 right eye images are used for testing purposes. The iris code is generated according to the following criteria:

- Three images are used for generating each person's database
- Check the majority bit, which is selected as either 0 or 1 by comparing the code, for example, by comparing the codes of a person's left eye1, left eye2 and left eye3, each of size M× 6 generated by the method mentioned above.

The whole process is well illustrated in Fig. 10.7 where we have 3 left-eye images in

$$\begin{array}{|c|c|c|c|c|}\hline 1&0&1&1&0\\\hline 1&1&0&0&0\\\hline 0&1&1&0&0\\\hline 0&1&0&1&0\\\hline 0&1&1&0&1\\\hline 0&1&1&1&1\\\hline 1&0&1&1&0\\\hline\end{array}\quad\begin{array}{|c|c|c|c|c|}\hline 1&1&0&0&1\\\hline 0&1&1&0&1\\\hline 1&1&0&0&1\\\hline 0&0&1&0&0\\\hline 1&1&1&1&1\\\hline 0&1&0&1&1\\\hline 0&1&0&0&1\\\hline\end{array}\quad\begin{array}{|c|c|c|c|c|}\hline 0&0&1&0&1\\\hline 0&1&0&1&0\\\hline 1&1&0&1&0\\\hline 1&1&0&1&1\\\hline 1&1&0&0&0\\\hline 0&0&0&1&0\\\hline 1&1&0&1&1\\\hline\end{array} = \begin{array}{|c|c|c|c|c|}\hline 1&0&1&0&1\\\hline 0&1&0&0&0\\\hline 1&1&0&0&0\\\hline 0&0&0&1&0\\\hline 1&1&1&0&1\\\hline 0&1&0&1&1\\\hline 1&1&0&1&1\\\hline\end{array}$$

Figure 10.7: *Iris code generated by applying majority bit selection*

matrix form which are normalised to 0 and 1 after applying the bitplane slicing. The fourth matrix is the result of the method mentioned above, and this matrix is later used for comparison after storing it in the database.

10.6 Matching and Fusion

The proposed approach is based on pair recognition of a fingerprint and iris, and every part provides its own Matching score. In fingerprints, minutiae matching is based on two stages: fingerprint alignment and fingerprint matching. In fingerprint alignment, a pair of minutiae (one from input minutiae and one from the template minutiae) are located and their polar location in polar coordinates is determined relative to the pair of aligned minutiae. For this purpose, we follow the technique proposed by Lindoso et. al. [257]. The output of this step gives each minutiae a triplet representation: (r, θ, o) , where θ is radial angle, r is radial distance and o is relative orientation. Finally, all the minutiae data set of the input image are compared with all the minutiae date set of the template. To compensate for errors of location the comparison uses an adaptive elastic algorithm. The final match ratio for the two fingerprints is stored.

After creating the biometric vectors the homogenous biometric vector from fingerprint and iris data is composed of binary sequences representing the unimodal biometric templates. The matching score for the iris is calculated through the Hamming distance (HD) between two final fused templates:

$$HD = \frac{1}{N} \sum_{i-1}^{N} XOR\left(Tr_{ic}\left(i\right), Ts_{ic}\left(i\right)\right) \tag{10.9}$$

where Tr_{ic} is the training feature vector and Ts_{ic} is the feature vector of the test image.

Once the matching scores of both biometric traits are obtained then these are fused using a simple sum rule. If the Fused matching score is larger than a pre-specified threshold, then the person is accepted or rejected.

10.7 Hardware Implementation

In software, usually one operation is performed at a time and its result is stored in RAM for the next operation. This is why it takes a longer time to perform a certain task which comprised of multiple sequential operations. In the hardware, these components can be combined to create parallel computing structures [197]. Almost all image processing algorithms contain operations that execute in sequence. This is a form of temporal parallelism [197]. Hence, this structure is ideal to have a separate processor for each operation. This is also known as a pipelined architecture. When processing images, data can usually begin to be output from an operation long before the complete image has been processed by that operation. The time between when data is first input to an operation and the corresponding output is available is the latency of that operation. When each operation only uses input pixel values from a small, local neighbourhood then its latency is lowest. This is because each output only requires data from only a few input pixel values. Operation pipelining can give significant performance improvements when all of the operations have low latency because a downstream processor may begin performing its operation before the upstream processors have completed.

10.7.1 Fingerprint Feature Extraction and Matching

For fingerprint feature extraction the first input image is enhanced. For image enhancement, a dynamically steerable Gaussian filter proposed by [202] is used. It is observed that with $\sigma_x = 4$ the width of the line Gaussian can be reduced from 25 to 17 pixels ($\pm 2\sigma$) without any significant effect. For this size window, this enables a simpler nearest-neighbour interpolation to be used, which significantly reduces the hardware complexity. To convert the 2-D filter into 1-D, the window is divided into two sub-windows hwind and vwind, which filter angles that are primarily horizontal and vertical respectively, as shown in Fig. 10.8. Angles within vwind require one pixel from each row within the window, while those in hwind require one from each column. The pixels corresponding to the required delays are selected and then multiplied by the Gaussian weights. Finally, these are summed up to get the final resultant value. After enhancement, the fingerprint thinning process becomes quite easy. In this process, 8 adjacent neighbours are processed with respect to a central pixel that determines whether to delete this pixel or not. The minutiae feature bifurcation and ending are obtained by a cross numbering approach [254]. In hardware, this approach is easy to implement with several resources like addition, subtraction, and shift registers.

For fingerprint alignment and matching, a pre-alignment algorithm is used [257]. Fig. 10.9 shows the hardware structure of this pre-alignment algorithm. To implement this, two memories are required. The first memory M1 consists of two sub-memories to store the extracted minutiae and related segments like position information and angle, while M2 stores the minutiae of polar coordinates. In all memories, for angles and coordinates, 8 bits are required while for minutiae only 1 bit is required. In the alignment block, the best pair of minutiae is searched. All the input minutiae are aligned by using the difference of position and angle between the two best pairs. The second block computes the modulus and angles using CORDIC. Finally, a matching block compares all the

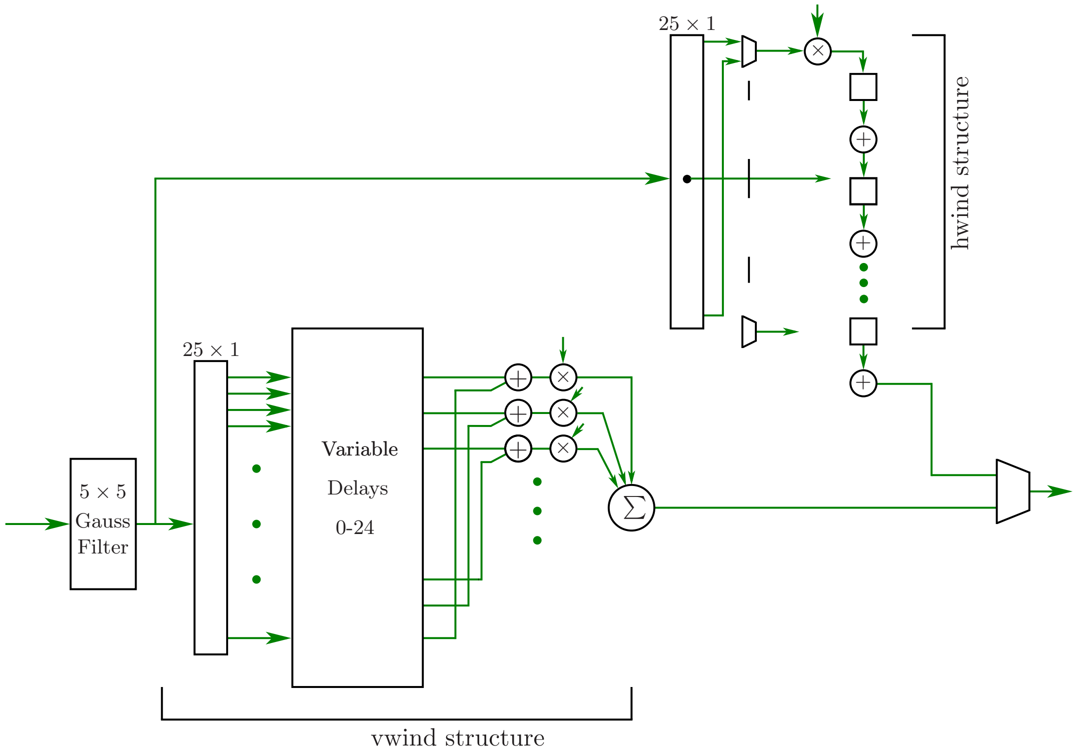


Figure 10.8: *Proposed guided-line Gaussian structure*

aligned minutiae in polar coordinates.

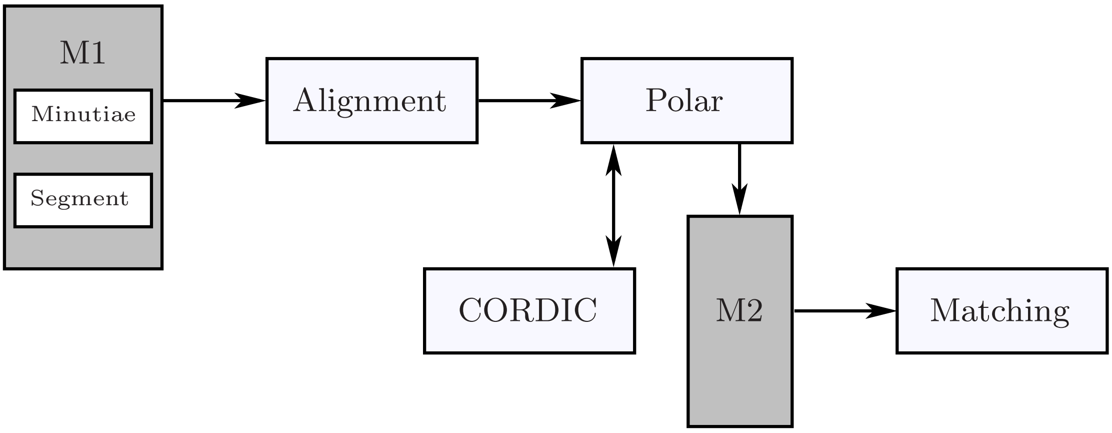


Figure 10.9: *Hardware structure of the fingerprint alignment algorithm*

10.7.2 Iris Feature Extraction and Matching

The proposed iris feature extraction is based on five steps, as shown in Fig. 10.10. The input image is first pre-processed by a mean subtraction. In the mean subtraction operation, the image is smoothed through a 2-D Gaussian filter of $\sigma = 5$. The output of this Gaussian filter is subtracted from the delayed image. The image is then thresholded at 0 by keeping only the sign bit. Two morphological operators, erosion and dilation, are applied to this binary image. The resultant image is then scanned for connected-component analysis. Two region properties, area and eccentricity, are used to isolate the pupil region from other regions of the eye image. Once the centre and radius of the pupil are located the image is cropped to contain both the pupil and iris regions. This cropped image is buffered in SRAM (off-ship memory). Using bilinear interpolation the image is normalised. Then using the first-order vertical gradient operator the limbic boundary is isolated to give the normalised iris.

After the normalisation, the image features are enhanced by using a local image nor-

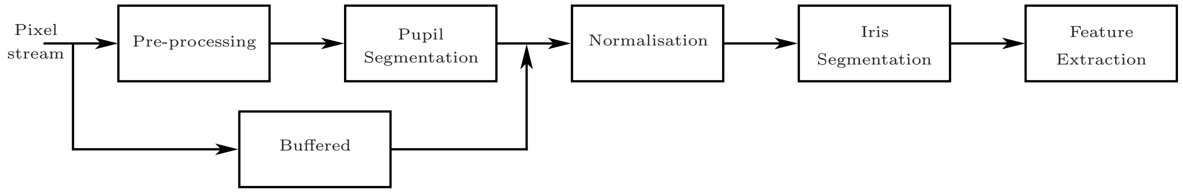


Figure 10.10: *Block diagram of iris features extraction process*

malisation. For local normalisation, the background is estimated by subtracting the mean image from the input image. For local contrast estimation, the magnitude is obtained by applying the absolute operator. Then dynamic compression is done by using a power-law transformation with $\gamma = 0.75$, applied to compress the high contrast. The resultant image is averaged locally with another Gaussian. Then by clipping the local contrast into the range [50-255] noise is suppressed. The resultant image is divided by the local contrast and the output is scaled to 128 and offset by 128. After enhancement, the next step is to implement the bit-plane slicing and create a feature vector. Both steps are quite easy to perform in hardware, as we already deal in bits in hardware. Finally, the input feature vector is compared with the already stored vector. For testing purpose we only store 5 persons feature in SRAM (off-ship memory).

10.7.3 Fusion

For fusion, for matching the scores of two biometric traits, the sum rule is used [258]. This requires normalising the scores before combining them. The reason is that the two biometric traits are of a different nature. The normalisation transforms the score into a common range between 0 and 1 [258]. Finally, the score is summed and the decision made on the basis of the threshold.

10.8 Experimental Results and Discussion

Two well-known parameters FAR and FRR are used for the performance evaluation. FAR is the number of times that access occurred for an incorrectly accepted unauthorised person. FRR is the number of times that the access of an authorised person is incorrectly rejected. First, a test is conducted on the full FVC2002 DB2A database using our proposed minutiae-based recognition system. This resulted in FAR 1.27% and FRR 18.38%. For the MMU v1 database [174], the proposed method resulted in FAR 4.29% and FRR 15.77%.

For the multimodal test, a database is created that consists 30 people in the selected fingerprint database and 30 people in the selected iris database. The classic fusion technique of matching score level is used for fusion. A Euclidean metric is applied to the HD of each subsystem. With the proposed approach for the multimodal biometric system, the following results have been obtained: FAR 1.97% and FRR 12.79%. The literature shows that fingerprint-based systems have worse accuracy than iris-based systems [259,260]. For this reason, we give higher weight to iris than fingering (0.4 for fingerprint and 0.6 for iris).

For the hardware implementation, a low-cost Cyclone IV GX P4CGX150F31FPGA is used. This FPGA combines an Intel embedded processor with Altera Cyclone IV GX FPGA. This is a full-featured computer system used for software-hardware co-designs. In Table 10.1 the detail of the hardware resource utilisation of the proposed iris recognition system is presented. It can be observed that the proposed method only consumes about 12K Logic elements, almost 12% of the total logic elements available. Our design also consumes about 10% of the logic registers. In Table 10.1 the detailed hardware resource utilisation of the proposed fingerprint recognition system is presented. If we compare the logic element consumption of both biometric traits then the proposed fingerprint recognition system consumes more logic cells and logic registers than the iris recognition. On the other hand, the proposed iris recognition system consumes more memory bits than the fingerprint recognition system. The reason is that, in iris recognition, the whole image needs to be buffered once for extracting the outer boundary. Also, the template size of the iris is much bigger than the fingerprint that requires more memory to store the feature for recognition.

In Table 10.3, the relative processing speed of the MATLAB-based proposed fingerprint recognition system, iris recognition system and fusion is compared with its FPGA-based structure. The proposed fingerprint recognition is over 240 times faster than the MATLAB-based implementation on a PCA. The proposed iris recognition is over 197 times faster than the MATLAB-based implementation. Our complete multimodal biometric system takes about 16 seconds to recognise a person. In the FPGA implementation, both biometric traits are processed in parallel, significant boosting the overall speed of the system. Our hardware-based multimodal system takes about 60 milliseconds to recognise a person; that is over 270 times faster than the MATLAB-based system. The reason for this high speed is the efficient use of parallelism in the FPGA.

10.9 Conclusion

This chapter presents a reliable multimodal biometric system that satisfies multiple constraints: low cost, real-time processing, hygienic, straightforwardness, user-friendliness, limited memory, etc. To achieve this, we present a hardware architecture of a multimodal biometric system that massively exploits the inherent parallelism. The proposed system is based on multiple biometric fusion that uses two biometric traits, fingerprint and iris. Both fingerprint and iris are highly accurate biometric traits. The proposed system is efficiently implemented in hardware. As far as the authors know, the proposed structure is the only one that gives the hardware implementation of a complete multimodal biometric using fingerprint and iris recognition. The proposed hardware system is over 270 times faster than the MATLAB-based system. We also plan to investigate the further optimisation of the two biometric traits to improve the FAR and FRR and to further optimise the hardware resource utilisation.

Table 10.1: *Detailed Hardware resource utilisation of the proposed iris recognition system on a low-cost Cyclone IV GX FPGA*

Resources	Available	Pupil Segmentation	Normalisation	Iris Segmentation and Enhancement	Iris recognition
Logic Elements	149760	2810	597	2986	5251
Logic register	149760	1174	335	1100	3212
Memory bits	6M	612k	364k	88k	386k

Table 10.2: *Detailed Hardware resource utilisation of the proposed fingerprint recognition stem on a low-cost Cyclone IV GX FPGA*

Resources	Available	Normalization	Orientation estimation	Filtering	Minutiae recognition
Logic Elements	149760	2286	7597	8268	4297
Logic registers	149760	917	3335	5334	2235
Memory bits	6M	60k	164k	58k	86k

Table 10.3: *Processing speed (in seconds) of proposed FPGA-based algorithm with proposed PC-based MATLAB structure*

FVC2004 Database Type	Proposed MATLAB(PC)	Proposed FPGA	Speedup
Fingerprint recognition	8.95	0.0360	240×
Iris recognition	5	0.0260	197×
Fusion	2.24	0.0120	188×

Chapter 11

Conclusions and Future Work

11.1 Conclusions

This dissertation presents a hardware implementation of a multimodal biometric system using fingerprint and iris biometric traits/identifiers. The proposed system is divided into two main blocks: software design and hardware design. Using MATLAB, a complete multimodal biometric system is first designed at the software level. The software design is further sub-divided into two blocks: fingerprint feature extraction and iris feature extraction. Both biometric identifiers are designed and optimised independently. Each biometric trait is first optimised at the software level, by addressing some of the issues that directly affect the FAR and FRR. Then the hardware architectures for both biometric traits are presented, followed by a final multimodal hardware architecture. The key contributions of the dissertation can be summarised as follows:

- A novel inpainting method for scar removal is presented that can be regarded as a pre-processing step in fingerprint image enhancement. The proposed method is computationally less expensive and gives an effective solution for low-quality fingerprint image enhancement.

- We propose a spatial entropy change that provides a reasonable stopping rule for isotropic as well as anisotropic diffusion.
- A modified local normalisation procedure is proposed that enhances the contrast of the foreground ridge/valley area uniformly with almost no normalisation for the background region. To circumvent the amplification of background noise, a correction factor is introduced in the form of a monotonically increasing function of local variance values. Our proposed design is single-pass and does not use any external memory (SRAM/DRAM). The use of a single-pass with streamed processing makes the proposed architecture significantly faster than existing implementations. As far as the authors know, no other design is based on a single-pass.
- An efficient hardware architecture for fingerprint image normalisation is proposed and its efficiency is compared with other state-of-the-art hardware normalisation systems. The proposed hardware structure manages to efficiently speed up the image processing time by a factor of 18, capable of processing one pixel per clock cycle at over 128.3 MHz. This is achieved by using an oriented line Gaussian and its separability property along with efficient hardware implementation using parallelism of streamed data.
- We have presented the design of a steerable anisotropic Gaussian filter for fingerprint image enhancement. The design allows the orientation of the filter to be changed with each clock cycle, giving true steerability with a single filter, rather than requiring a bank of filters. The whole design was constructed with a relatively modest use of resources on a small low-cost FPGA and was able to give a real-time performance, being capable of processing one pixel per clock cycle at over 90 MHz.
- A reliable orientation-field (OF) estimation is implemented in hardware. The proposed design efficiently uses CORDIC in vectoring mode to determine 2θ, combined

with a second CORDIC in rotation mode to calculate $sin(2\theta)$ and $cos(2\theta)$.

- A new local histogram and standard deviation based pupil segmentation method is proposed. Using the standard deviation and finding the region that has the highest probability of having the pupil region from the local histogram has overcome the drawbacks faced when locating the pupil by using the global histogram.

- A fast and novel method for pupil segmentation is developed that is based on a shape detector and an intensity-based threshold. To estimate the pupil centre and radius, a LoG filter along with a region growing is used. The true pupillary boundary is refined using the zero crossings of a second LoG filter. Next, the true orientation of the eye in the image is estimated using a third LoG filter. Using the zero crossings of the LoG, the search is initially started from the stable zones and then extended to the occlusion zones. The discontinuities are located, giving indications of eyelids in the iris region. Finally, using the interpolation the iris outer boundary as well as eyelid arcs are estimated.

- A real-time hardware implementation of iris segmentation is presented that uses morphological operators along with two region properties, eccentricity and area. The proposed method fully utilises the parallel and pipelined architecture of an FPGA than other existing algorithms with 80% reduction in resources compared with Ngo's implementation. The proposed FPGA implementation is 25% faster than the Ngo FPGA implementation.

- A reliable multimodal biometric system is presented that respects multiple constraints: real-time processing, low cost, hygienic, straightforward, user-friendly, limited memory, etc. Our proposed architecture massively exploits the inherent parallelism and achieves a speed 270 times faster than the MATLAB-based system.

11.2 Future Work

In future the following aspects of a Multimodal biometric system can be considered for further research and development:

- In future, our aim is to integrate more biometric features like face and speech recognition with our designed system. The use of a common imaging device for capturing the face and voice template is expected to cut down the overall cost of implementation. This combination would also boast of high accuracy. The need of integrating these biometric with our designed system is based on the fact that they have been used routinely in the law enforcement community. Most successful commercial biometric systems currently rely on either fingerprint, face, iris or voice. Further, these biometric indicators complement one another in their advantages and strengths. Another advantage of integrating more biometric features is that, by making use of multiple methods of identification, a system can preserve higher threshold recognition settings and a system administrator can make a decision on the level of security that is needed. For an extremely high-security site/area, one might need to use up to three biometric identifiers and for a lower security site/area, one could possibly require one or two credentials. If one of the identifiers fails for any unknown reason, the system can still utilise another one or two of them to provide accurate identification of a person. In this way, it will significantly reduce the probability of admitting an imposter.

- The proposed scar removal strategy uses several iterative processes for filling the scar that is poorly less suited for hardware. The proposed design needs to be optimised for hardware implementation.

- We also plan to investigate further optimisation of the orientation estimation and anisotropic filtering stages of the algorithm. Orientation diffusion can be replaced with Gaussian filtering in the hardware implementation of the fingerprint orientation estimation block. This can improve the overall efficiency of the fingerprint enhancement module.

- In the presence of dense eyelashes and eyebrows the proposed LoG filter based iris method fails to locate the true pupillary boundary. Such dense eyebrows and eyelashes affect the LoG filtering (along with region growing). This can be improved by using some other tools along with LoG filtering, like the fast Hough transform. Similarly, instead of a region growing method some other techniques can be applied to further improve the efficiency of iris segmentation.
- The existing LoG-based iris segmentation is rather costly in terms of its hardware implementation. Further investigations are required to optimise this technique for an efficient hardware implementation.
- The proposed hardware architecture of iris segmentation needs to save the image on off-chip memory SRAM during processing. It requires a minimum of two passes to segment the pupillary and limbic boundaries. The work can be extended to find an optimal solution that avoids the SRAM usage.

Appendix A

List of Acronyms

1-D	One-dimensional
2-D	Two-dimensional
3-D	Three-dimensional
ADSP	Advanced Digital Signal Processor
AFIS	Automated Fingerprint Identification System
ARM	Advanced RISC Machines
ATM	Automated Teller Machine
BP	Bifurcation point
CASIA	Chinese Academy of Sciences Institute of Automation

CED	Coherence Enhanced Diffusion
CHT	Circular Hough Transform
CLAHE	Contrast Limited Adaptive Histogram Equalisation
CORDIC	COordinate Rotation DIgital Computer
Cp	Current point
DB	Database
DFB	Directional Filter Bank
DNA	Deoxyribonucleic Acid
DRAM	Dynamic Random Access Memory
DSP	Digital Signal Processing
EBCM	Edge Based Contrast Measure
EER	Equal Error Rate
FAR	False Acceptance Rate
FFT	Fast Fourier Transform

FIFO	Fist In Fist Out
FIR	Finite Impulse Response
FN	False Negative
FP	False Positive
FPGA	Field-programmable gate array
FRR	False Rejection Rate
FTE	Failure-to-enrol Rate
FVC2002	Fingerprint Verification Competition 2002
FVC2004	Fingerprint Verification Competition 2004
GB	Giga bytes
GF	Gabor Filter
GHz	Gigahertz
GI	Goodness Index
ICA	Independent Component Analysis

ID	Identification
IDO	Integral Differential Operator
KDDA	Kernal Direct Discriminant Analysis
LDA	Linear Discriminant Analysis
LOG	Laplacian of Gaussian
LUT	Look Up Table
MATLAB	Matrix Laboratory
MGF	Modified Gabor Filter
MMU	Multimedia University
NIST	National Institute of Standards and Technology
NN	Neural networks
Np	North point
OF	Orientation Field
PC	Personal Computer

PIN	Person Identification Number
PSO	Particle Swarm Optimisation
RAM	Random Access Memory
RF	Ridge Frequency
RISK	Reduced Instruction Set Computing
RGB	Red Green Blue
SDRAM	Static Dynamic Random Access Memory
SE	Structuring Element
SOC	System On Chip
SNR	Signal to Noise Ratio
SRAM	Static Random Access Memory
STFT	Short Time Fourier Transform
TN	True Negative
TP	True Positive

VGA	Video Graphics Array
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Integrated Circuit
Wp	West Point

Bibliography

[1] L. Hong, Y. Wan, and A. Jain, "Fingerprint image enhancement: Algorithm and performance evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 777–789, 1998.

[2] L. H. A. K. Jain and Y. Kulkarni, "A multimodal biometric system using fingerprint, face and speech," in *Second International Conference on AVBPA*, 1999, pp. 182–187.

[3] U. M. Bubeck, "Multibiometric authentication: An overview of recent developments." San Diego State University, California, USA., Tech. Rep., 2003.

[4] A. K. Jain, "Biometric recognition: How do i know who you are?" in *International Conference on Image Analysis and Processing, Springer LNCS,*, vol. 3617/2005, pp. ., 2005, pp. 19–26.

[5] Y. Chen, S. C. Dass, and A. K. Jain, "Fingerprint quality indices for predicting authentication performance," in *Fifth International Conference on Audio and Video-Based Biometric Person Authentication (AVBPA*, 2005, pp. 373–381.

[6] M. Golfarelli, D. Maio, and D. Maltoni, "On the error-reject tradeoff in biometric verification systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 786–796., 1997.

[7] U. Uludag, A. Ross, and A. K. Jain, "Biometric template selection and update: A case study in fingerprints." *Pattern recognition*, vol. 37, no. 7, pp. 1533–1542, 2004.

[8] T. Matsumoto, H. Matsumoto, K. Yamada, and S. Hoshino, "Impact of artificial gummy fingers on fingerprint systems," *SPIE Optical Security and Counterfeit Deterrence Techniques IV*, vol. 4677, pp. 275–289, 2002.

[9] A.Ross and A. Jain, "Information fusion in biometrics," *Pattern Recognition Letters*, vol. 24, 2003.

[10] A. Jain, S. P. S, and L. Hong, "A multichannel approach to fingerprint classification," *IEEE Transactionons on Pattern Analysis and Machine Intelligence*, vol. 21, no. 4, pp. 348–359, 1999.

[11] S. Chaudhary and R. Nath, "A multimodal biometric recognition system based on fusion of palmprint, fingerprint and face," in *Advances in Recent Technologies in Communication and Computing, 2009. ARTCom '09. International Conference on*, Oct 2009, pp. 596–600.

[12] A. Ross and A. K. Jain, "Multimodal biometrics: An overview," in *Signal Processing Conference, 2004 12th European*, Sept 2004, pp. 1221–1224.

[13] Z. Akhtar, C. Micheloni, and G. L. Foresti, "Biometric liveness detection: Challenges and research opportunities," *IEEE Security Privacy*, vol. 13, no. 5, pp. 63–72, Sept 2015.

[14] S. Marcel, M. S. Nixon, and S. Z. Li, Eds., *Handbook of Biometric Anti-Spoofing*. Springer, 2014.

[15] R. Cappelli, D. Maio, and D. Maltoni, "Combining fingerprint classifiers," in *First Internat. Workshop on Multiple Classifier Systems.*, 2000, pp. 351–361.

[16] T. K. Ho, J. J. Hull, and S. N. Srihari, "Decision combination in multiple classifier systems," *IEEE Transactionons on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 66–75, 1994.

[17] B. G. Sherlock, D. M. Monro, and K. Millard, "Fingerprint enhancement by directional fourier filtering," in *IEEE Proc. Vision, Image Signal Processing*, vol. 141, no. 2, 1994.

[18] S. Prabhakar and A. K. Jain, "Decision-level fusion in fingerprint verification," *Pattern Recognition*, vol. 35, no. 4, pp. 861–874, 2002.

[19] A. K. Jain, S. Prabhakar, and S. Chen, "Combining multiple matchers for a high security fingerprint verification system," *Pattern Recognition Letters*, vol. 20, no. 1113, pp. 1371 – 1379, 1999. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865599001087

[20] G. L. Marcialis and F. Roli, *Experimental Results on Fusion of Multiple Fingerprint Matchers*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 814–820.

[21] L. H. A. K. Jain and R. Bolle, "On-line fingerprint verification," *IEEE Transactionons on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 302–314, 1997.

[22] F. Alonso-Fernandez, J. Fierrez-Aguilar, H. Fronthaler, K. Kollreider, J. Ortega-Garcia, J. Gonzalez-Rodriguez, and J. Bigun, "Combining multiple matchers for fingerprint verification: A case study in biosecure network of excellence," *Annales Des Télécommunications*, vol. 62, no. 1, pp. 62–82, 2007.

[23] L. Hong and A. K. Jain, "Integrating faces and fingerprints for personal identification," *IEEE Transactionons on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 1295–1307, 1998.

[24] R. P. D. J. Kittler, M. Hatef and J. G. Matas, "On combining classifiers," *IEEE Transactionons on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 226–239, 1998.

[25] F. Wang and J. Han, "Multimodal biometric authentication based on score level fusion using support vector machine," *Opto-Electronics Review*, vol. 17, no. 1, pp. 59–64, 2008.

[26] P. P. U. Dieckmann and T. Wagner, "Sesam: A biometric person identification system using sensor fusion," *Pattern Recognition Letters*, vol. 18, no. 9, pp. 827–833, 1997.

[27] P. Verlinde and G. Cholet, "Comparing decision fusion paradigms using k-nn based classifiers, decision trees and logistic regression in a multi-modal identity verification application," in *Second International Conference on AVBPA*, 1999, pp. 188–193.

[28] E. Bigun, J. Bigun, B. Duc, and S. Fischer, "Expert conciliation for multimodal person authentication systems using bayesian statistics," in *First International Conference on AVBPA*, 1997, pp. 291–300.

[29] R. Brunelli and D. Falavigna, "Person identification using multiple cues," *IEEE Transactionons on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 955–966, 1995.

[30] A. K. Jain, S. Prabhakar, and A. Ross., "Fingerprint matching: Data acquisition and performance evaluation," MSU-TR:99-14, Michigan State University, Tech. Rep., 1999.

[31] A. Ross, K. Nandakumar, and A. K. Jain, *Handbook of multibiometrics.* Springer-Verlag, 2006.

[32] C. Chia, N. Sherkat, and L. Nolle, *Confidence Partition and Hybrid Fusion in Multimodal Biometric Verification System.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 212–219.

[33] H. B. Kekre, V. A. Bharadi, V. I. Singh, V. Kaul, and B. Nemade, "Hybrid multimodal biometric recognition using kekres wavelets, 1d transforms & kekres vector quantization algorithms based feature extraction of face & iris," in *International Conference and Workshop on Emerging Trends in Technology*, 2011, pp. 29–34.

[34] R.-L.Hsu, "Face detection and modeling for recognition." Ph.D. dissertation, Department of Computer Science and Engineering, Michigan State University, East Lancing, USA., 2002.

[35] X. Liu and T. Chen, "Geometry-assisted statistical modeling for face mosaicing." in *International Conference on image Processing (ICIP)*, vol. 2, Barcelona, Spain., 2003, pp. 883–886.

[36] R. Raghavendra, A. Rao, and G. H. Kumar, "Multisensor biometric evidence fusion of face and palmprint for person authentication using particle swarm optimization (pso)," *International Journal of Biometrics*, vol. 2, no. 1, pp. 19–33, 2010.

[37] G. Feng, K. D. adn D. Hu, and D. Zhang, "When faces are combined with palmprint: A novel biometric fusion strategy," in *First International Conference on Biometric Authentication (ICBA)*, Hong Kong, 2004, pp. 701–707.

[38] A. Kumar, D. C. M. W. amd H. C. Shen, and A. K. Jain., "Personal verification using palmprint and hand geometry biometric." in *Time-Varying Image Processing and Moving Object Recognition*, Guildford, UK, 2003, pp. 668–678.

[39] J. Fierrez-Aguilar, L. Nanni, J. Ortega-Garcia, R. Cappelli, and D. Maltoni, *Combining Multiple Matchers for Fingerprint Verification: A Case Study in FVC2004*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 1035–1042.

[40] Y. Wang, T. Tan, and A. K. Jain, “Combining face and iris biometrics for identity verification,” in *Fourth International Conference on Audio- and Video-based Biometric Person Authentication (AVBPA)*, Guildford, UK, 2003, pp. 805–813.

[41] X. Lu, Y. Wang, and A. K. Jain, “Combining classifiers for face recognition,” in *International Conference on Multimedia and Expo (ICME)*, vol. 3, 2003, pp. 13–16.

[42] A. Ross and R. Govindarajan, “Feature level fusion using hand and face biometrics,” in *SPIE Conference on Biometric Technology for Human Identification II*, vol. 5779, Orlando, USA, 2005, pp. 196–204.

[43] V. C. Subbarayudu and M. V. N. K. Prasad, “Multimodal biometric system,” in *First International COnference on Emerging Trends in Engineering and Technology*, 2008.

[44] L. Hong and A. K. Jain, “Integrating faces and fingerprints for personal identification,” *IEEE Transactionons on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 1295–1307, 1998.

[45] A. Ross, A. K. Jain, and J. Reisman, “A hybrid fingerprint matcher,” *Pattern Recognition*, vol. 36, pp. 1661–1673, 2003.

[46] V. Conti, G. Milici, P. Ribino, S. Vitabile, and F. Sorbello, “Fuzzy fusion in multimodal biometric systems,” in *Knowledge-Based Intelligent Information and Engineering Systems*, 2010, pp. 108–115.

[47] F. Yang and B. Ma, “A new mixed-mode biometrics information fusion based-on fingerprint, hand-geometry and palm-print,” in *4th International IEEE Conference on Image Graphics*, 2007, pp. 689–693.

[48] P. Yu, D. Xu, H. Zhou, and H. Li, “Decision fusion for hand biometric authentication,” in *International Conference on Intelligent Computing and Intelligent Systems (ICIS)*, vol. 4, Shanghai, China., 2009, pp. 486 –490.

[49] S. S. Athalea, D. Patilb, P. Deshpandec, and Y. H. Dandawated, “Hardware implementation of palm vein biometric modality for access control in multilayered security system,” *Procedia Computer Science*, vol. 28, pp. 492–498, 2015.

[50] J.-H. Yoo, J.-G. Koa, Y.-S. Chung, S.-U. Jung, K.-H. Kim, K.-Y. Moon, and K. Chung, “Design of embedded multimodal biometric systems,” *International IEEE Conference on Signal-Image Technologies and Internet-Based System*, pp. 1058–1062, 2007.

[51] A. Poinsot, F. Yang, and V. Brost, “Palmprint and face score level fusion: hardware implementation of a contactless small sample biometric system,” *Optical Engineering*, vol. 50, no. 2, pp. 027 002–1–027 002–12, 2011.

[52] J. Wang, Y. Li, P. Liang, G. Zhang, and X. Ao, “An effective multi-biometrics solution for embedded device,” in *IEEE International Conference on Systems, Man and Cybernetics*, 2009, pp. 917–922.

[53] R. Moganeshwaran, M. K. Hani, and M. A. Suhaini, “Fingerprint-fingervein multimodal biometric authentication system in field programmable gate array,” in *IEEE International Conference on Circuits and Systems (ICCAS)*, 2012, pp. 337–342.

[54] V. Contia, C. Militelloa, F. Sorbelloa, and S. Vitabile, “A multimodal technique for an embedded fingerprint recognizer in mobile payment systems,” *Mobile Information Systems*, vol. 5, pp. 105–124, 2009.

[55] V. Conte, C. Militello, and F. Sorbello, “A frequency based approach for feature fusion in fingerprint and iris multimodal biometric identification systems,” *IEEE Transactions of System, Man and Cybernetic-Part C*, vol. 40, no. 4, pp. 384–395, 2010.

[56] A. K. Jain, L. H. S. Pankanti, and R. Bolle, “An identity-authentication system using fingerprints,” in *IEEE Proceedings*, vol. 85, 1997, pp. 1365–1388.

[57] C. Gottschlich, “Curved-region-based ridge frequency estimation and curved gabor filters for fingerprint image enhancement,” *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 2220–2227, April 2012.

[58] D. Maio, D. Maltoni, R. Capelli, j. L. Wayman, and A. K. Jain, “FVC2004: Third fingerprint verification competation,” in *International Conference on Biometrics*, Hong Kong, 2004, pp. 1–7.

[59] J. Yang, N. Xiong, and V. A. Vasilakos, “Two-stage enhancement scheme for low-quality fingerprint images by learning from the images,” *IEEE Transactions on Human-Machine Systems*, vol. 43, no. 2, pp. 235–248, March 2013.

[60] L. Hong, Y. Wan, and A. Jain, “Fingerprint image enhancement algorithm and performence evolution,” in *IEEE Transications on Pattern analysis and Machine Intelligence*, vol. 20, 1998, pp. 777–789.

[61] T. Aach, A. Kaup, and R. Mester, “On texture analysis: Local energy transforms versus quadrature filters,” *Signal Processing*, vol. 45, pp. 173–181, 1995.

[62] M. Wen, Y. LIang, Q. Pan, and H. Zhang, "A gabor filter based fingerprint enhancement algorithm in wavelet domain," in *IEEE International Symposium on-Communications and Information Technology*, 2005, pp. 1468–1471.

[63] J. Yang, L. Liu, T. Jiang, and Y. Fan, "A modified Gabor filter design method for fingerprint image enhancement," *Pattern Recognition Letters*, vol. 24, pp. 1805–1817, 2003.

[64] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12(7), pp. 629–639, 1990.

[65] A. Almansa and T. Lindberg, "Fingerprint enhancement by shape adaptation of scale-space operators with automatic scale-selection," *IEEE Transactions on Image Processing*, vol. 9, no. 10, pp. 2027–2042, July 2000.

[66] J. Weickert, *Anisotropic Diffusion in Image Processing.* B. G. Teubner (Stuttgart), 2001.

[67] G. Yang, P. Burger, D. Firmin, and S. R. Underwood, "Structure adaptive anisotropic filtering," *Image and Vision Computing*, vol. 14, pp. 135–145, 1996.

[68] S. Greenberg, M. Aladjem, D. Kogan, and I. Dimitrov, "Fingerprint image enhancement using filtering techniques," in *15th International Conference on Pattern Recognition*, vol. 3, 2000, pp. 322–325.

[69] M. A. U. Khan, K. Ullah, A. Khan, and I. U. Islam, "Robust multi-scale orientation estimation: Directional filter bank based approach," *Applied Mathematics and Computation*, pp. 814–824, 2014.

[70] P. Perona, “Orientation diffusions,” *IEEE Transactions on Image Processing*, vol. 7, pp. 457–467, 1998.

[71] S. Chikkerur, A. Cartwright, and V. Govindaraju, “Fingerprint image enhancement using STFT analysis,” *Pattern Recogn*, vol. 40, pp. 198–211, 2007.

[72] S. Jirachaweng, Z. Hou, W.-Y. Yau, and V. Areekul., “Residual orientation modeling for fingerprint enhancement and singular point detection,” *Pattern Recognition*, vol. 44, no. 2, pp. 431–442, 2011.

[73] J. Yang, N. Xiong, and A. V. Vasilakos, “Two-stage enhancement scheme for low-quality fingerprint images by learning from the image,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 2013.

[74] F. Belhadj, S. Akrouf, S. Harous, and S. A. Aoudia, “Efficient fingerprint singular points detection algorithm using orientation-deviation features,” *Journal of Electronic Imaging*, vol. 24, no. 3, pp. 1–6, 2015.

[75] H. GENG, J. LI, J. ZHOU, and D. Chen, “An improved gabor enhancement method for low-quality fingerprint images,” in *AOPC 2015: Image Processing and Analysis,*, 2015.

[76] L. Nanni, C. Casanova, S. Brahnam, and A. Lumini, “Empirical tests on enhancement techniques for a hybrid fingerprint matcher based on minutiae and texture,” *International Journal on Artificial Intelligence Tools*, 2012.

[77] “FVC2002: the Second International Fingerprint Verification Competition, http://bias.csr.unibo.it/fvc2002/.”

[78] “NIST Special Database 4, NIST 8-Bit Gray Scale Images of Fingerprint Image Groups (FIGS), http://www.nist.gov/srd/nistsd4.htm.”

[79] M. Kcken and A. Newell, “Fingerprint formation.” *Theor Biol. 2005*, vol. 235, no. 1, pp. 71–83, July 2005.

[80] M. M. Oliveira, B. Bowen, R. McKenna, and Y.-S. Chang, “Fast digital image inpainting,” in *The International Conference on Visualization, Imaging and Image Processing (VIIP 2001)*, 2001.

[81] C. Gottschlich, P. Mihailescu, and A. Munk, “Robust orientation field estimation and extrapolation using semilocal line sensors,” *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 4, pp. 802–811, Dec 2009.

[82] W. Y. Y. ann Xudong Jiang and W. Ser, “Enhancing oriented pattern using adaptive directional fft filter,” in *IEEE 2nd International Conference on Information, Communication & Signal Processing, Singapore*, December 1999.

[83] G. Sulong, T. Saba, A. Rehman, and Saparudin, “A new scar removal technique of fingerprint images,” in *International Conference on Instrumentation, Communications, Information Technology, and Biomedical Engineering (ICICI-BME)*, 2009.

[84] M. Bertalmio, A. L. Bertozzi, and G. Sapiro., “Navier-stokes, fluid dynamics, and image and video inpainting,” in *ICCV 2001*, 2001, pp. 1335–1362.

[85] A. Telea, “An image inpainting technique based on the fast marching method,” *Journal of Graphics Tools*, vol. 9, no. 1, pp. 23–34, 2004.

[86] G. Sulong, T. Saba, A. Rehman, and Saparudin, “A new scar removal technique of fingerprint images,” in *International Conference on Instrumentation, Communications, Information Technology, and Biomedical Engineering (ICICI-BME)*, vol. 1, November 2009, pp. 131–135.

[87] W. Y. Yau, X. Jian, and W. Ser, “Enhanced oriented pattern using adaptive directional FFT filter,” in *IEEE 2nd International Conference on Information, Communication & Signal Processing*, 1999.

[88] A. Bazen and S. Gerez, “Segmentation of fingerprint images,” in *Workshop on Circuits Systems and Signal*, 2001.

[89] A. J. Frame, P. E. Undrill, M. J. Cree, J. A. Olson, K. C. McHardy, P. F. Sharp, and J. V. Forrester, “A comparison of computer based classification methods applied to the detection of microaneurysms inophthalmic fluorescein angiograms,” *Computers in Biology and Medicine*, vol. 28, no. 3, pp. 225–238, 1998.

[90] M. Niemeijer, B. van Ginneken, J. J. Staal, M. S. A. Suttorp-Schulten, and M. D. Abramoff, “Automatic detection of red lesions in digital color fundus photographs,” *IEEE Trans. Med. Imag.*, vol. 24, pp. 584–592, 2005.

[91] B. Zhang, X. Wu, J. You, Q. Li, and F. Karray, “Detection of microaneurysms using multi-scale correlation coefficients,” *Pattern Recognition*, vol. 43, p. 22372248, 2012.

[92] L. Streeter and M. J. Cree, “Microaneurysm detection in colour fundus images,” *Image Vision Computing New Zealnd Conference*, pp. 280–284, 2003.

[93] R. Jagoe, J. Arnold, C. Blauth, P. L. C. Smith, K. M. Taylor, and R. Wootton, “Measurement of capillary dropout in retinal angiograms by computerised image analysis,” *Pattern Recognit. Lett*, vol. 13, p. 143151, 1992.

[94] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (3rd Edition).* Prentice-Hall, Inc, 2006.

[95] M. Sharma and A. Kumar, “Non uniform background illumination removal (nubir) from microscopic images,” *International Journal of Advanced Research in Technol-*

ogy, Engineering and Science (A Bimonthly Open Access Online Journal), vol. 1, pp. 23–29, 2014.

[96] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems Man and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[97] M. Jacob and M. Unser, "Design of steerable filters for feature detection using Canny-like criteria," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 8, pp. 1007–1019, August 2004.

[98] T. Lindeberg, "Edge detection and ridge detection with automatic scale selection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition.* IEEE, 1996, pp. 465 –470.

[99] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever, "Multiscale vessel enhancement filtering," in *Medical Image Computing and Computer-Assisted Intervention*, 1998, pp. 130–137.

[100] M. Kass and A. Witkins, "Analyzing oriented patterns," *Computer Vision, Graphics, and Image Processing*, vol. 37, pp. 362 – 385, 1987.

[101] B. Jahne, *Image described in Practical Handbook on Image Processing for Scientific and Technical Applications.* CRC Press, 1997.

[102] M. A. Khan, K. Ullah, A. Khan, and I. U. Islam, "Robust multi-scale orientation estimation: Directional filter bank based approach," *Elsevier Journal of Applied Mathematics and Computation*, vol. 242, pp. 814–824, Sept. 2014.

[103] D. Maio, D. Maltoni, R. Capelli, J. L. Wayman, and A. K. Jain, "FVC2004: Third fingerprint verification competition," in *ICBA Hong Kong*, 2004.

[104] M. D. Garris and R.M.McCabe, "NIST special database 27: Fingerprint minutiae from latent and matching tenprint images," Nat. Inst. Stand. Technol., Gaithersburg, MD, NIST Tech. Rep. 6534, Tech. Rep., 2000.

[105] H. Fronthaler, K. Kollreider, and J. Bigun, "Local features for enhancement and minutiae extraction in fingerprints," *IEEE Trans. Image Process*, vol. 17, pp. 354–363, 2008.

[106] T. M. Khan, M. A. U. Khan, and Y. Kong, "Fingerprint image enhancement using multi-scale DDFB based diffusion filters and modified Hong filters," *Optik-International Journal for Light and Electron Optics*, vol. 124, no. 16, pp. 4206–4214, August 2014.

[107] T. M. Khan, M. A. Khan, and Y. Kong, "Boosting ced using robust orientation estimation," *The International Journal of Multimedia & Its Applications*, vol. 6, no. 2, April 2014.

[108] A. Julasayvake and S. Choomchuay, "An algorithm for fingerprint core point detection," in *International Symposium on Signal Processing and its Applications IAAPA-2007*, 2007.

[109] J. Abraham, P. Kwan, and J. Gao, *Fingerprint Matching using A Hybrid Shape and Orientation Descriptor, State of the art in Biometrics.* Dr. Jucheng Yang (Ed.), 2011.

[110] J. Abraham, "Matlab code for filengerprint matching algorithm. url: http://www.mathworks.com/matlabcentral/fileexchange/29280-fingerprint-matching -algorithm-using-shape-context-and-orientation-descriptors," 2010.

[111] R. Fluss, D. Faraqqi, and B. Reiser, "Estimation of the Youden index and its associated cutoff point." *Bimetrical Journal*, vol. 47, no. 4, pp. 458–472, 2005.

[112] C. Gottschlich and C.-B. Schnlieb, “Oriented diffusion filtering for enhancing low-quality fingerprint images,” *IET Biometrics*, vol. 1, no. 2, pp. 105 – 113, 2012.

[113] J. Bernd, *Spatio-Temporal Image Processing*, 1st ed. Springer-Verlag Berlin Heidelberg, 1993, vol. 751.

[114] A. Z. Averbuch, F. G. Meyer, J. O. Stromberg, R. R. Coifman, and A. Vassiliou, “Low bit-rate efficient compression for seismic data,” *IEEE Transactions on Image Processing*, vol. 10, no. 12, pp. 1801–1814, 2001.

[115] J. Weickert, “Coherence enhancing diffusion of colour images,” *Image and Vision Computing*, vol. 17, pp. 201–212, 1999.

[116] M. A. U. Khan and T. M. Khan, “Fingerprint image enhancement using data driven directional filter bank,” *Optik-International Journal for Light and Electron Optics*, vol. 124, no. 23, pp. 6063–6068, December 2013.

[117] M. A. U. Khan, A. Khan, T. M. Khan, M. Abbas, and N. Mohammad, “Fingerprint image enhancement using principal component analysis (pca) filters,” in *International Conference on Information and Emerging Technologies (ICIET)*, 2010.

[118] M. Kaas and A. Witkin, “Analyzing oriented patterns,” *Computer Vision Graphics Image Processing*, vol. 37, pp. 362–385, 1987.

[119] M. A. U. Khan, T. M. Khan, O. Kittaneh, and Y. Kong, “Stopping criterion for anisotropic image diffusion,” *Optik-International Journal for Light and Electron Optics*, vol. 127, no. 1, pp. 156–160, Jan 2016.

[120] G. Hellwig, *Partial differential equations.* Teubner Stuttgart, 1977.

[121] I. Petrowski, *Vorlesungen uber partielle Differentialgleichungen.* Teubner, Leipzig, 1955.

[122] G. H. Cottet and L. Germain, “Image processing through reaction combined with nonlinear diffusion,” *Mathematics of Computation*, vol. 61, pp. 659–673, 1993.

[123] J. Weickert, *Anisotropic Diffusion in Image Processing.* ECMI Series, Teubner-Verlag, Stuttgart, Germany, 1998.

[124] A. Almansa and T. LIndeberg, “Fingerprint enhancement by shape adaptation of scale-space operators with automatic scale selection,” *IEEE Transactions on Image Processing*, vol. 9, pp. 2027–2041, 2000.

[125] M. A. K. Khan, T. M. Khan, and S. A. Khan, “Coherence enhancement diffusion using multi-scale dfb,” in *7th International Conference on Emerging Technologies*, 2011.

[126] M. Nitzberg and T. Shiota, “Nonlinear image filtering with edge and corner enhancement,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 826–833, 1992.

[127] J. Weickert, “Coherence-enhancing diffusion filtering,” *Int. J. Computer Vision,*, vol. 31, pp. 111–127, 1999.

[128] D. Chen, S. MacLachlan, and M. Kilmer, “Iterative parameter-choice and multigrid methods for anisotropic diffusion denoising,” *SIAM Journal on Scientific Computing*, vol. 33, pp. 2972–2994, 2011.

[129] A. Ilyevsky and E. Turkel, “Stopping criteria for anisotropic pdes in image processing,” *Journal of Scientific Computing*, vol. 45, pp. 333–347, 2010.

[130] G. Gilboa, N. Sochen, and Y. Zeevi, “Estimation of optimal pde-based denoising in the snr sense,” *IEEE Transactions on Image Processing*, vol. 15 (8), pp. 2269–2280, 2006.

[131] G. Gilboa, "Nonlinear scale space with spatially varying stopping time," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, pp. 2175–2187, 2008.

[132] J. Weickert, "Scale-space properties of nonlinear diffusion filtering with a diffusion tensor," Report No. 110, Laboratory of Technomathematics, University of Kaiserslautern, P.O. Box 3049, 67653 Kaiserslautern, Germany, Tech. Rep., 1994.

[133] M. Ferraro, G. Boccignone, and T. Caelli, "On the representation of image structures via scale space entropy conditions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1999.

[134] J. Sporring, "The entropy of scale-space," in *Procedding of ICPR'96*, 1996.

[135] ——, "The entropy of scale-space," Department of Computer Science / University of Copenhagen Universitetsparken 1 / DK-2100 Copenhagen East DENMARK, Tech. Rep., 1996.

[136] A. P. Witkin, "Scale space filtering," in *Proc. of International Joint Conference on Artificial Intelligence (IJCAI)*, 1983.

[137] J. J. Koenderink, "The structure of images," *Biological Cybernetics,*, pp. 363–370, 1984.

[138] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed. Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 2006.

[139] B. Zhang, X. Wu, J. You, Q. Li, and F. Karray, "Detection of microaneurysms using multi-scale correlation coefficients," *Pattern Recognition*, vol. 43, pp. 2237–2248, 2010.

[140] P. MRAZEK, “Selection of optimal stopping time for nonlinear diffusion filtering,” *International Journal of Computer Vision*, vol. 52, no. 2, pp. 189–203, 2003.

[141] C. Tsiotsios and M. Petrou, “On the choice of the parameters for anisotropic diffusion in image processing,” *Pattern Recognition*, vol. Article in Press, 2013.

[142] A. Beghdadi and A. L. Negrate, “Contrast enhancement technique based on local detection of edges,” *Conputer Vision Graphics Image Processing*, vol. 46, pp. 162–174, 1989.

[143] MATLAB, “Matlab central,” MATLAB Central - MathWorks, 20015.

[144] C. Gottschlich and C. B. S. nlieb, “Oriented diffusion filtering for enhancing low-quality fingerprint images,” *IET Biometrics*, vol. 1, pp. 105–113, 2012.

[145] K. Zuiderveld, *Contrast limited adaptive histogram equalization.* Academic Press Professional, Inc. San Diego, CA, USA, 1994.

[146] T. M. Khan, M. A. Khan, S. A. Malik, S. A. Khan, T. Bashir, and A. H. Dar, “Automatic localization of pupil using eccentricity and iris using gradient based method,” *Optics and Lasers in Engineering*, vol. 49, no. 2, pp. 177–187, 2011.

[147] T. M. Khan and Y. K. M. A.U. Khan, “Fingerprint image enhancement using multi-scale DDFB based diffusion filters and modified hong filters,” *Optik-International Journal for Light and Electron Optics*, vol. 25, pp. 4206–4214, 2014.

[148] M. T. Ibrahim, T. M. Khan, S. A. Khan, M. A. Khan, and L. Guan, “Iris localization using local histogram and other image statistics,” *Optics and Lasers in Engineering*, vol. 50, no. 5, pp. 645–654, 2012.

[149] M. T. Ibrahim, T. mehmood, M. A. khan, and L. Guan, “A novel and efficient feedback method for pupil and iris localization,” in *Image Analysis and Recognition.* Springer, 2011, pp. 79–88.

[150] A. Ross, “Iris recognition: the path forward,” *IEEE Computer*, vol. 43, no. 2, pp. 30–35, 2012.

[151] J. G. Daugman, “High confidence visual recognition of person by a test of statistical independence,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 1148–1161, 1993.

[152] R. Wildes, “Iris recognition: an emerging biometric technology,” in *IEEE*, vol. 85, 1997, pp. 1348–1363.

[153] L. Ma, T. Tan, Y. Wang, and D. Zhang, “Local intensity variation analysis for iris recognition,” *Pattern Recognition*, vol. 37, pp. 1284–1298, 2004.

[154] J. Daugman, “New methods in iris recognition,” *IEEE Transactions on Systems Man and Cybernetics Part B*, vol. 37, pp. 1167–1175, 2007.

[155] L. Liam, A. Chekima, L. Fan, and J. Dargham, “Iris recognition using self organizing neural network,” in *Student Conference on Research and Development*, 2002, pp. 169–172.

[156] T. M. Khan, Y. Kong, and M. A. Khan, “Hardware implementation of fast pupil segmentation using region properties,” in *The International Conference on Quality Control by Artificial Vision 2015.* International Society for Optics and Photonics, 2015, pp. 95 340F–95 340F.

[157] J. Daugman, “How iris recognition works,” *IEEE Trans. Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 21–30, 2004.

[158] W. Boles and B. Boashah, “A human identification technique using images of the iris and wavelet transform,” *IEEE Trans. Signal Processing*, vol. 46, pp. 1185–1188, 1998.

[159] Z. Sun, T. Tan, and Y. Wang, “Robust encoding of local ordinal measures: A general framework of iris recognition,” in *Biometric Authentication.* Lecture Notes in Computer Science, 2004, vol. 3087, pp. 270–282.

[160] L. Yu, D. Zhang, and K. Wang, “The relative distance of key point based iris recognition,” *Pattern Recognition*, vol. 40, no. 2, pp. 423–430, 2007.

[161] Z. He, T. Tan, and Z. Sun, “Iris localization via pulling and pushing,” in *18th International Conference on Pattern Recognition*, vol. 4, 2006, pp. 366–369.

[162] J. Daugman, “How iris recognition works,” in *International Conference on Image Processing*, vol. 1, 2002.

[163] K. W. Bowyer, K. Hollingsworth, and P. J. Flynn, “Image understanding for iris biometrics: a survey,” *Computer Vision and Image Understanding*, vol. 110 (2008), pp. 281–307, 2008.

[164] G. X. Z. Zhang and Y. Ma, “A novel and efficient method for iris automatic location,” *Journal of China University of Mining and Technology*, vol. 17, no. 3, pp. 441–446, 2007.

[165] S. Dey and D. Samanta, “A novel approach to iris localization for iris biometric processing,” *International Journal of Biological, Biomedical and Medical Sciences*, no. 3, pp. 180–191, 2008.

[166] F. Jan, I. Usman, and S. Agha, “Reliable iris localization using hough transform, histogram-bisection, and eccentricity,” *Signal Processing*, vol. 93, pp. 230–241, 2013.

[167] N. Wang, Q. Li, A. A. A. El-Latif, T. Zhang, and X. Niu, "Toward accurate localization and high recognition performance for noisy iris images," *Multimedia Tools Appl*, vol. 71, no. 3, pp. 1411–1430, 2014.

[168] A. Radman, K. Jumari, and N. Zainal, "Fast and reliable iris segmentation algorithm," *IET Image Processing*, vol. 7, no. 1, pp. 42–49, 2013.

[169] A. Basit and M. Y. Javed, "Iris localization via intensity gradient and recognition through bit planes," in *International Conference on Machine Vision (ICMV)*, 2007, pp. 23–28.

[170] F. Jan and I. Usman, "Iris segmentation for visible wavelength and near infrared eye images," *Optik - International Journal for Light and Electron Optics*, vol. 125, no. 6, pp. 4274–4282, 2014.

[171] T. Lindeberg, "Feature detection with automatic scale selection," *Internation Journal of Computer Vision*, vol. 30, no. 2, pp. 79–116, 1998.

[172] T. Lindberg, "Scale-spcae for discrete signals," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 234–254, 1990.

[173] F. Jan, I. Usman, and S. Agha, "A non-circular iris localization algorithm using image projection function and gray level statistics," *Optik - International Journal for Light and Electron Optics*, vol. 124, no. 18, pp. 3187–3193, 2013.

[174] "Mmu iris database," 2007. [Online]. Available: http : //www.persona.mmu.edu.my. ccteo/.

[175] "Specifications of CASIA iris image database," Chineese Academy of Sciences. [Online]. Available: www.nlpr.ia.ac.cn/english/irds/irisdatabase.htm.

[176] L. Masek and P. Kovesi, "Matlab source code for a biometric identification system based on iris patterns," The School of Computer Science and Software Engineering, The University of Western Australia, Tech. Rep., 2003.

[177] N. Otero-Mateo, M. Vega-Rodriguez, J. Gomez-Pulido, and J. Sanchez-Perez, "A fast and robust iris segmentation method," in *Pattern Recognition and Image Analysis*. Lecture Notes in Computer Science, 2007, vol. 4478, pp. 162–169.

[178] W. Yuan, Z. Lin, and L. Xu, "A rapid iris location method based on the structure of human eyes," in *27th Annual Conference of Engineering in Medicine and Biology*, 2005, pp. 3020–3025.

[179] J. Cui, Y. Wang, T. Tan, L. Ma, and Z. Sun, "A fast and robust iris localization method based on texture segmentation," *In Proc. of The SPIE*, pp. 401–408, 2004.

[180] R. D. Labati and F. Scotti, "Noisy iris segmentation with boundary regularization and reflections removal," *Image and Vision Computing*, vol. 28, no. 2, pp. 270–277, 2010.

[181] T. M. Khan, M. A. U. Khan, and O. K. Y. Kong, "Stopping criterion for linear anisotropic image diffusion: a fingerprint image enhancement case," *EURASIP Journal on Image and Video Processing*, pp. 1–20, 2016.

[182] P. Soille and L. Najman, "On morphological hierarchical representations for image processing and spatial data clustering," *Lecture Notes in Computer Science*, vol. 7356, pp. 43–67, 2012.

[183] S. Greeenberg, "Fingerprint image enhancement using filtering techniques," in *15th International Conference on Pattern Recognition, 2000.*, vol. 3, 2000, pp. 322–325.

[184] M. Qin, “A fast and low cost SIMD architecture for fingerprint image enhancement,” Master’s thesis, Department of Electrical Engineering, TUDelft, 2007.

[185] S. Vitabile, V. Conti, G. Lentini, and F. Sorbello, “An intelligent sensor for fingerprint recognition,” *Lecture Notes in Computer Science*, vol. 3824, pp. 27–36, 2005.

[186] F. Fons, M. Fons, and E. Canto, “Approaching fingerprint image enhancement through reconfigurable hardware accelerators,” in *IEEE International Symposium on Intelligent Signal Processing (WISP).* IEEE, October 2007, pp. 1–6.

[187] M. Kocevar, B. Kotnik, A. Chowdhury, and Z. Kacic, “Real-time fingerprint image enhancement with a two-stage algorithm and blocklocal normalization,” *Journal of Real-Time Image Processing*, 2014.

[188] J. Lee, “Digital image enhancement and noise filtering by use of local statistics,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 2, no. 2, pp. 165–168, 1980.

[189] M. Haidekker, *Advanced Biomedical Image Analysis.* Wiley, 2010.

[190] S. S. Singh, T. T. Singh, H. M. Devi, and T. Sinam, “Local contrast enhancement using local standard deviation,” *International Journal of Computer Applications*, vol. 47, pp. 31–35, 2012.

[191] M. Sepasian, W. Balachandran, and C. Mares, “Image enhancement for fingerprint minutiae-based algorithms using clahe, standard deviation analysis and sliding neighborhood,” in *Proceedings of the World Congress on Engineering and Computer Science*, 2008, pp. 978–988.

[192] J. He and J. Bao, "Normalization of finmgerprint image using the local feature," in *Internationl Conference on Computer Science and Service System*, 2012, pp. 1643–1646.

[193] P. Singh and A. K. Garg, "Morphology based non uniform background removal for particle analysis: A comparative study," *International Journal of Computing and Corporate Resaerch*, vol. 1, no. 3, 2011.

[194] L. Dragut, C. Eisank, and T. Strasser, "Local variance for multi-scale analysis in geomorphometry," *Geomorphology*, vol. 130, pp. 162–172, 2011.

[195] J. Weickert, "Coherence-enhancing diffusion filtering," *Int. J. Computer Vision,*, vol. 31, pp. 111–127, 1999.

[196] B. Zhang, S. Zhang, J. Zhang, and X. Jing, "A method of the region of intrest extraction based on orientation entropy," in *International Conference on Broadband Network & Multimedia Technology (IC-BNMT)*, 2011, pp. 664–669.

[197] D. G. Bailey, *Design for Embedded Image Processing on FPGAs.* John Wiley & Sons, Ltd., 2011.

[198] R. Zoss, A. Habegger, V. Bandi, J. Goette, and M. Jacomet, "Comparing signal processing hardware-synthesis methods based on the Matlab tool-chain," in *6th International Symposium on Electronic Design, Test and Applications, Queenstown, New Zealand*, 2011, pp. 281–286.

[199] F. Fons, M. Fons, E. Canto, and M. Lopez, "Flexible hardware for fingerprint image processing," in *Microelectronics and Electronics Conference, PRIME*, 2007, pp. 169–172.

[200] S. Pankanti, S. Prabhakar, and A. K. Jain, "On the individuality of fingerprints," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 1010–1025, 2001.

[201] A. M. Bazen, G. T. B. Verwaaijen, S. H. Gerez, L. P. J. Veelenturf, and B. J. van der Zwaag, "A Correlation-Based Fingerprint Verification System," 2000.

[202] T. M. Khan, D. G. Bailey, M. A. Khan, and Y. Kong, "Efficient hardware implementation for fingerprint image enhancement using anisotropic gaussian filter," *IEEE Transactions on Image Processing*, 2016.

[203] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, and A. K. Jain, "FVC2002: Second fingerprint verification competition," in *16th International Conference on Pattern Recognition*, vol. 3, 2002, pp. 811–814.

[204] "American national standard institute. common biometric exchange file format, (http://www.itl.nist.gov/div895/isis/bc/cbeff/)." [Online]. Available: http://www.itl.nist.gov/div895/isis/bc/cbeff/

[205] U. Watchareeruetai, K. Suppasriwasuseth, and S. Tantaratana, "Separable gabor filter realization for fast fingerprint enhancement," in *IEEE International Conference on Image Processing, ICIP*, 3, Ed., 11-14 Sept 2005, pp. III – 253–6.

[206] V. Areekul, U. Watchareeruetai, and S. Tantaratana, "Fast separable Gabor filter for fingerprint enhancement," in *Proceeding International Conference on Biometric Authentication (ICBA2004) LNCS3072, Springer*, 2004., pp. 403–409.

[207] J. M. Geusebroek, A. W. Smeulders, and J. van de Weijer, "Fast anisotropic Gauss filtering." *IEEE Transactions on Image Processing*, vol. 12, no. 8, pp. 938–43, 2003.

[208] M. F. F. Fons and E. Canto, “Fingerprint image processing acceleration through run-time reconfigurable hardware,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 57, no. 12, pp. 991–995, 2010.

[209] M. L. Garcia and E. F. C. Navarro, “FPGA implementation of a ridge extraction fingerprint algorithm based on microblaze and hardware coprocessor,” in *International Conference on Field Programmable Logic and Applications*, 2006, pp. 1–5.

[210] S. Kawada and T. Maruyama, “An approach for applying large filters on large images using FPGA.” in *International Conference on Field Programmable Technology, Kitakyushu, Japan*, 2007, pp. 201–208.

[211] A. Joginipelly, A. Varela, D. Charalamapidis, R. Schott, and Z. Fitzsimmons, “Efficient FPGA implementation of steerable Gaussian smoothers,” in *44th IEEE Southeastern Symposium on System Theory*, 2012, pp. 78–83.

[212] Y. Wang, J. Hu, and F. Han, “Enhanced gradient-based algorithm for the estimation of fingerprint orientation fields,” *Applied Mathematics and Computation*, vol. 185, pp. 823–833, 2007.

[213] J. Yang, N. Xiong, and V. Vasilakos, “Two-stage enhancement scheme for low-quality fingerprint images by learning from the images,” *IEEE Transactions on Human-Machine Systems*, vol. 43, no. 2, pp. 235 – 248, March 2012.

[214] J. Daugman, “Probing the uniqueness and randomness of iriscodes: Results from 200 billion iris pair comparisons,” *Proceedings of the IEEE*, vol. 94, no. 11, pp. 1927–1935, Nov 2006.

[215] L. M., Daugman, J., and C. E., “Hardware-software co-design of an iris recognition algorithm,” *Information Security, IET*, vol. 5, no. 1, pp. 60–68, March 2011.

[216] Y. Tian, T. Kanade, and J. Cohn, “Dual-state parametric eye tracking,” in *Face and Gesture Recognition*, 2000, pp. 110–115.

[217] Cheung and R. C. Frecker, “Estimation of ocular torsion with dynamic changes in pupil size,” in *Proc. of IEEE Eng. in Med. Bid. 16th Annu*, vol. 2, 1994, pp. 924–925.

[218] J. Kim and K. Park, “An image processing method for improved pupil size estimation accuracy,” in *Proc. of the 25th Annual International Conference of the IEEE EMBS*, vol. 2, 2003.

[219] Iskander, D. Robert, and C. etc, “Automatic pupillometry from digital images,” in *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 9, 2004, pp. 1619–1627.

[220] S. I. Kim, D. Lee, S. Y. Kim, O. S. Kwon, and J.Cho, “An algorithm to detect a center of pupil for extraction of point of gaze,” in *Proc. of the 26th Annual International Conference of the IEEE EMBS*, 2004.

[221] C. B. and Solomon, “S. dictionary of eye terminology.” in *Triad Publishing Company*, Gainsville, Florida, 1990.

[222] C. Tisse, L. Martin, L. Torres, and M. Robert., “Person identification technique using human iris recognition,” in *International Conference on Vision Interface, Canada*, 2002.

[223] X. Bai, L. Wenyao, and etc., “Research on iris image processing algorithm,” *Journal of Optoelectronics Laser*, vol. 14, pp. 741–744, 2003.

[224] M. T. Ibrahim, , T. M. Khan, M. A. khan, and G. Ling, “Automatic segmentation of pupil using local histogram and standard deviation,” in *Visual Communications*

and Image Processing 2010. International Society for Optics and Photonics, 2010, pp. 77 442S–77 442S.

[225] M. Karaman and A. Atalar, "Design and implementation of a general- purpose median filter unit in cmos vlsi," *IEEE Journal of Solid- State Circuits*, vol. 25, no. 2, April 1990.

[226] H.-S. Yu, J.-Y. Lee, and J.-D. Cho, "A fast vlsi implementation of sorting algorithm for standard median filters," in *12th Annual IEEE International ASIC/SOC Conference Proceeding*, 1999.

[227] M. Vega-Rodriguez, J. Sanchez-Perez, and J. Gomez-Pulido, "An fpga-based implementation for median filtering meeting the real- time requirements of automated visual inspection systems," in *Proceedings of the 10th Mediterranean Conference on Control and Automation*, 2002.

[228] S. Fahmy, P. Cheung, and W. Luk, "High-throughput one- dimensional median and weighted median filters on fpga," *IET Comput. Digit. Tech*, vol. 3, no. 4, pp. 384–394, 2009.

[229] M. Gundam and D. Charalampidis, "Median filter on fpgas," in *System Theory (SSST), 2012 44th Southeastern Symposium on*, March 2012, pp. 83–87.

[230] A. Rosenfeld and J. Pfaltz, "Sequential operations in digital picture," *Jounal of the ACM*, vol. 13, no. 4, pp. 471–494, 1966.

[231] M. Jablonski and M. Gorgon, "Handel-c implementation of classical component labelling algorithm," *Euromicro Symposium on Digital System Design*, pp. 387–393, 2004.

[232] H. Alnuweiti and V. Prasanna, “Parallel architectures and algorithms for image component labeling,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 10, pp. 1014–1034, 1992.

[233] D. Crookes and K. Benkrid, “Fpga implementation of image component labelling,” *Reconfigurable Technology: FPGAs for Computing and Applications, SPIE*, vol. 3844, pp. 17– 23, 1999.

[234] K. Benkrid, S. Sukhsawas, D. Crookes, and A. Benkrid, “An fpga-based image connected component labeller,” *Field-Programmable Logic and Applications. Springer Berlin*, pp. 1012–1015, 2003.

[235] D. Bailey and C. Johnston, “Single pass connected components analysis,” *Image and Vision Computing*, 2007.

[236] S. Ajith, S. Bandarupalli, and M. Borgaonkar, “Image processing using fpga,” Maseeh College of Engineering and Computer Science at PSU, Tech. Rep., 2009.

[237] J. Daugman, “Biometric personal identification system based on iris analysis,” USA A Patent US5 291 560 A, 1994.

[238] Y. Chen, M. Adjouadi, C. Han, J. Wang, A. Barreto, N. Rishe, and J. Andrian, “A highly accurate and computationally efficient approach for unconstrained iris segmentation,” *Image and Vision Computing*, vol. 28, no. 2, pp. 261 – 269, 2010, segmentation of Visible Wavelength Iris Images Captured At-a-distance and On-the-move. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0262885609000833

[239] A. Ross and S. Shah, “Segmenting non-ideal irises using geodesic active contours,” in *Biometric Consortium Conference, 2006 Biometrics Symposium: Special Session on Research at the*, Sept 2006, pp. 1–6.

[240] T. M. Khan, D. G. Bailey, M. A. U. Khan, and Y. Kong, “Real-time edge detection and range finding using FPGAs,” *Optik-International Journal for Light and Electron Optics*, vol. 126, no. 17, pp. 1545–1550, 2015.

[241] H. Ngo, J. Shafer, R. Ives, R. Rakvic, and R. Broussard, “Real time iris segmentation on fpga,” in *Application-Specific Systems, Architectures and Processors (ASAP), 2012 IEEE 23rd International Conference on*, July 2012, pp. 1–7.

[242] H. Al-Mamory, “Iris Detection Using Morphology,” *Journal of Babylon University/Pure and Applied Science*, vol. 22, no. 9, pp. 2277–2282, 2014.

[243] Y. Chen, Y. Liu, and X. Zhu, “Robust iris segmentation algorithm based on self-adaptive chanvese level set model,” *JOURNAL OF ELECTRONIC IMAGING*, vol. 24, no. 4, pp. 043 012–1 –043 012–12, 2015.

[244] S. Zhang, G. Hou, and Z. Sun, “Eyelash removal using light field camera for iris recognition,” *Lecture Notes in Computer Science*, vol. 8833, pp. 319–327, 2014.

[245] M. A. M. Ali and N. M. Tahir, “Half iris gabor based iris recognition,” in *IEEE 10th International Colloquium on Signal Processing & its Applications (CSPA).* Kuala Lumpur: IEEE, 2014, pp. 282 –287.

[246] D. G. Bailey, “Efficient implementation of greyscale morphological filters,” in *International Conference on Field Programmable Technology*, Beijing, China, 2010, pp. 421–42.

[247] A. Rosenfeld and J. Pfaltz, “Sequential operations in digital picture processing,” *Journal of the ACM*, vol. 13, no. 4, pp. 471–494, 1996.

[248] M. J. Klaiber, D. G. Bailey, Y. O. Baroud, and S. Simon, "A resource-efficient hardware architecture for connected components analysis," *IEEE Transactions on Circuits and Systems for Video Technology*, 2015.

[249] R. Y. F. Ng, Y. H. Tay, and K. M. Mok, "Iris verification algorithm based on texture analysis and its implemenation on dsp," in *Intl. Conference on Signal Acquisition and Processing (ICSAP2009)*, 2009, pp. 198–202.

[250] F. J. Giacometto, J. M. Vilardy, C. O. Torres, and L. Mattos, "Design and implementation of an algorithm for creating templates for the purpose of iris biometric authentication through the analysis of textures implemented on a fpga," in *Journal of Physics*, ser. 1-13, vol. 274, no. 1, 2011, pp. J: Conference Series, vol. 274, no. 1, 2011, pp. 1–13.

[251] M. A. U. Khan, T. M. Khan, D. G. Bailey, and Y. Kong, "A spatial domain scar removal strategy for fingerprint image enhancement," *Pattern Recognition*, vol. 60, pp. 258–274, 2016.

[252] T. M. Khan, D. G. Bailey, M. A. Khan, and Y. Kong, "Efficient hardware implementation strategy for local normalization of fingerprint images," *Journal of Real-Time Image Processing doi:10.1007/s11554-016-0625-8*, 2016.

[253] S. A. Sudiro, "Thinning algorithm for image convertion in fingerprint recognition system," National Seminar Sodt-Computing Inteligent System and Information Technology, Universitas Kristen Petra, Surabaya, Tech. Rep., 2005.

[254] S. A. Sudiro and R. T. Yuwono, "Adaptable fingerprint minutiae extraction algorithm based-on crossing number method for hardware implementation using FPGA." *International Journal of Computer Science, Engineering and Information Technology (IJCSEIT)*, vol. 2, no. 3, pp. 1–30, 2012.

[255] G. Stockman and L. G. Shapiro, *Computer Vision.* Prentice Hall PTR Upper Saddle River, NJ, USA, 2001.

[256] T. M. Khan, D. G. Bailey, Y. Kong, and M. A. K. Khan, "Real-time implementation of fast iris segmentation and normalization on fpga," *submitted in IEEE Transactions on Human-Machine Systems*, 2016.

[257] A. Lindoso, L. Entrena, and J. Izquierdo, "Fpga-based acceleration of fingerprint minutiae matching," in *3rd Southern Conference on Programmable Logic*, 2007, pp. 81–86.

[258] P. S. Patil and A. S. Abhyankar, "Multimodal biometric identification system based on iris & fingerprint," *IOSR Journal of VLSI and Signal Processing*, vol. 1, no. 6, pp. 76–83, 2013.

[259] S. Prabhakar, A. K. Jain, and J. Wang, "Microaneurysm minutiae verification and classification," Dept. Comput. Eng. Sci. Univ. Michigan State, Tech. Rep., 1998.

[260] N. K. Ratha, R. M. Bolle, V. D. Pandit, and V. Vaish, "Robust fingerprint authentication using local structural similarity," in *5th IEEE workshop Appl. Comput. Vis*, 2000, pp. 29–34.