High-Quality 3D Fingerprint Generation: Merging Skin Optics, Machine Learning and 3D Reconstruction Techniques

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science
in
Electronics and Communication Engineering by Research

by

Apoorva Srivastava
2019702014
apoorva.srivastava@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500032, INDIA
August 2023

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled ”Split and Knit Algorithm for 3D fingerprint reconstruction using Single Camera” by Apoorva Srivastava, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date	Adviser: Dr. Anoop M. Namboodiri

To UNIVERSE

Acknowledgments

I express my heartfelt gratitude to my esteemed advisor, Dr. Anoop Namboodiri, for his invaluable support and deep insights in the fields of computer vision, biometrics, and computational photography. His guidance and encouragement have been instrumental in shaping my research work and adopting a holistic approach to solving complex problems. I am also indebted to the research center CVIT and the newly founded BaSIL lab for providing a stimulating and growth-oriented environment to carry out my research work. My experience at IIIT Hyderabad has been incredibly enriching, thanks to its innovative culture that fosters enthusiasm and zeal for research. I cannot overstate my gratitude to my family for their unwavering support and encouragement throughout my research journey. My mother, in particular, played the role of a sounding board, patiently listening to my research ideas. I also wish to express my heartfelt appreciation to my life partner, Pranjal Pandey, whose unwavering support and encouragement were critical in completing my research work. Finally, I would like to extend my sincere thanks to all my friends who have been a constant source of inspiration and support. Their presence during difficult times has been invaluable, and their encouragement has helped me stay focused and motivated. I feel incredibly fortunate to have such a supportive network of friends and family.

Abstract

Fingerprints are a widely recognized and commonly used method of identification. Contact-based fingerprints, which involve pressing the finger against a surface to obtain images, are a popular method of capturing fingerprints. However, this process has several drawbacks, including skin deformation, unhygienic conditions, and high sensitivity to the moisture content of the finger. These factors can negatively impact the accuracy of the fingerprint. Moreover, fingerprints are three-dimensional anatomical structures, and two-dimensional fingerprints do not capture the depth information of the finger ridges. While $3D$ fingerprint capture is less sensitive to skin moisture levels and avoids skin deformation, it is limited in adoption due to the high cost and system complexity associated with it. The complexity and cost are mainly attributed to the use of multiple cameras, projectors, and sometimes synchronously moving mechanical parts.

Photometric stereo offers a promising solution to build low-cost, simple sensors for high-quality 3D capture using only a single camera and a few LEDs. However, the method assumes that the surface being imaged is lambertian, which is not the case for human fingers. Existing 3D fingerprint scanners based on photometric stereo also assume that the finger is lambertian, resulting in poor reconstruction results.

In this context, we introduce the Split and Knit algorithm (SnK), a 3D reconstruction pipeline based on Photometric Stereo for finger surfaces. The algorithm splits the reconstruction of the ridge-valley pattern and finger shape and combines them to obtain the 3D fingerprint reconstruction for the full finger with a single camera for the first time. To reconstruct the ridge-valley pattern, SnK introduces an efficient way of estimating the direct illumination component by using a trained U-Net without extra hardware, which reduces the non-Lambertian nature of the finger image and enables a higher-quality reconstruction of the entire finger surface. To obtain the finger shape using a single camera, the algorithm introduced two novel approaches, a) using IR illumination and b) using a mirror and parametric modeling for the finger shape. Finally, we combine the overall finger shape and the ridge-valley point cloud to obtain a 3D finger phalange. The high-quality 3D reconstruction results in better matching accuracy of the captured fingerprints. Splitting the ridge-valley pattern from the finger provides an implicit way to convert 3D fingerprint into 2D fingerprint, making the SnK algorithm compatible with the 2D fingerprint recognition systems. To apply the SnK algorithm to fingerprints, we designed a 3D printed photometric stereo-based setup that captures contactless finger images and obtains their 3D reconstructions.

Contents

Chapter Page

List of Figures

Chapter 1

Introduction

In this age of technological advancements and a constantly increasing population, it is crucial to assign a unique identity to each individual across multiple devices and platforms. While identity authentication methods such as ID cards and secret passwords are still in use, they come with an inherent risk of misplacement or duplication. One of the basic ways to identify and recognize any object is by observing its unique characteristics, and humans are no exception. These unique characteristics are known as biometrics, and over the years, several traits have been established as possessing the quality of unique human identification, such as fingerprint, face, iris, palmprint, DNA, palmvein, finger vein, hand geometry, voice, gait, sclera, periocular, EEG, and ECG [11]. Among these, fingerprint, face, and iris are the most popular forms of biometrics. The popularity of a biometric trait is defined based on factors such as universality, distinctiveness, permanence, collectability, performance, acceptability, and circumvention [11]. Fingerprint, being the oldest and most popular form of biometrics, qualifies all these factors.

1.1 Fingerprint Biometrics

Fingerprints are developed in the fetus due to the force acting on the base of the skin epidermis (basal layer). They are believed to be unique patterns, and given the world population of 7 billion and ten fingers of each individual, we have 70 billion unique patterns [26]. Fingerprints have been identified and utilized as a unique bio-signature long before the design of Automatic fingerprint recognition systems (Including the figure of ink and optical fingerprints). Automated Fingerprint Identification System (AFIS) is a biometric solution consisting of a computer database of fingerprint records, which can search and compare them to identify known or unknown fingerprints. Modern AFISes can search over a billion fingerprint records in a single second. It elevates the need for human experts and makes fingerprint recognition scalable according to the human population. The traditional way of fingerprint acquisition required ink to be applied to the finger and pressing the finger against a surface. After the advent of AFIS, the ink was replaced by optical/capacitive signals, but pressing the finger against a surface re-

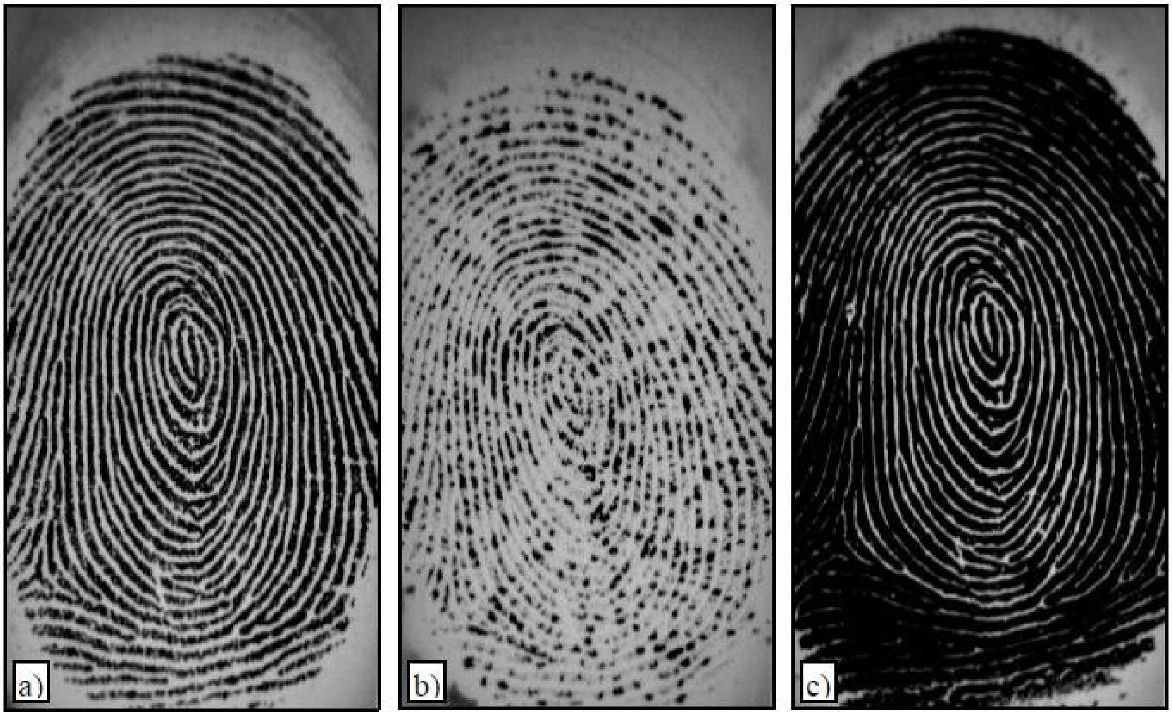


Figure 1.1 Appearance of fingerprint of a finger under different moisture content of skin. a) Fingerprint of normal moisture content finger skin, b) fingerprint of dry finger skin, c) fingerprint of moist finger skin. The figure is taken from [8]

mained the same. The acquisition process of the fingerprints is the weakest point of the 2D fingerprint systems.

1.2 Contactless Fingerprint Acquisition

Before designing a fingerprint scanner, we tried to understand the characteristics of the ideal fingerprint scanner should be as follows:

1. **Contact-less capture ability** Contact-less capture has following advantages over the contact based capture of the fingerprints-

   a) The Contact based capture introduces non-uniform pressure leading to elastic deformation of the skin, which eventually distorts the image of the ridge-valley pattern.

   b) The fingerprint captured depends on the skin condition. The fingerprint captured for a wet finger differs from that captured for a dry finger surface as shown in figure 1.1. The figure is taken from [8] *c)* The contactless capture also prevents the dirt and sweat imprints on the scanner along with transmissions of germs. Due to added hygiene and protection, this is a desirable quality of an ideal scanner

2. Simple Compact and Low-Cost The bulky nature, high cost and complexity of any device which is required for the day to day use, reduces its popularity and prevents commercialization at scale.

3. Full-Rolled fingerprint The information contained in a full-rolled fingerprint is much more than frontal fingerprint. Therefore, the ideal fingerprint capture device should be able to capture the rolled fingerprint for better matching accuracy.

4. Depth information Ridge-valley of the fingerprint is the 3D anatomical structure, when pressed against a surface, leaves the projection of the frontal view of the finger surface. Hence the finger shape and ridge-valley depth information is lost in the capturing process. So the ideal fingerprint solution should be able to acquire the available depth information along with fingerprint pattern.

5. High Quality The quality of the fingerprint captured directly relates to the accuracy of its recognition. So it is a must have property for an ideal fingerprint scanner.

The utilization of 3D fingerprint reconstruction presents a promising solution to address the limitations of traditional 2D fingerprint identification methods. The additional depth and shape information provided by 3D fingerprints can help alleviate the issues associated with skin deformation, hygiene, and moisture content, ultimately improving the accuracy and reliability of fingerprint recognition. Moreover, the conversion of 3D fingerprints into 2D fingerprints enables the compatibility of 3D fingerprint recognition with existing 2D fingerprint recognition pipelines. The availability of algorithms for matching and recognizing 3D fingerprints provides an opportunity to integrate these techniques with traditional 2D matching algorithms, further enhancing the overall accuracy and robustness of fingerprint recognition systems. The major challenge in adopting 3D fingerprint scanners is the bulky nature of the scanner and its high cost. Few examples of the existing 3D fingerprint scanners are shown in the figure 1.2. These limitations exist due to multiple cameras and projectors for 3D fingerprint reconstruction. The Photometric Stereo is the only 3D fingerprint technique that provides the opportunity to reconstruct the finger using a single camera and a few LEDs.

1.3 3D fingerprint using Photometric Stereo

Photometric stereo is a technique in computer vision for estimating the surface normals of objects by observing that object under different lighting conditions. It is based on the fact that the amount of light reflected by a surface depends on the surface's orientation in relation to the light source and the observer; hence if we know each light direction, then the surface normal can be determined using a single camera and multiple light directions. Photometric Stereo provides the most portable and lowest cost 3D fingerprint reconstruction setup. It uses a single camera and LEDs, while other 3D reconstruction

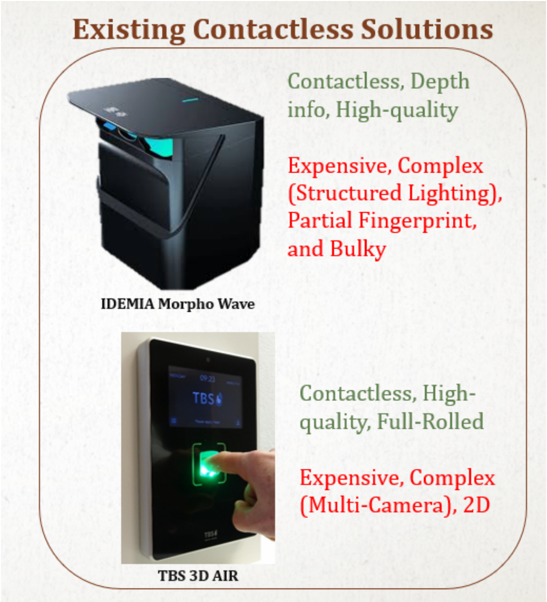
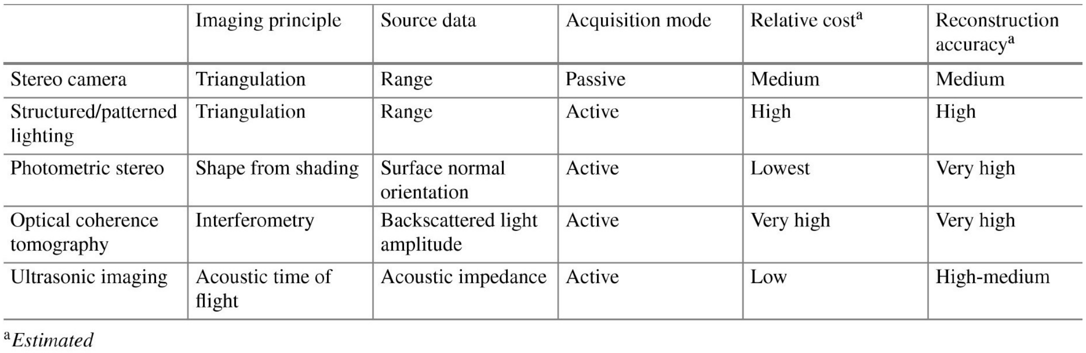

Existing Contactless Solutions
Contactless, Depth info, High-quality
Expensive, Complex (Structured Lighting), Partial Fingerprint, and Bulky
IDEMIA Morpho Wave
Contactless, High-quality, Full-Rolled
Expensive, Complex (Multi-Camera), 2D
TBS 3D AIR

Figure 1.2 Commercially available contactless fingerprint solutions

methods use multiple cameras and projectors Etc. As mentioned in $Table$ 2.1 of [14], as shown in the figure 1.3. Photometric Stereo also provides the highest quality of 3D reconstruction. It is possible because of the implicit correspondence in subsequent images captured for Photometric Stereo. Hence, to design a portable $3D$ fingerprint setup, we choose Photometric Stereo as a suitable $3D$ reconstruction technique.

Despite its advantages, Photometric Stereo has a significant drawback in that it assumes the imaged surface to be Lambertian, which reflects light equally in all directions. This assumption makes it vulnerable to even the slightest illumination changes, leading to high-frequency illumination noise. Moreover, the finger surface is non-Lambertian, and it exhibits sub-surface scattering and specular reflection. Sub-



	Imaging principle	Source data	Acquisition mode	Relative cost[a]	Reconstruction accuracy[a]
Stereo camera	Triangulation	Range	Passive	Medium	Medium
Structured/patterned lighting	Triangulation	Range	Active	High	High
Photometric stereo	Shape from shading	Surface normal orientation	Active	Lowest	Very high
Optical coherence tomography	Interferometry	Backscattered light amplitude	Active	Very high	Very high
Ultrasonic imaging	Acoustic time of flight	Acoustic impedance	Active	Low	High-medium

[a] *Estimated*

Figure 1.3 The Table compares the contactless $3D$ fingerprint solutions as given in the book [14]. Photometric Stereo gives the highest quality of reconstruction and the lowest cost.

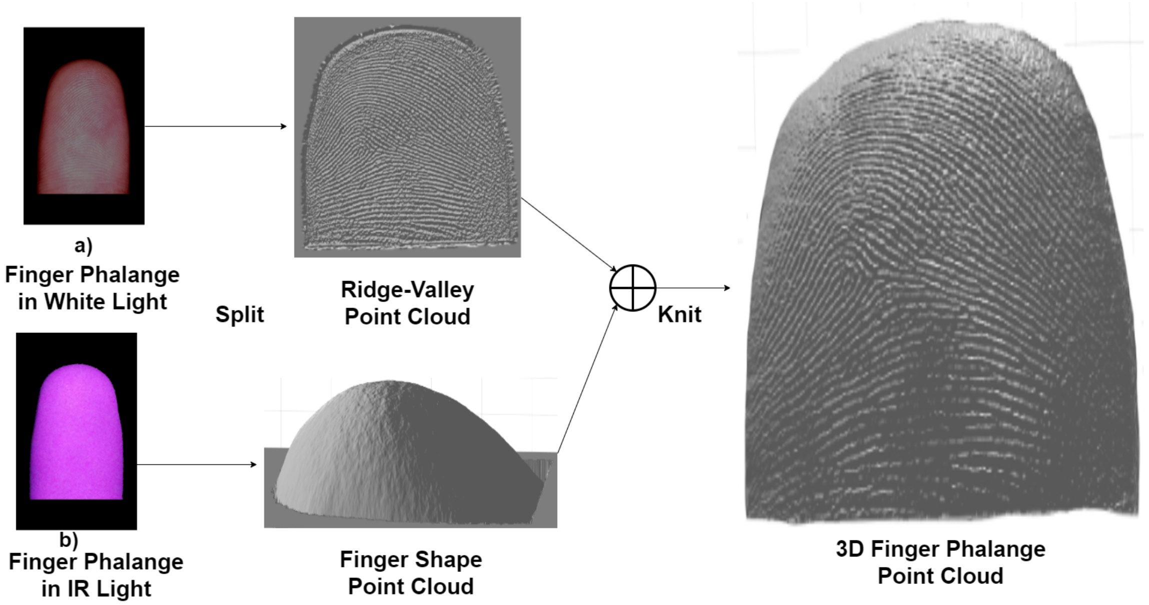


Figure 1.4 The **SnK** algorithm uses a single camera and multiple LEDs to reconstruct a detailed finger phalange point cloud. a) The visible spectrum image is transformed into a lambertian image using a trained U-Net to obtain the ridge-valley point cloud using photometric stereo. b) NIR(Near-Infrared) images obtained with the same camera are used for the finger shape. The proposed algorithm thus *splits* the finger information into two parts: the lambertian component of the visible spectrum and the NIR spectrum image, containing the ridge-valley pattern and the overall finger shape, respectively. c) The algorithm then *knits* together the recovered 2.5D images to obtain the complete 3D finger phalange shape.

surface scattering occurs when light enters the finger skin, gets scattered within the finger surface, and emits out at a different location, giving a reddish appearance to the finger. Specular reflection occurs when a significant amount of light gets reflected in a single direction from the finger surface, giving it a shiny appearance at that particular point. Despite these known properties of the finger surface, current sensors based on Photometric Stereo assume the finger surface to be Lambertian, leading to fingerprint reconstruction only for a finger patch without capturing the finger shape. Further research is required to develop techniques that can handle non-Lambertian surfaces to improve the accuracy and robustness of 3D fingerprint reconstruction.

1.4 Motivation and Contributions

Our research work, aims to resolve the problem of reconstructing a 3D finger with the Photometric Stereo. The major limitation of the finger surface is its non-Lambertian nature. We propose the Split and Knit algorithm, a $3D$ reconstruction pipeline, which makes finger images suitable for applying the Photometric Stereo. The Split and Knit algorithm does not assume the finger surface to be Lambertian and hence tries to reduce the non-Lambertian nature of the finger image. It helps obtain a better quality ridge-valley point cloud for the whole finger surface rather than a patch. The algorithm reduces the non-Lambertian nature of the finger images by applying CLAHE enhancement followed by global-direct component separation. The global-direct component is separated in a novel way using a trained U-Net. Also, it introduces two novel methods to obtain the finger shape using a single camera. Further, it adds the ridge-valley point cloud and the finger shape to obtain the $3D$ finger phalange. Figure 1.4 explains the Split and Knit algorithm in a nutshell. The shape of the finger can be obtained in two novel ways using a single camera: 1) Using IR light 2) Using mirrors and parametric modeling. The first way is used in the Split and Knit algorithm. Using the second method, we try to obtain a rolled-equivalent 3D finger using a single camera.

Contributions: The significant contributions of our work are as follows-

1. We extract the ridge-valley pattern from the finger and re-mix it with the finger shape to obtain the $3D$ finger phalange with detailed fingerprint and un-distorted overall shape, all by a single camera for the first time.

2. We introduce global-direct component separation using a trained U-Net without extra hardware, which reduces the non-Lambertian nature of the finger image and helps obtain a detailed ridge-valley pattern.

3. We enable the acquisition of the overall finger shape using a single camera via two novel approaches a) using IR illumination and b) using a mirror and parametric modeling for the finger shape.

4. We created a Photometric Stereo based acquisition setup with mirror for capturing the finger images for 3D fingerprint reconstruction.

5. Splitting the ridge-valley pattern from the finger provides us an implicit way to convert 3D fingerprint into 2D fingerprint making the Split and Knit algorithm back compatible with the 2D fingerprint recognition systems.

1.5 Thesis Organization

The thesis is organized as follows:

1. Chapter 2 details the literature review for the 3D fingerprint reconstruction techniques. Further the existing techniques to separate Global Direct component of an image are discussed.

2. Chapter 3 details the Split and Knit algorithm.

3. Chapter 4 details about the reconstruction of 3D fingerprint setup for Split and Knit Algorithm and rolled-equivalent finger reconstruction. This chapter also details the code pipeline for the setup.

4. Chapter 5 presents the Experiments and Results

5. Chapter 6 provides the Conclusion and Future Works.

Chapter 2

Related Works

The research works carried out for obtaining the contactless fingerprints can be broadly classified into two categories:

1. 2D Contactless Fingerprint Systems

2. 3D Contactless Fingerprint Systems

Under 2D Contactless Fingerprint systems we obtain the fingerprint in image format similar to the traditional 2D fingerprint method. The only difference lies in the fact that the fingerprint is captured via a camera from a distance rather than pressing the finger against a surface. This technique elevates the hygiene and the deformation issues of traditional 2D fingerprint scanners. But, this method does not capture the additional depth information of the ridge-valley and doesn't produce the fingerprint in a 3D format. This brings us to the second category of 3D fingerprint contactless systems. Here we use 3D reconstruction techniques to get the fingerprint in the 3D format along with the finger and also they are contactless.

2.1 2D Contactless Fingerprint Systems

2D Contactless fingerprint systems are dependent on the optical devices like camera and lenses. They can be further classified as:

1. Smartphone Camera based systems

2. Digital Camera based systems

2.1.1 Smartphone Camera based systems

The smartphone based systems try to capture a high quality finger image directly and match them using the 2D fingerprint matching algorithms.

Derawi *et.al*[3] show a biometric performance with an Equal Error Rate (EER) of 4.5% by applying a commercial extractor/comparator and without any pre-processing on 1320 finger images. They highlighted the effect of simple and complex backgrounds on the quality of finger image captured. The quality of the finger image also depended on the fact whether the flash was used during the capture or not. Again the benefits of flash usage depended on the type of the background. They concluded that in a plain white background with or without flash, the information is not lost but differs a lot for the two cases. Moving from a white background to a more pattern based background they observed a smaller change in the quality for with and without flash images. They also discussed the role of geometric distortion due to possibility of multiple positions for the finger placement. We have utilized their insights over the effect of background and finger placement in designing of 3D fingerprint setup.

Stein$et.\ al.$ [33] showed that the usage of flash in the dark environment removed much of the finger image's noise and improved the quality for fingerprint matching. Their work focused on overcoming these challenges and developing a complete authentication system for a mobile operating system based on fingerphotos. They developed a capture process for fingerphotos in combination with efficient post-processing algorithms. The implemented algorithms provide a very good detection and segmentation of the finger and ensure the suitability of the captured photos for finger photo recognition. Also, the capture process allows the user to present the finger to the camera at any desired orientation angle. However, the quality of the fingerphoto hugely depended on the capture device and the matching accuracy could have been improved if the minutiae extractors optimized for the fingerphotos are developed.

Sankaran $et.\ al.$ [30] majorly focused to aid the matching process of smartphone captured fingerprints and to attenuate the effect of capture variations. Further, they created a publicly available smartphone fingerphoto database having three different subsets addressing the challenges of environmental illumination and background, along with their corresponding live scan fingerprints. They categorized major challenges in fingerphoto matching into three categories and provided solution for each of them:

1. Preprocessing and segmentation: Fingerphoto images can be captured at any time and at any place. This leads to the challenge of variations in background and illumination. To this they provided an algorithm for fingerphoto segmentation and enhancement.

2. Feature representation: Due to the unconstrained nature of image capture, the representation should be in-variant to translation, rotation, and pose. Hence they proposed a novel Scattering Network(ScatNet) based feature representation

3. Database: The lack of publicly available fingerphoto database hence they created a public fingerphoto database to study and analyze the two important challenges affecting fingerphoto recognition: (a) illumination variations and (b) complex background information.

2.1.2 Digital Camera based systems

Compared with smartphone based acquisitions, digital camera based acquisitions are more flexible in system design. In digital camera-based systems, white or color LEDs are usually utilized to provide predicable lighting. Tsai *et.al.* [36] propose adapting a digital variable-focus liquid lens to achieve fast scanning at pre-elected focal planes. The compact touchless fingerprint reader uses a combination of the camera module, digital liquid lens, and LED illumination to capture fingerprint images. To achieve high minutiae match accuracy, the reader needs to capture fingerprints with the same image magnification, which requires a narrow depth of field (DOF) feature. The digital liquid lens enables the reader to scan multiple focal planes to achieve a narrow DOF while also extending the image capture range. This feature also helps to circumvent the problem of locating the position of a floating finger. The prototype of the touchless fingerprint reader was demonstrated, and it includes a digital liquid lens mounted on a commercial 2MP webcam module with a LED illustrator. The digital liquid lens operates on the electrostatic principle, where the liquid-liquid interface movement actuated by electric forces balances at the edge of the electrode. The AC electrode arrangement involves connecting one electrode to the alcohol mixture, and the digital liquid lens has better operation characteristics than the analogy operation of the conventional liquid lens. The response time of the digital liquid lens is fast and stable due to its controllable characteristics, and over-driving voltage is a convenient method to fast up the response without losing focal length change accuracy. The lens module has seven presets of focal length, and its mechanical dimensions are 19 mm in diameter and 3.8 mm in thickness. The response time is smaller than 6ms while driving voltage is at 30 Vrms of 1 kHz in square wave. The calculated MTF of the lens module for a compact fingerprint reader is also provided. The compact touchless fingerprint reader can be manufactured in the size of 9.89.85 (mm) based on the optical design of the lens module. The proposed touchless fingerprint reader offers a compact and efficient solution for fingerprint recognition, using a digital liquid lens and LED illumination to scan at multiple focal planes. The prototype demonstrated the feasibility of the reader, and its compact size makes it suitable for mobile applications.

Weissenfeld *et. al.* [38] designed a handheld device for biometric identification. The device was used to capture the fingerphoto of 4 fingers and the image of the face. They designed a fast algorithm for the four finger segmentation and determining the quality of the fingerprint. Further after the enhancement of the fingerprint, it was matched using traditional $2D$ fingerprint matching methods.

Further Genovese *et. al.* [7] designed a digital camera based device for the capture of fingerprint pores. It was the first time that the third level features of the fingerprint, *i.e.* fingerprint pores were extracted using a touchless sensor. The features were extracted using the novel sequence of steps using standard image processing and computational intelligence techniques. They proved that the third level of fingerprint features can be extracted using fingerphoto.

Khodadoust *et. al.* [12] designed a device to capture three modalities of a finger using the digital camera. They performed score level fusion for the fingerprint, finger-vein and the finger knuckle. They also designed a verification algorithm to improve the multi-biometric system's accuracy effectively.

2.2 3D Contactless Fingerprint Systems

The $3D$ reconstruction can be majorly performed via three famous techniques *viz.* $i)$ Photometric Stereo $ii)$ Structure Lighting and $iii)$ Stereo Vision Camera. The fundamental difference between the first and the last two techniques is that the former uses single static camera and varies the illumination while the latter 2 vary the camera position or technically use one of more camera and projector. To design a low cost system we proceed with the Photometric Stereo as the preferred method.

Apart from Stereo Vision and the Structured Lighting, other methods also exist for $3D$ fingerprint reconstruction like Laser-based systems, OCT(Optical Coherent Tomography) based methods. So we further divide the sub-sections as Photometric Stereo based method and Non- Photometric Stereo based methods.

2.2.1 Photometric Stereo Based Methods

The work by Kumar *et al.* [15] is most closely related to our work. They pioneered the use of photometric stereo for $3D$ fingerprint reconstruction. They addressed the problem of bulky nature of the $3D$ fingerprint setup due to the use of multiple camera and projectors. Hence, they introduced single camera $3D$ reconstruction of the fingerprint. Along with the $3D$ reconstruction, they also introduced $3D$ finger surface features using Finger Surface Codes. The conventional minutiae reconstruction was enhanced into the $3D$ minutiae and matching was performed. Further they mixed $3D$ and $2D$ fingerprint minutiae matching scores to give the highest possible accuracy for $3D$ fingerprints. Their pipeline consisted of clicking image of the fingers in a photometric stereo based setup. The images were clicked under blue light LEDs present in 7 illumination directions. The images were pre-processed to get the enhanced fingerprint impressions. Afterwards, the pre-processed images are used to reconstruct the $3D$ fingerprint using Photometric Stereo. Also, in a parallel pipeline they extracted $2D$ fingerprint features. The $3D$ fingerprint features were also extracted from the reconstructed $3D$ fingers. The two features were fused together on score level to obtain 96% *GAR @ 1% FAR*. However, the matching score obtained by matching the $3D$ minutiae only was 85% *GAR @ 1% FAR*. This has emerged due to the poor quality of reconstruction of fingerprints via Photometric Stereo. The $3D$ reconstruction using Photometric Stereo was poor for their case because they assumed the finger surface to be lambertian; hence, they could obtain the $3D$ fingerprint only for a finger patch without proper finger shape. The ignorance of the Lambertian nature of the fingers lead to the involvement of sub-surface scattering and specular reflection property of the finger in reconstruction hence distorting the reconstruction. Eliminating the non-Lambertian nature of the finger while reconstructing it with Photometric Stereo is one of the major focus of our research work.

Further, Zheng *et al.* [46] obtained even higher matching accuracy for $3D$ fingerprints using minutiae obtained from surface normals and albedo. They avoided the step of $3D$ finger reconstruction and hence they removed the error due to integrability problem faced while converting surface normals into the $3D$ surface. They obtained 7 images of the finger under 7 illumination directions. They created

7 sets of 6 images by leaving one image out of seven every time. Now on each of the 7 sets, they applied photometric stereo and obtained normals and albedo. So, for each set they have 4 images (x,y,z vectors of normals and albedo image). This provided $7 * 4 = 28$ images. From each of the 28 images they obtained the $3D$ minutiae features by convoluting spatial filter with each image [16]. Then the final minutiae was selected and was used for the matching. This approach improved the $3D$ minutia matching accuracy to 96% *GAR @ 1% FAR*. Even if they obtained the high matching score, they assumed the finger surface to be lambertian. Hence the retrieved surface normals suffered from sub-surface scattering and specular reflection. Also the retrieval was obtained for a patch, rather than the whole finger surface.

In the follow-up work by C.Lin *et. al.* [20], they improved the $3D$ matching accuracy by using colored LEDs to detect and correct finger motion. Also, they used the 3 color channels of the finger to reconstruct the $3D$ fingerprint via Photometric Stereo. Their work focused on improving the accuracy and speed of the $3D$ minutiae matching algorithm using a hierarchical tetrahedron-based matching approach. This algorithm was used to speed up the process of aligning and matching $3D$ minutiae algorithms. This approach helped them improved the accuracy and the matching time by 15 times of the state of the art. But they also mixed the $2D$ and $3D$ fingerprint features to obtain their best results. Similar to the other previous methods *All the above methods focused on improving the fingerprint matching accuracy instead of improving the* $3D$ *fingerprint reconstruction quality. All of them assumed the finger surface to be lambertian and used the same method of the reconstruction as mentioned by Kumar* et al. *in [15].* Hence, we consider [15] as the *baseline* for our work and try to improve the reconstruction quality of the $3D$ fingerprint using Photometric Stereo.

Xie *et. al.* [42] proposed an HK model for modeling finger surfaces to decrease the effect of the sub-surface scattering property of the finger. It is the only work based on the photometric stereo that focuses on addressing the non-lambertian nature of the finger and improving the $3D$ reconstruction quality. They tried to model the surface reflectance property for the finger surface so that we can take the non-Lambertian nature of the finger into account and perform Photometric Stereo based reconstruction. According to them the irradiance of the finger surface can be given as follows:

$$L_r = \mu \cdot L_i + \rho \cdot \theta_i \tag{2.1}$$

where $\mu and \rho$ are the parameters based on skin properties. The first term signified the skin property and the second term signified the direction of illumination and together they decided the irradiance values for a pixel. They formulated an ojective function as

$$argmin_x E(x) \tag{2.2}$$

where

$$E(x) = \sum (L_r - I_{pixel})^2 \tag{2.3}$$

where $x = (n_x, n_y, n_z, \rho, d, \sigma_s, \sigma_a)$ is a list of unknowns. The first three parameters are components of surface normals and other are the properties of finger,namely, thickness of epidermis, absorption and scattering coefficients respectively. The x can be obtained by optimizing equation 2.2 using Levenberg-Marquardt optimization. However, the results obtained were not significantly different from the lambertian finger model, as mentioned in figure 3.18 of [14].

2.2.2 Non- Photometric Stereo Based Methods

Stereo Vision based methods Stereo Vision-based methods involve using bulky and complex setups having two or more cameras to obtain fingerprint images from multiple viewpoints. Among the multiple views, they find correspondences to get good quality $3D$ fingerprints [4, 21, 27]. However, the primary issue with stereo vision-based methods is finding the correspondences. The correspondences are found for the blocks of the pixels rather than the pixels themselves. It leads to less accurate fingerprint reconstructions as block correspondences make the $3D$ reconstruction noisy. We will get into the methodology of each of the references mentioned above in detail. Liu *et.al.* [21] used three cameras to reconstruct the $3D$ fingerprints. Their pipeline consisted of calculating camera parameters and performing correspondence establishment. The correspondence were established first using the SIFT features and applying RANSAC to remove outliers. Further the obtained correspondence were used as a baseline for obtaining correspondence via ridge map. The ridge-map was obtained by applying enhancement techniques on the fingerphoto images. After finalizing the correspondences, the 3D coordinates were computed. Alongside the finger shape was estimated using the structured lighting method. Then the two $3D$ structures were interpolated together to obtain the final $3D$ finger. The major issue and challenge faced by them was in establishing the correspondence as the fingerprints are very high frequency structures and the correspondences are obtained for blocks rather than pixels. Parziale *et.al.* [27] designed a device named *Surround Imager*. It consisted of 5 cameras at fixed positions forming a semi-circular arrangement. They captured the finger image and then using the Stereo Vision and Photogrammetry algorithms reconstructed the $3D$ of the finger. Due to the arrangement of camera in a semi-circular fashion they were able to obtain the rolled -equivalent $3D$ structure of the fingerprint. They also virtually rolled the finger on the $2D$ plane to obtain the $2D$ rolled fingerprint. They introduced new representation for the minutiae in $3D$ by extending the minutiae features from (x,y,θ) to (x,y,z,θ, ϕ), where the orienation of the minutiae in $3D$ space is represented using 2 angles and a depth parameter is introduced using the z coordinate. Labati *et.al.* [4] designed a fingerprint capture setup using two camera for a less constrained environment. The setup did not required contact with any surface or a finger placement guide. To compensate the possible differences in finger placement they designed an exclusive algorithm to compute $3D$ model of the shape of the finger. Also, they devised new matching strategy based on computation of multiple touch-compatible images. The system was not able to reconstruct the $3D$ of the ridge-valley however it reconstructed the finger shape which is invariant to the angle and distance of the finger to the camera. They obtained the finger image under green light and obtained the finger ridge-valley texture which they mapped on the finger shape. From this stitched $3D$ structure they obtained the touch compatible images for performing the matching at multiple arbitrary rotations. These images were used to perform $2D$ fingerprint matching and provided an EER of 0.06%.

Structured Light based methods In Structured-Lighting-based systems, multiple pattern-shifted images are projected and captured. The correspondences are established based on the projected pattern, which is more reliable than the stereo vision method. In methods proposed in [17, 37, 9, 43], the advantage is that they can recover ridge-valley details and achieve relatively accurate $3D$ depth information along with a smooth overall shape. However, the hardware system of these methods is expensive and bulky. Also, the increase in projection and capture frequency requires a more sophisticated projector and camera. Wang *et.al.* [37] obtained the $3D$ fingerprints by using Structured-Lighting method. The focus of their research was to know if the depth of the ridge-valley pattern help in improving the matching capability of the fingerprints. They first reconstructed the $3D$ fingerprints and then provided a detailed flattening algorithm based upon unfolding an elastic tube fit to the peaks and the valleys of ridges identified within the scan. After unrolling the fingerprint they compare the unrolled depth map and the albedo with the ink-print rolled equivalent fingerprints and showed an improvement in the matching accuracy. Labati *et.al.* [17] used a single image captured using 2 cameras under a single projected pattern. They used the projected pattern to obtain the correspondence points among the images captured with the two cameras. Further they devised a special algorithm to separate the finger information from the projected pattern and hence reconstructed the $3D$ fingerprint. After the reconstruction they unrolled the fingerprint using the unrolling method similar to [37]. They compared the accuracy of the fingerprints reconstructed without the projected pattern and two views with their results and showed the improvement in the quality of the reconstruction and the speed of reconstruction due to single image capture. Other methods like Optical Coherence Tomography [2, 32] and LASER-based methods [6] to obtain the $3D$ finger point cloud, but both of these methods have bulky and non-portable systems. Hence, all these methods do not suit the portable 3D fingerprint setup. Ganesh *et.al.* [43] used the phase measuring profilometry method of the structured lighting for the reconstruction of the $3D$ objects using structured lighting. Their aim was to reduce the error in the $3D$ reconstruction by using multi-frequency approach in the structured lighting. Along with fingerprints they also reconstructed several other objects. In the phase measuring profilometry as sinusoidal pattern shifted laterally across a surface. The surface is images at uniform intervals and the phase is recovered which helps in setting up the correspondences for the $3D$ reconstruction. They studied the use of multiple frequency of projection and used low-frequency projection to unwrap the high frequency and yield a non-ambiguous depth. Using their multi-frequency approach they obtained accuracy of $0.127mm$ standard deviation in depth with $0.92mm$ pixel spacing. Huang *et.al.* [9] used the fringe projection technique to obtain the $3D$ fingerprint features along with the color and the texture information. They projected a series of color sinusoidal fringe patterns with optimum three-fringe numbers. The fringe patterns get distorted by the finger structure and hence this distortion depicts the $3D$ shape of the fingerprint. The speciality of the optimum three-fringe number selection method is that it computes absolute phase pixel-by-pixel and hence gives a better result than traditional phase unwrapping methods. For the case of fingerprints, traditional method fails due to multiple discontinuity in the surface due to the ridge-valley structure. They successfully obtained the fingers with fingerprint and the color and texture information using the above method.

Other methods Apart from Structured Light and Stereo based methods, there are other auxiliary methods to obtain the $3D$ fingerprints such as OCT (Optical Coherent Tomography) [2, 32], Laser-based reconstruction [6] method which will discuss here briefly. The major drawback of these two methods is bulky setup and huge cost involved in $3D$ fingerprint reconstruction.

Darlow *et.al.* [2] used OCT to image the sub-surface layers of the finger. OCT (Optical Coherent Tomography) is a three dimensional imaging technology used to image light-scattering media. As the finger surface is a translucent surface with sub-surface scattering property, the OCT can be deployed to image the inner surface of the finger. The OCT method provides an excellent method to avoid spoofing and obtain the fingerprints irrespective of skin condition and skin injuries. However, the OCT scans are corrupted by speckle noise and have low contrast. Darlow *et.al.* [2] researched image enhancement techniques for OCT scan images to improve the internal fingerprint quality. They presented a novel fingerprint mapping technique by detecting the papillary junctions which provided improved RMS contrast of 97%. On the similar lines, Sousedik *et.al.* [32] developed a technique to detect the boundaries of the skin layers in an OCT scan. This method helped in easily detecting the genuine finger from a spoof finger.

Galbally *et.al.* [6] devised a LASER-based setup to reconstruct the $3D$ fingerprint. They focused on obtaining the true depth of the finger by directly reconstructing the finger point cloud via the sensor. They eliminated the need of post-processing step of reconstruction as we have in all other $3D$ reconstruction techniques. They designed a separate setup having a camera installed at 45° degree along with a LASER projector at 90° degree. They obtained the depth of each point of the finger surface by mechanically moving the LASER projector across the length of the finger and performing triangulation at each point to obtain the depth. They also developed a complete recognition system as an alternative to traditional minutiae based detection. They used LBP and HOG features of the $3D$ point cloud for this purpose. They also developed a $3D-FLARE$ database consisting of 3000 $3D$ finger point clouds. They obtained their best EER of 0.9%

2.2.3 Methods for Obtaining Finger Shape:

The finger shape can be easily obtained using structured light and stereo vision methods as they are independent of the optical properties of the surface. Yan *et al.* [44] used six cameras to create a visual hull of the finger to obtain a $3D$ finger shape, vein model, and finger texture. They established a database and benchmark $LFMB-3dFB$ for $3D$ Finger Biometrics. Along with the device designing and the dataset they also designed evaluation protocols for boh identification and verification tasks for $2D$ and $3D$ finger trait recognition and score-level fusion. Labati *et al.* [18] provided a simulation method for synthesizing fingertips based on 2 virtual views. They computed the realism of the simulated models by comparing them with real contactless acquisitions. They took a rolled contact-based fingerprint sample and used it to perform silhouette computation. Using the obtained silhouette they computed the $3D$ finger shape. On the computed shape they further computed the $3D$ ridges. After this they simulated finger skin color, illumination for a given source location. Using the simulated finger they further created finger model under 2 views. These two views were utilized to perform the $3D$ reconstruction for the fingerprint. These models can be used for the tasks of minutiae extraction and matching. However, to the best of our knowledge, it is the first time we have obtained the finger shape using a single camera and photometric stereo.

Chapter 3

Split and Knit: A Single Camera 3D Finger Reconstruction Algorithm

3.1 Introduction

The major quest for this research work is to obtain a contactless portable setup to reconstruct the 3D fingerprint for the whole finger along with the overall finger shape. As mentioned in the introduction chapter 3.3.3, Photometric Stereo provides a possibility to reconstruct a 3D finger phalange in a portable setup as it requires a single camera. But it's assumption of the imaged surface to be Lambertian disqualifies the finger surface from being properly reconstructed by the Photometric Stereo. The Photometric Stereo not only requires the surface to be Lambertian but is also sensitive to high-frequency illumination noise. In contrast, the finger surface is non-Lambertian and exhibits sub-surface scattering and specular reflection.

The non-Lambertian nature of the finger poses two *challenges*:

1. It prevents the capture of ridge-valley point cloud for the whole finger area due to sub-surface scattering.

2. It distorts the overall finger shape due to specular reflection.

We proposed *The Split and Knit algorithm*, a pipeline to reconstruct a 3D finger using Photometric Stereo by using image processing and computational illumination techniques. The essence of the Split and Knit algorithm lies in the fact that it visualizes the information contained in the finger surface as a combination of high and low-frequency spatial patterns. It visualizes the finger as low-frequency shape information wrapped around with a high-frequency ridge-valley pattern. The Split and Knit algorithm splits the finger image's low-frequency and high-frequency information and reconstructs them separately. Afterward, it Knits them together to obtain the finger phalange. This derives it's name as **Split and Knit Algorithm**. This chapter details the theory and the pipeline of the Split and Knit Algorithm.

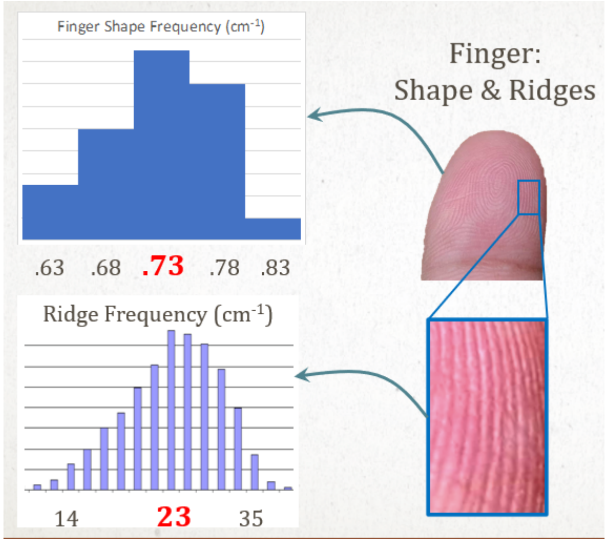


Figure 3.1 Histogram of spatial frequency of ridge-valley pattern and finger shape in humans

3.2 Theory

3.2.1 Properties of Human finger

Spatial frequency of the finger components Spatial frequency of the finger denotes the change in the structure of the finger per unit length of the finger. If we observe the finger in transverse direction, we find that the finger is composed of two highly different spatial frequencies.*1)* The ridge-valley pattern have very high frequency. According to a research [40] on an average there are 23 ridges per cm on a human finger as shown in figure 3.1. *2)* As compared to the ridge-valley the finger shape has much lower spatial frequency of approx. 0.73 cycles per cm. The difference between the two frequencies is shown in the figure 3.1. The finger shape and the ridge-valley patterns were getting corrupted during reconstruction with the Photometric Stereo due to the non-Lambertian nature of the finger skin. To reconstruct the finger using Photometric Stereo we wanted to *separate* these two frequency components and reconstruct them. This might help us deal with the limitations of the Lambertian nature of the finger. But the direct separation of finger shape and ridge-valley pattern using frequency domain of a finger image is not possible directly due to the optical characteristics of the finger skin, discussed in the next sub-section.

Interaction of finger skin with visible light waves When visible light falls on the skin, some portion of the light is reflected back from the top of the skin and another portion gets transmitted within the skin, as shown in the figure3.2. Following are the four terms related to the *visible light-skin interaction-*

1. *Direct Component*: The component of light which reaches the camera by getting directly reflected from the surface of the skin is called the *Direct Component* of light.

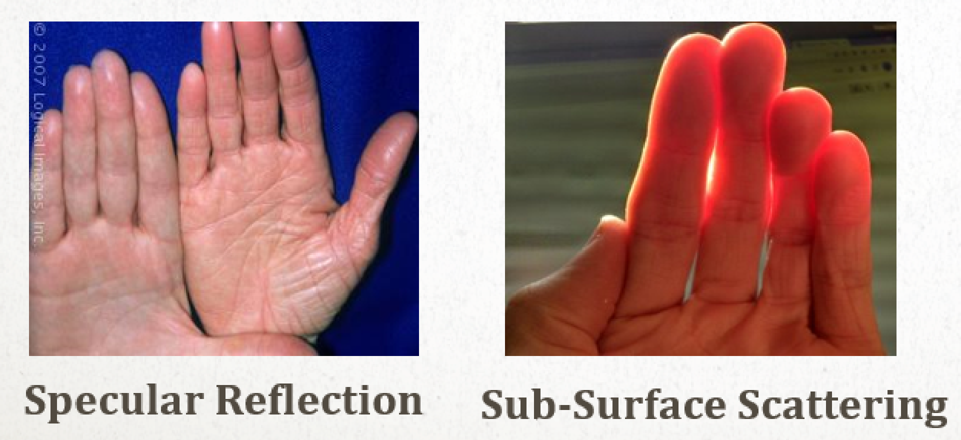


Figure 3.2 Two properties of finger skin Specular Reflection and Sub-surface scattering depicting its Non-Lambertian nature. Due to these properties the finger does not appear same from all directions and hence can't be perfectly reconstructed using Photometric Stereo.

2. *Specular Reflection*: It is the property of an object which makes it appear shiny. When the incident light gets reflected hugely in a single direction (the viewing direction) then that point appears very bright as a light source. This contributes to the non-Lambertian nature of the finger surface. This phenomena is shown in the figure 3.2.

3. *Global Component* The light rays reaching the camera from different directions due to multiple optical interactions of light at different points of surrounding as well as within the objects are called global component. Sub-surface scattering is a type of Global Component.

4. *Sub-Surface Scattering*: It is the optical property of a surface which allows light to transmit through it partially and get scattered within the inner layers of the surface and egressing from a randomly different point of the surface. Such surface when viewed in a camera appears translucent. It is because the camera gets the pixel intensity at each point as the combination of the light directly reflecting from that point and the light ray coming from within the surface due to being scattered within the layers of the surface. When this phenomena occurs in the fingers it gives a reddish hue to the finger. It is because of the light reflecting from the epidermis of the finger surface. This phenomena is shown in the figure 3.2

The true geometry of an object is depicted by the direct component of the light which gets corrupted by the spcular reflection and the sub-surface scattering, hence the direct separation of ridge-valley frequency and the finger shape frequency is not possible using a visible light finger image.

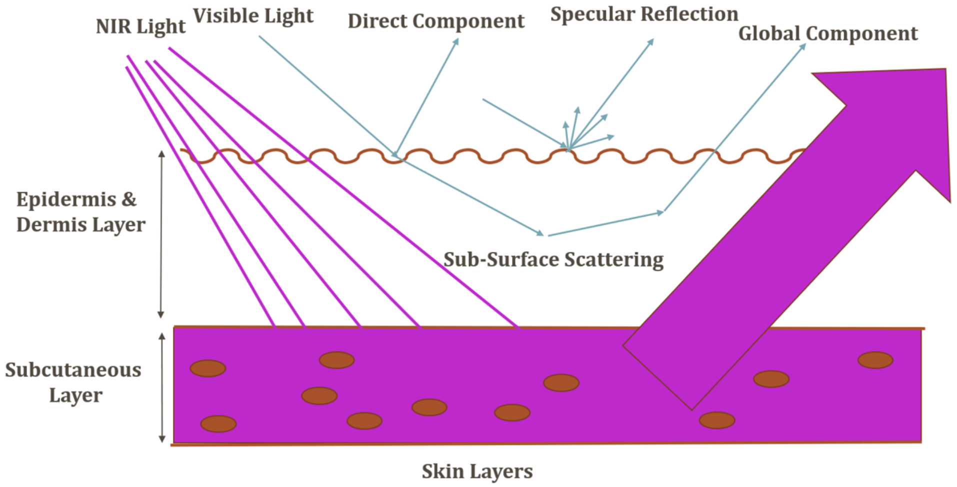


Figure 3.3 Optical properties of finger skin with respect to visible light and IR light. Different frequency of light interact uniquely with various skin layers. This observation helped us in segregating finger shape from rest of the finger via computational illumination.

Interaction of finger skin with NIR light When the NIR light falls over the skin, due to its longer wavelength than the visible light it gets strongly absorbed by the water and fat tissue of the skin. The water and the fat tissue of the skin is present in the subcutaneous layer of the skin which is present below the epidermis and dermis layer of the skin as shown in the figure 3.3. As a result of absorption at the sub-cutaneous layer the sub-surface scattering, specular reflection does not happen on the top skin and the epidermis. Also, due to the uniform absorption of the IR light, the ridge-valley information stored in the epidermis of the skin is not visible. Due to the above three reasons, the finger appears homogenous and lambertian under IR light as shown in the figure 3.4. This appearance provides us the opportunity to separate the finger shape from the ridge-valley point cloud. Hence we use computational illumination of different frequencies to separate the spatial frequency components.

3.2.2 Photometric Stereo for Surface Normal Reconstruction

The prominent technique used for obtaining the 3D reconstruction techniques is using triangulation between the two views. Stereo Vision and Structure lighting both follow the principle of triangulation by observing the objects from different position or under different coded patterns. The fundamental step of the triangulation technique is to obtain the correspondence in the multiple views of the object. The errors in triangulation also arise from this very same step of establishing correspondences. Also, for the case of reconstructing the fine fingerprint pattern this error becomes more critical. The idea of the Photometric Stereo is to vary the direction of incident illumination between successive images, while holding the viewing direction constant for a static scene. Due to this the correspondence between the successive images is inherent. It is called Photometric as it uses the radiance values recorded at a single image location, in successive view, rather than relative position of the displaced feature. The above facts provide Photometric Stereo an inherent advantage over the Non-Photometric Stereo based methods for fingerprint reconstruction [39]. For relating the radiance values recorded in an image to the object shape requires a model of the way surface reflects light. The reflectance map provides a way to relate the scene illumination, surface reflectance and imaging geometry into a single model allowing us to write the image intensity as a function of surface orientation. But the reflectance function is non-invertible as the surface orientation has two degrees of freedom while the image intensity has single degree of freedom. This leads to the multiple observations of the surface under different illumination conditions. As we are performing multiple observations under different illumination conditions to know the geometry of a single point, it is required by the surface to reflect light independent of the illumination direction and according to the surface geometry only. In other words, the surface should be Lambertian reflecting light equally in all directions.

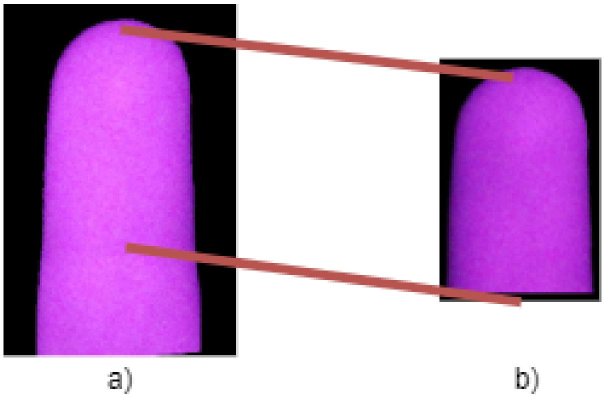


Figure 3.4 a) Appearance of finger in IR light b) Cropped Phalange appearance in IR light

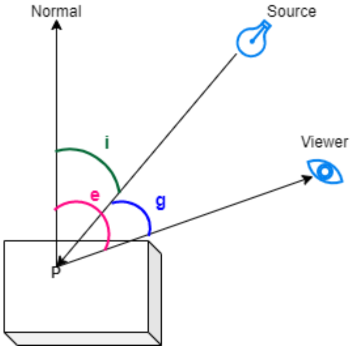


Figure 3.5 The incident angle (i), emergent angle (e) and phase angle (g) between surface normal, viewing position and illumination direction at point P

Reflectance Map The fraction of light reflected by an object surface in a given direction depends on the following: a) Optical Properties of the surface material b) Surface microstructure c) Spatial, spectral and state of polarization of the incident illumination. However, for many surfaces the fraction of the incident illumination refelcted in a given direction depends only on the surface orientation. The reflectance function of such surface can be modelled using three angles i incident angle, e emergent angle and g phase angle as shown in figure 3.5. The function of these three angles $i.e.\phi(i, e, g)$ is related to the BRDF of the surface representing the rating of surface radiance to irradiance measured per unit surface area, per unit solid angle, in the viewing direction. Under the orthographic projection of the scene with the viewing axis aligned to the negative z-axis, an object's surface can be expressed with the values of its z coordinate at a given spatial location x,y i.e.

$$z = f(x, y) \tag{3.1}$$

Here the spatial location x,y of the object correspond to the u,v pixel location in the image due to the orthographic projection. Given the equation of the surface, we can calculate the surface normal vector as

$$[\frac{\partial f(x, y)}{\partial x}, \frac{\partial f(x, y)}{\partial y}, -1] \tag{3.2}$$

If we replace

$$p = \frac{\partial f(x, y)}{\partial x} \tag{3.3}$$

and

$$q = \frac{\partial f(x, y)}{\partial y} \tag{3.4}$$

then the surface normal can be written as $[p, q, -1]$ where p and q are the surface gradient of the surface $f(x, y)$. Hence, using the gradients we can express the surface orientation. In Photometric Stereo, it is assumed that the light source is a single distant point source to make the irradiance constant for the whole object. It is also assumed that the imaging device produces image irradiance proportional to the scene radiance. Also, in an orthographic projection the phase angle g as shown in the figure 3.5 is constant. Thus for a fixed light source and viewer geometry, the ratio of scene radiance to irradiance depends only on the surface orientation given by the surface gradients. If each object surface element receives the same radiance then we can relate scene radiance and hence image intensity with the gradient coordinates p and q. Hence, the reflectance map $R(p, q)$ determines the image intensity as a function of p and q.

If we consider the surface normal vector as $n = [p, q, -1]$, the vector pointing in the light direction as $l = [a, b, -1]$ and the vector in the viewing direction $v = [0, 0, -1]$ then from the figure 3.5

$$\cos(i) = \frac{n \cdot l}{|n||l|} \tag{3.5}$$

$$\cos(i) = \frac{1 + pa + qb}{\sqrt{1 + p^2 + q^2}\sqrt{1 + a^2 + b^2}} \tag{3.6}$$

$$\cos(e) = \frac{n \cdot v}{|n||v|} \tag{3.7}$$

$$\cos(e) = \frac{1}{\sqrt{1 + p^2 + q^2}} \tag{3.8}$$

$$\cos(g) = \frac{l \cdot v}{|l||v|} \tag{3.9}$$

$$\cos(g) = \frac{1}{\sqrt{1 + a^2 + b^2}} \tag{3.10}$$

For an ideal lambertian surface, the surface which appears equally bright from all the directions the reflectance function $\phi(i, e, g)$ can be given as

$$\phi(i, e, g) = R(p, q) = k \cdot \cos(i) \tag{3.11}$$

Hence we can write

$$R(p, q) = \frac{k(1 + pa + qb)}{\sqrt{1 + p^2 + q^2}\sqrt{1 + a^2 + b^2}} \tag{3.12}$$

The effect of changing the light direction corresponds to changing the reflectance function $R(p, q)$. as we are changing the vector l. Let the reflectance maps corresponding to $I_1(x, y)$ and $I_2(x, y)$ be $R_1(p, q)$ and $R_2(p, q)$. The two views hence can be characterized by

$$I_1(x, y) = R_1(p, q) \tag{3.13}$$

$$I_2(x, y) = R_2(p, q) \tag{3.14}$$

If the phase angle g is same in both the views $i.e.$ the direction of illumination l is rotated along the viewing direction v,then the two reflectance maps are rotations of each other. Given that the reflectance map is for a lambertian surface given by 3.11, three views are sufficient to determine the surface orientation and reflectance factor k at each image point. Hence we can take a third illumination direction keeping the phase angle g constant such that

$$I_3(x, y) = R_3(p, q) \tag{3.15}$$

From equation 3.5, 3.11,3.13,3.14 and3.15

$$I_1 = k \cdot \begin{bmatrix} n_1 & n_2 & n_3 \end{bmatrix} \cdot \begin{bmatrix} l_{11} \\ l_{12} \\ l_{13} \end{bmatrix} \tag{3.16}$$

$$I_2 = k \cdot \begin{bmatrix} n_1 & n_2 & n_3 \end{bmatrix} \cdot \begin{bmatrix} l_{21} \\ l_{22} \\ l_{23} \end{bmatrix} \tag{3.17}$$

$$I_3 = k \cdot \begin{bmatrix} n_1 & n_2 & n_3 \end{bmatrix} \cdot \begin{bmatrix} l_{31} \\ l_{32} \\ l_{33} \end{bmatrix} \tag{3.18}$$

$$\begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = \begin{bmatrix} l_1^T \\ l_2^T \\ l_3^T \end{bmatrix} \cdot k \cdot \begin{bmatrix} n_1 & n_2 & n_3 \end{bmatrix} = L \cdot k \cdot N \tag{3.19}$$

The equation above can be summarized as

$$I = L \cdot G \tag{3.20}$$

$$G = k \cdot N \tag{3.21}$$

To ensure that the matrix L is invertible the illumination direction vectors must not be co-planar. Then we can obtain G as

$$G = L^{-1} \cdot I \tag{3.22}$$

$$N = \frac{1}{k} \cdot G \tag{3.23}$$

Here the constant k is known as albedo of the surface. Albedo is the proportion of the incident light or radiation that is reflected by a surface. It is a constant ranging from zero to one. Albedo is zero for the black body which absorbs all the incident radiation while it is one for the surface which reflects away all the incident radiation. Three lighting direction are needed to get an exact solution for the surface normals, but to remove the noise more observations are needed. Thus we took seven illumination directions and converted the equation 3.19 as follows

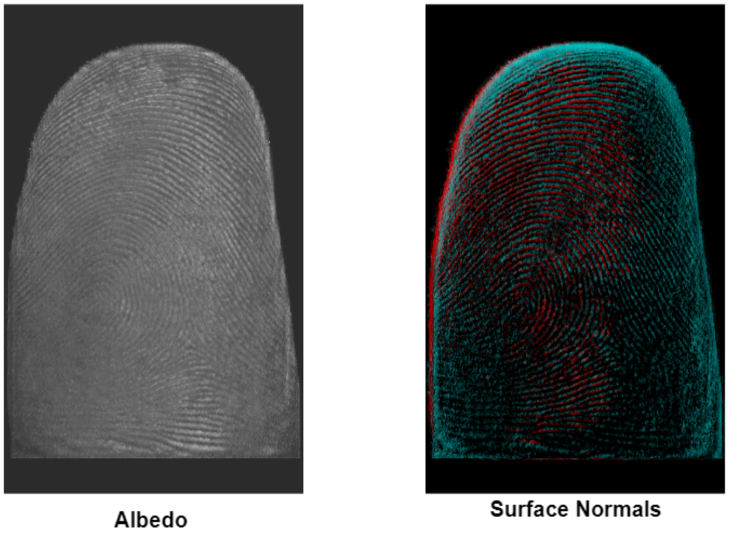
$$\begin{bmatrix} I_1 \\ . \\ . \\ . \\ I_7 \end{bmatrix} = \begin{bmatrix} l_1^T \\ . \\ . \\ . \\ l_7^T \end{bmatrix} \cdot k \cdot \begin{bmatrix} n_1 & n_2 & n_3 \end{bmatrix} = L \cdot k \cdot N \tag{3.24}$$

To obtain the surface normal N, we obtain the least- square solution as below

$$I = L \cdot G \tag{3.25}$$

$$L^T \cdot I = L^T \cdot L \cdot k \cdot N \tag{3.26}$$

$$N = \frac{1}{k} \cdot (L^T L)^{-1} \cdot (L^T I) \tag{3.27}$$



Albedo

Surface Normals

Figure 3.6 The figure shows an example of albedo and surface normals of a finger obtained using Photometric Stereo.

Here $L^T L$ is always invertible due to being symmetric matrix.

For a fixed viewing angle, the pixel intensity at a point P is proportional to the dot product of surface normal, and the illumination direction at the point P, $I = k(N \cdot L)$ where I is the pixel intensity, N is the unit surface normal, and L is the unit vector of the illumination direction at point P. To determine the value of unit surface normal, we need at least three observations of point P in different illumination directions. However, following [15] we took seven images in different illumination directions to minimize noise. Photometric Stereo is for reconstructing static scenes. Hence, we create a support for the finger to restrict the finger's movement during the capture. The finger is kept on the support and held with the other hand. The Photometric Stereo-based setup design details can be found in the next chapter.

3.2.3 Application of Photometric Stereo on the finger surface

The basic assumption of the Photometric Stereo is that the surface has to be lambertian in nature. It needs to appear same when viewed from multiple directions as the pixel intensity produced in a given illumination direction should be only dependent on the geometrical structure of the surface. The appearance of object should not vary because of the illumination direction.

But unfortunately the finger surface is non-Lambertian in nature and do not appear same under different illumination direction. The two major properties because of which the finger surface is non-Lambertian are:

1. Sub-Surface Scattering

2. Specular Reflection

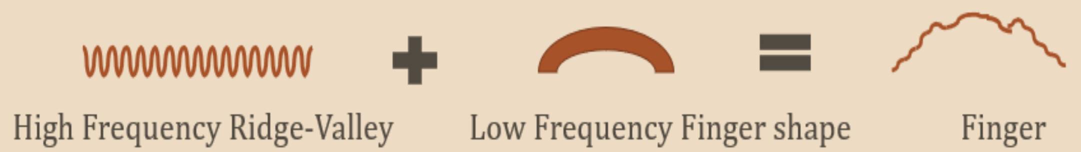


Figure 3.7 The Split and Knit Algorithm visualizes the finger as the sum of low and high spatial frequency components. Finger shape has low spatial frequency while ridge-valley pattern has high spatial frequency.

As mentioned in the section 3.3, the Split and Knit algorithm visualizes the finger as a combination of lower frequency shape and higher frequency ridge-valley structure as shown in the figure 3.7. Split and Knit algorithm aims to reconstruct finger shape and the ridge-valley pattern separately. The Photometric Stereo relates the pixel intensities with the surface orientation of an object via a reflectance map [39]. The surface orientation can be expressed in terms of surface gradients. Hence the pixel intensity can be related to the reflectance function using equation 3.28.

$$I = R(p, q) \tag{3.28}$$

where p and q are surface gradients in the x and y directions, R is the surface reflectance function, and I is the image pixel intensity at a given point P. Under the orthographic projection of the scene with the viewing axis aligned to the z-axis, an object's surface can be expressed with the values of its z coordinate at a given spatial location x,y i.e.

$$z = f(x, y) \tag{3.29}$$

The Split and Knit observes the finger surface (z) as the sum of the finger ridge-valley pattern (z_1) and the finger shape (z_2).

$$z = z_1 + z_2 = f_1(x, y) + f_2(x, y) \tag{3.30}$$

Hence the gradient of the surface becomes-

$$\nabla z = \nabla(f_1(x, y) + f_2(x, y)) \tag{3.31}$$

Therefore, from equations 3.28 and 3.31

$$I = R(\nabla(f_1(x, y)) + \nabla(f_2(x, y)) \tag{3.32}$$

As $f_1(x, y)$ and $f_2(x, y)$ are unknown, they also contain noise due to sub-surface scattering and specular reflection. Hence, we can't calculate them directly as we have single-valued intensity at each point. To calculate them, we segregate the pixel intensity for $f_1(x, y)$ and $f_2(x, y)$ through *computational illumination* by observing fingers in visible and IR light. We use visible light to calculate $f_1(x, y)$, the $3D$ ridge-valley surface.

$$I_{White} = R(\nabla(f_1(x, y)) + \nabla(f_2(x, y)) + \Psi \tag{3.33}$$

Here I_{White} is composed of radiance from the ridge valley, finger shape, and noise due to sub-surface scattering and specular reflection represented by Ψ. We remove the sub-surface scattering effect by extracting the direct component and obtaining the $3D$ reconstruction. Further, we extract the high-frequency element from the reconstructed finger using Shapelet reconstruction to remove the specular reflection and the low-frequency shape component from equation 6.

We observe the finger in NIR light to obtain $f_2(x, y)$.

$$I_{NIR} = R(\nabla(f_2(x, y))) \tag{3.34}$$

The NIR light provides a homogenous appearance to the finger; hence it removes high-frequency ridge-valley from equation 5, and sub-surface scattering and specular reflection do not occur. We can directly reconstruct the finger shape using photometric stereo and Frankot-Chellappa reconstruction.

3.2.4 Point Cloud Reconstruction from Surface Normals

Introduction The generic method to obtain the surface from the surface normals is to integrate the gradient field.The gradient field of a surface should have zero curl or in other words it must be integrable. This means that the integral along any closed loop should be zero and the reconstruction should be independent of the choice of the integration path. But, in practice the gradient fields are non-integrable because of noise in their estimation and the ambiguities due to ill-posed problem. There are three approaches proposed in the literature to solve the problem of non-integrable gradient fields:

1. Impose the integrability constraint as a regularize in the reconstruction equation.

2. Project the non-integrable gradient fields onto the integrable set of basis functions. This converts the problem into solving a least square solution.

3. Further reduce the effect of the outliers in the least square solution by applying PDE (partial differential equation) based image restoration algorithms to get smooth surface reconstruction.

Below sections elaborate on the three methods out of all the above approaches which we used in our research. The Poisson solver worked as a baseline method. Over which Frankot-Chellappa method provided a better reconstruction. Then the shapelets were useful from the frequency based filtration of surface gradients.

i) Poisson Solver Consider a HxW rectangular grid (y,x) of image pixels. Let p,q be the non-integrable gradient field over this grid. Given p, q we want to obtain surface Z. Let's suppose that the gradient field of surface Z is $[Z_x, Z_y]$. The least square solution can be then formulated as:

$$f(Z) = \int\int((Z_x - p)^2 + (Z_y - q)^2)dxdy \tag{3.35}$$

The Poisson equation is a partial differential equation that relates the gradient of the scalar function, f, to its Laplacian:

$$\nabla^2 f = -Divergence(\nabla f) \tag{3.36}$$

This scaler function is required to obtain the correct gradient field. We can assume

$$[Z_x, Z_y] = [p, q] + [\epsilon_x, \epsilon_y] \tag{3.37}$$

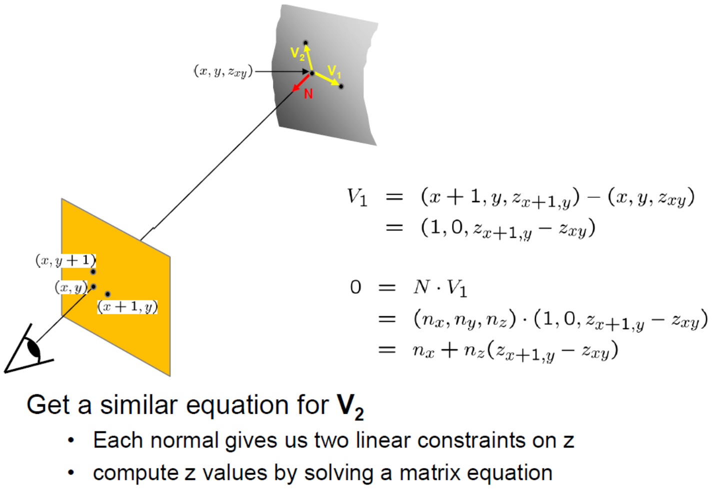
where

$$[\epsilon_x, \epsilon_y] \tag{3.38}$$

is *correction gradient field* which is added to make the non-integrable gradient field, integrable. So from equation 3.35 and 3.36 we can conclude that poisson solver minimizes the norm of the correction gradient field, *i.e.*

$$f(Z) = \int\int(\epsilon_x^2 + \epsilon_y^2)dxdy \tag{3.39}$$

The gradient field can be thought of as the normal vectors at each point on the surface. The surface is reconstructed by finding a scalar function that satisfies the Poisson equation. In practice, the Poisson Solver involves discretizing the surface into a grid of points and then solving a system of linear equations to find the scalar function. The Laplacian of the scalar function is approximated using finite differences and the gradient of the scalar function is computed from the normal vectors provided in the input gradient field. Once the scalar function is found, the surface can be reconstructed by taking isosurfaces of the scalar function at a particular threshold value. Figure 3.8 shows the mathematical equations and steps to obtain surface reconstruction using Poisson solver.



$$\begin{aligned} V_1 &= (x+1, y, z_{x+1,y}) - (x, y, z_{xy}) \\ &= (1, 0, z_{x+1,y} - z_{xy}) \end{aligned}$$

$$\begin{aligned} 0 &= N \cdot V_1 \\ &= (n_x, n_y, n_z) \cdot (1, 0, z_{x+1,y} - z_{xy}) \\ &= n_x + n_z(z_{x+1,y} - z_{xy}) \end{aligned}$$

Get a similar equation for **V₂**

- Each normal gives us two linear constraints on z
- compute z values by solving a matrix equation

Figure 3.8 Implementation of Poisson Solver to obtain surface reconstruction from surface normals

ii) Frankot-Chellappa Reconstruction To find a better least-square fit for the surface gradients to obtain the surface reconstruction, Frankot-Chellappa enforced the integrability constraint by orthogonally projecting the non-integrable estimate of surface gradients onto a vector subspace spanning the set of integrable surface gradients. They used Fourie basis function for this purpose which also provides an intuitive frequency domain interpretation of the surface gradients. This algorithm provided the least-square fit of integrable gradients to non-integrable gradients in one pass of algorithm, which earlier used to be an iterative process.

The least square solution tries to optimize the loss function mentioned 3.35 where it wants to minimize the norm of the correction gradient field. Here Frankot-Chellappa method optimizes the equation 3.35 by explicitly defining an integrability constraint as

$$\frac{\partial z_x}{\partial y} = \frac{\partial z_y}{\partial x} \tag{3.40}$$

where z_x and z_y are the gradients of surface z with respect to x and y directions. We can easily find an integrable projection field if we represent surface slopes by a finite set of integrable basis functions where each function satisfies the equation 3.40. Suppose we represent the surface $z(x, y)$ by the function $\phi(x, y, \omega)$ so that

$$z(x, y) = \sum_{\omega \in \Omega} C(\omega) \cdot \phi(x, y, \omega) \tag{3.41}$$

where $\omega = (\omega_x, \omega_y)$ is a two-dimensional index and Ω is a finite set of indexes. Also the gradient of the surface can be expressed in the same way as:

$$z_x(x, y) = \sum_{\omega \in \Omega} C(\omega) \cdot \phi_x(x, y, \omega) \tag{3.42}$$

$$z_y(x, y) = \sum_{\omega \in \Omega} C(\omega) \cdot \phi_y(x, y, \omega) \tag{3.43}$$

where $\phi_x = \partial\phi/\partial y$ and $\phi_y = \partial\phi/\partial y$.

Among the many feasible basis functions, Fourier basis functions $\phi(x, y, \omega) = \exp(j\omega_x x + j\omega_y y)$ are computationally convenient and gives intuitive idea about the frequency domain of the gradient field. Hence, using the basis function the surface is represented by

$$z_x(x, y) = \sum_{\omega \in \Omega} C(\omega) \cdot \exp(j\omega_x x + j\omega_y y) \tag{3.44}$$

The derivative of the Fourier basis function possess the following properties:

$$\phi_x = j\omega_x\phi \tag{3.45}$$

and

$$\phi_y = j\omega_y\phi \tag{3.46}$$

The above two properties tell us that the coefficients of the gradient are added proportional to the frequency they represent and hence the lower frequency components are diminished and are often get submerged in the noise. *This is one of the major obstacle in obtaining smooth overall finger shape using Frankot-Chellappa algorithm in presence of specular and sub-surface scattering noise.* This can be removed by replacing the low-frequency coefficients using the information from the other sources. We extract this low frequency information by observing the finger in the near-IR illumination. Further to separate out the useful high frequency information we used Shapelet based filtration described next.

iii) Shapelet Reconstruction The Shapelet reconstruction is a surface reconstruction technique from surface normals using basis functions, referred as shapelets. The surface gradients of the shapelets are correlated with the gradients of the surface and the correlations summed to form the reconstruction. This approach also tries to solve the problem of non-integrability of the surface gradients and dependence on the choice of integration path. Many approaches including Frankot-Chellappa reconstruction method utilize non-redundant set of orthonormal set of basis functions for projecting the gradients. Due to which the surface discontinuities and the ambiguity in the tilt of the surface has to be dealt specially. The surface reconstruction suffer with the ambiguity of π in the tilt.

Understanding slant and tilt: Let's assume the x-axis points horizontally from side to side, the y-axis points vertically from top to bottom, and the z-axis points forwards and backwards. Let's assume that we are observing the surface towards negative z-axis then the angle of the surface normal with respect to z axis is slant and the angle of the surface normal with respect to the y axis is its tilt. The tilt ambiguity of π means that if two surfaces with their surface normals at an angle $\theta and(\theta + \pi)$ with respect to the y axis. Both will lead to the same reconstruction without revealing which direction surface faces.

The approach of Shapelet reconstruction uses a redundant set of non-orthogonal basis functions of finite support. The correlation with the basis function is formulated with respect to slant and tilt rather than in terms of gradients of x and y. This is analogous to using circular coordinate system instead of euclidean coordinate system. This helps in allowing the ambiguities in tilt of π and even allow for the cases where no tilt data is available.

The choice of shapelet is inspired by the basis functions used for image decomposition and reconstruction. When the decomposition is done in terms of continuous wavelet transform then the correlation results are like band pass components of the image [13]. The same approach is tried for the surface reconstruction where we want to reconstruct the surface as a combination of band-pass components using Shapelet reconstruction of surface normals.

Our aim is to obtain a range image from the surface normals. Range image can be represented as the sum of set of basis functions. Hence we need to find a basis function which can make a range image from the gradients of the normals. It means that from the gradient data we decide the value of the basis function which adds to form the range image. Correlation between the gradient of a signal and gradient of a basis function *is equivalent to* correlation between signal and basis function because differentiation is linear operation with respect to correlation. So if we correlate the surface gradient with gradient of bank of shapelet basis function, we can reconstruct the surface by summing up the correlation values. Here, summing performs and implicit integration and applies continuity constraint. Hence, *Correlating the gradient of one shapelet filter with the gradient of the signal means extracting a band of frequencies from the signal gradients.*

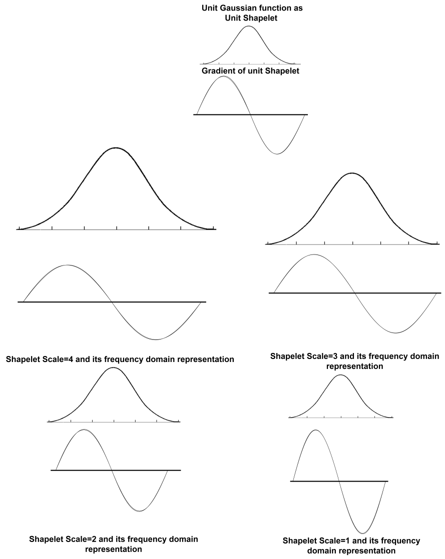


Figure 3.9 The figure shows the unit Gaussian function as the fundamental shapelet function. Here the Gaussian function is symmetric and non-negative giving its gradient and frequency domain representation as odd symmetric. Also as the shapelet scale increases it captures lower frequency range. Also the amplitude of the frequency is inversely proportional to the scale to counteract the scaling of the gradient spectrum.

Understanding the choice of shapelet basis function:

1. Shapelet function must be symmetric
2. Shapelet function must have minimum ambiguity with respect to its gradient
3. Shapelets must allow preservation of phase information of the signal
4. The bank of shapelet must provide uniform coverage of the signal spectrum so that its faithfully reconstructed.

So as per above conditions a unit Gaussian can satisfy all the above conditions as shown in the figure 3.9:

1. It is symmetric and non-negative.
2. Its gradient is odd symmetric, hence displays minimum shape ambiguity.
3. As we increase the scale in the spatial domain, we observe a decrease in the frequency domain. Hence by varying the shapelet scale we can uniformly cover the signal spectrum.

4. to preserve the phase information in the shapelet gradient, the positive frequencies should have non-negative values and vice-versa for the negative frequencies. It is because to preserve phase we need to segregate the positive frequencies from the negative frequencies without ambiguities. This is possible by making the Gaussian function with finite support.

5. As the shapelets are highly redundant and non-orthogonal basis set, one can take each basis function and find other that is approximately orthogonal to it. This helps the smooth transition in the basis coefficients leading to capture of small changes in signal features. *Hence this makes up for the limitations of Frankot-Chellappa for low frequency signal capture.*

3.2.5 Global Direct Component Separation

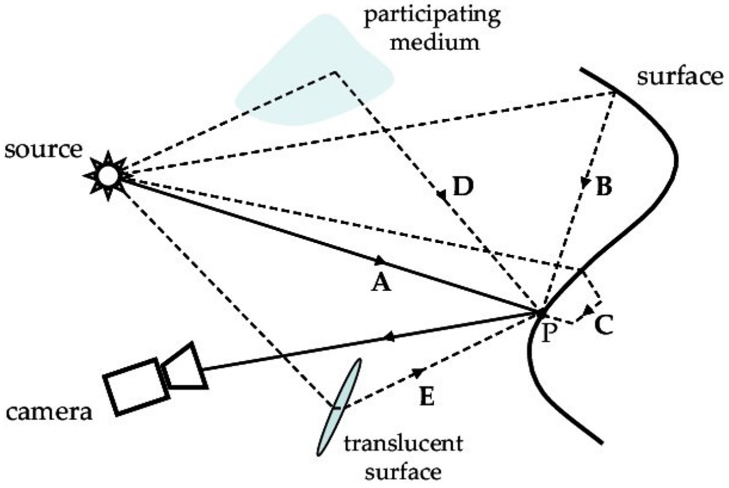


Figure 3.10 Global and Direct Components of illumination for a static scene. Only ray A constitute the Direct component, rest rays B, C, D, E constitute global component of illumination.The figure is taken from [23]

When a source of light illuminates a scene, the radiance of each point in the scene has two components, namely, direct and global. The direct component of the radiance is due to the source's direct illumination of the surface point, depicting surface geometry. The global radiance component is due to the illumination of the surface point by interior surface points and scene points, depicting its surrounding composition. The figure 3.10(originally present in [23]) shows the difference between Global component and the Direct Component. The component of light which is shown to reflect from the object surface to the camera, represented by ray A, is the Direct component of the object while all the other rays, B,C,D, E are the global component. The direct component is due to the surface geometry hence it is useful for understanding the surface geometry. The global component is due to the environmental and internal composition of the object and hence it is important for understanding them. Nayar *et* *al.* [23] presented a methodology to separate the global-direct component of a scene using computational illumination. The finger surface exhibits translucent subsurface scattering, a global illumination component emitting beneath the finger surface. For a good $3D$ reconstruction, we require the image's direct component and the global component's removal. This helps in getting the correct geometry of the ridge-valley pattern reconstructed.

Theory of Separating Global Direct Components: If we assume that a source generates a set of illumination rays and each ray correspond to a single source element, as in the case of a digital projector. Each point of surface, when lit by the source creates a significant scattering towards the camera observing the surface. The author assumed that every camera pixel can observe at most one significant scattering event of the surface. That is two surface points can not send their direct component to a single camera pixel. The radiance of the camera pixel is a combination of the direct component (L_d) and the global component (L_g) created out of the scattering event at a surface point. Hence the total radiance of a camera pixel due to a surface point can be given as $L = L_d + L_g$ If we divide the surface into total N patches, M of which are directly visible to the source. Each of these M patches correspond with the source pixels. Hence the radiance of the camera c due to the patch i of the surface is given by

$$L[c, i] = L_d[c, i] + L_g[c, i] \tag{3.47}$$

The global component of the patch i due to other patches j belonging to previously mentioned set of N patches can be given by:

$$L_g[c, i] = \sum_{j \in P} A[i, j]L[i, j] \tag{3.48}$$

where, $P = [j|1 \leq j \leq N, j \neq i]$. $L[i, j]$ is the radiance of the patch j in the direction of patch i and $A[i, j$ is a relative constant between patch $iandj$ depicting the ratio of radiance reaching from patch j to patch i. Now $L_g[c, i]$ can be further divided into the direct component of other patch j reaching the patch i, $L_{gd}[c, i]$ and the global component of other patch j reaching the patch i, $L_{gg}[c, i]$. Hence,

$$L_g[c, i] = L_{gd}[c, i] + L_{gg}[c, i] \tag{3.49}$$

Now if we assume that only fraction α of the source pixel are activated and these activated pixels are causing a high frequency illumination over the scene surface, then the $L_{gd_a}[c, i] = \alpha \cdot L_{gd}[c, i]$ and $L_{gg_a}[c, i] = \alpha \cdot L_{gg}[c, i]$

Now if we consider two scene images, where in the first image L^+ the scene is lit by a high frequency pattern illumination that has fraction α activated source pixels and the second image L^- where the scene has fraction $1 - \alpha$ activated source pixels. Then if the patch i is directly lit by a source pixel in first image L^+ having fraction α then it is not lit directly in the second image L^- having fraction $1 - \alpha$. Hence,

$$L^+[c, i] = L_d[c, i] + \alpha \cdot L_g[c, i] \tag{3.50}$$

and

$$L^-[c, i] = (1 - \alpha) \cdot L_g[c, i] \tag{3.51}$$

If we assume the brightness of a deactivated source pixel as fraction b, where $0 \leq b \leq 1$ then the above equation can be modified as

$$L^{+}[c,i] = L_d[c,i] + \alpha \cdot L_g[c,i] + b \cdot (1-\alpha)L_g[c,i] \tag{3.52}$$

and

$$L^{-}[c,i] = b \cdot L_d[c,] + (1-\alpha) \cdot L_g[c,i] + \alpha \cdot b \cdot L_g[c,i] \tag{3.53}$$

Now if we wish to maximize the sampling frequency of illumination in both $L^{+} and L^{-}$, we can choose $\alpha = 0.5$. Hence we can modify the above equation as the

$$L^{+}[c,i] = L_d[c,i] + 0.5 * (1+b)L_g[c,i] \tag{3.54}$$

and

$$L^{-}[c,i] = b \cdot L_d[c,] + 0.5 * (1+b)L_g[c,i] \tag{3.55}$$

Here we can observe that L^{+} is always greater than L^{-}. Hence if we record multiple brightness values for pixel i, then the maximum value of each pixel $L_{max} = L^{+}$ and $L_{min} = L^{-}$ and hence we can obtain at $b = 0$

$$L_d = L_{max} - L_{min} \tag{3.56}$$

and

$$L_g = 2 * L_{min} \tag{3.57}$$

Methods for Separating Global Direct Components: There have been 4 methods described by Nayar *et al.* [23] for global direct separation which are as follows:

1. **Checkerboard illumination shift** A high frequency illumination can be created by a checkerboard placed in front of the light source and its complimentary pattern.However, the ideal pattern is difficult to obtain using an off-the-shelf digital projector due to various factors such as light leakages, custom image processing, and limited depth of field. To overcome these issues, a larger number of images are captured than what the theory requires, by shifting the pattern multiple times to capture a total of 25 images. At each pixel, the maximum and minimum measured brightness over all the 25 images, were used to compute the direct and global estimates (L_d, L_g) using equation 3.54 and 3.55 The method is successful for scenes dominated by diffuse inter-reflections, specular components, subsurface scattering, and strong volumetric scattering effects.

2. **Source Occluders** This is a second method to perform global direct separation. It is more suitable for the case of uncontrolled source of light like sunlight. The occluder, which could be a simple line or a more complex mesh, is used to cast shadows on the scene, and a video camera is used to capture the scene while the occluder is swept across it. By analyzing the resulting video, the maximum and minimum brightness values at each point in the scene can be determined, and the direct and global illumination components can be separated. In the case of a line occluder, the

captured video must be long enough to ensure that all points in the scene have been subjected to the shadow. However, a more complex mesh occluder can be used to capture the scene more efficiently. By using a 2D grid of circular holes, only a small circular motion of the occluder is needed to ensure that all points in the scene are captured in and out of shadow. However, it may be limited in its applicability to certain types of scenes and lighting conditions. Additionally, the method may require significant post-processing to separate the direct and global illumination components and to obtain an accurate representation of the scene's lighting.

3. **Other High Frequency Patterns:** Beside shadows other high frequency patterns can also be applied for the global-direct component separation. Structured-light range finders that use coded illumination patterns can easily incorporate this method, especially binary coded patterns that have high frequency stripes. By using a sinusoidal function that varies over space and/or time as the illumination pattern, we can separate the direct and global components of the scene using just three patterns. To generate the illumination patterns, we can use a projector that can generate any positive illumination function. A convenient class of functions is based on the sinusoidal function, where the brightness of a given projector pixel is $a = 0 : 5 + 0 : 5 \sin \phi$ and the frequency ϕ ranges from 0 to $2 * \pi$. Three patterns are generated by changing the phases of the sinusoidal of all the pixels by $2\pi/3$ and $4\pi/3$. By capturing three images of the scene using these three patterns, we can find the direct component L_d, the global component L_g, and the correspondence between camera and projector pixels. With this information, we can compute the 3D structure of the scene. Another way to generate the illumination pattern is to use $\sin(x + siny)$, which has high frequencies along both spatial dimensions. By capturing three images of the scene using this pattern and two shifted versions of it, we can also find L_d, L_g and ϕ. Overall, the separation method can be applied to a variety of high frequency illuminations, allowing us to separate the direct and global components of the scene and compute its 3D structure.But this method is complex to generate and requires specialized projectors.

4. **Separation using a Single Image:** The direct and global components can be also produced using a single image but at a lower resolution.To do this, we filter each color channel of the captured image to find local peaks and valleys, and then interpolate the brightness values at these peaks and valleys to obtain full-resolution L_{max} and L_{min} images. If the separation results are to be computed at a lower resolution than the captured image, we can simply average the values of the L_{max} and L_{min} images within kxk blocks in the high-resolution images to obtain L_{max} and L_{min} images at the lower resolution. Then, we can use equation 3.54 and 3.55 to compute L_d and L_g.

Out of all the above stated methods the checkerboard illumination shift suited most for the direct global component separation of the finger image due to its simplicity, suitability for sub-surface scattering and ability to produce high resolition results for a controlled illumination condition.

3.3 Pipeline

The Split and Knit Algorithm focuses on separate reconstruction of ridge-valley pattern and the finger shape and then combining them together to obtain the finger phalange. Hence, there are three components of the Split and Knit Algorithm:

1. *Reconstruction of the ridge-valley point cloud:* Under this component, the ridge-valley pattern is captured for the whole finger surface rather than a patch by alleviating the non-Lambertian nature of the finger image by Global Direct Component Separation.

2. *Reconstruction of finger shape point cloud:* Under this component, the overall shape of the finger, which gets distorted due to specular reflection and image processing, is retrieved using IR illumination.

3. *Mixing the ridge-valley and finger shape to obtain* $3D$ *finger phalange:* Under this component, we re-mix the higher spatial component and the lower spatial component of the finger to obtain a fine finger phalange.

Figure 3.11 details the Split and Knit algorithm. The below sections elaborate upon each component of the algorithm.

3.3.1 Reconstruction of 3D Ridge-Valley Point Cloud

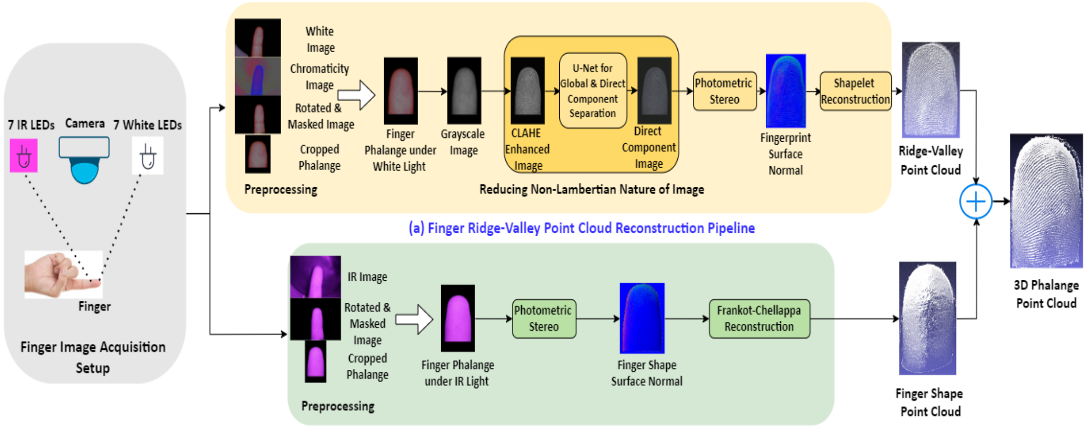
Overview The Split and Knit algorithm obtains the finger ridge-valley point cloud in four steps:

1. Pre-Processing of the finger image. The pre-processing step aims to make the captured finger image suitable for applying the Photometric Stereo. This involves segmenting out the finger from the background by generating a mask, correcting the orientation and segmenting the first phalange.

2. Reducing the non-Lambertian nature of the finger image. This step involves applying the CLAHE enhancement and trained U-Net based direct component separation to make the finger image qualify the criteria of being Lambertian.

3. Obtaining fingerprint surface normals using Photometric Stereo. In this step we actually apply the Photometric Stereo on the pre-processed Direct component of the image. The output of this step is the surface normals for the finger images captured under multiple light direction.

4. Ridge-valley point cloud reconstruction from the finger surface normals. This step has to choose the correct method to obtain the 3D surface from the finger surface normal. This method has to be robust against the noise due to the specular reflection which is still present in the direct component of the finger image. Here we use Shapelet reconstruction method to obtain the 3D surface from the finger surface normals.



(a) Finger Ridge-Valley Point Cloud Reconstruction Pipeline

(b) Finger Shape Point Cloud Reconstruction Pipeline

(c) 3D Finger Phalange Reconstruction

Figure 3.11 The Split-and-Knit Algorithm(SnK) consists of three primary components: (a) Reconstruction pipeline for finger ridge-valley point cloud. The white light images are preprocessed to obtain cropped phalange. Using CLAHE [28] and global-direct component separation by U-Net, the non-lambertian nature of the finger image is reduced. Further, using photometric stereo and shapelet reconstruction [13], finger ridge-valley point cloud is obtained (b) Reconstruction pipeline for finger shape point cloud. The NIR light images are preprocessed to obtain cropped phalange. Further, we apply photometric stereo and Frankot-Chellappa reconstruction [5] to obtain the finger shape point cloud. (c) The phalange point cloud is obtained by pixel-wise addition of the ridge-valley and finger shape point cloud.

The sub-sections below elaborate upon the steps in obtaining the finger ridge-valley point cloud, shown in Figure 3.11(a). All the pre-processing steps and the steps to reduce non-Lambertian nature of the finger to obtain the ridge-valley point cloud are summarised in the figure 3.16

3.3.1.1 Pre-Processing

The preprocessing steps aim to obtain the first phalange segmented at a zero-degree yaw angle for all the finger images. It ensures uniformity in the $3D$ finger phalange output irrespective of the orientation, size, and skin complexion. This step makes the finger image ready for the application of forth-coming image processing algorithm. The pre-processing step has three components: *i) Mask Generation*, *ii) Orientation Correction* and *iii) Phalange Segmentation*

1. Mask Generation The clean mask generation for the finger plays a crucial role in the finger reconstruction and its matching afterwards. This is because Photometric Stereo operates on each pixel for the 3D reconstruction. Due to variability of appearance of the finger with different background and under different light directions, extraction of the finger from the background becomes complex. Use of Neural network based segmentation methods may manipulate the ridge-valley details as they are very minute and of high frequency. Also the graph cut based method need human intervention to provide prior for foreground and background. Hence, we designed a chromaticity based algorithm for the mask generation.

Effect of background color We also observed that finger appeared different with different background under the same setting of the camera and we observed the most detailed fingerprint available in the black background. Hence, we fixed the background as black paper and designed the chromaticity-based algorithm for the same. This was due to the high contrast between the skin color and the black color. Also the black paper is required to be in matte finish to avoid any specular spot in the background.

Chromaticity Image based mask generation According to the dichromatic reflection model, the reflection of in-homogeneous material consists of diffuse reflection and specular reflection [31, 35]. The spectral composition of the diffuse reflection describes the spectral characteristic of the material, while the spectral power distribution of the specular reflection is very similar to that of the illuminant. The appearance of the finger varied with illumination condition. Hence obtaining the diffuse component of the finger image will be close to the original appearance of the finger. This original appearance of the finger is obtained as a chromaticity image of the finger, as shown in the figure 3.13.The chromaticity of an object is an objective specification of the quality of a color regardless of its luminance. We obtained chromaticity image as follows:

$$I_{min} = min_i(I(x, y, i)) \tag{3.58}$$

where I is the mean of the seven RGB images of the finger, each under different illumination and i represent the color channels.

$$I_{sub} = I - I_{min} \tag{3.59}$$

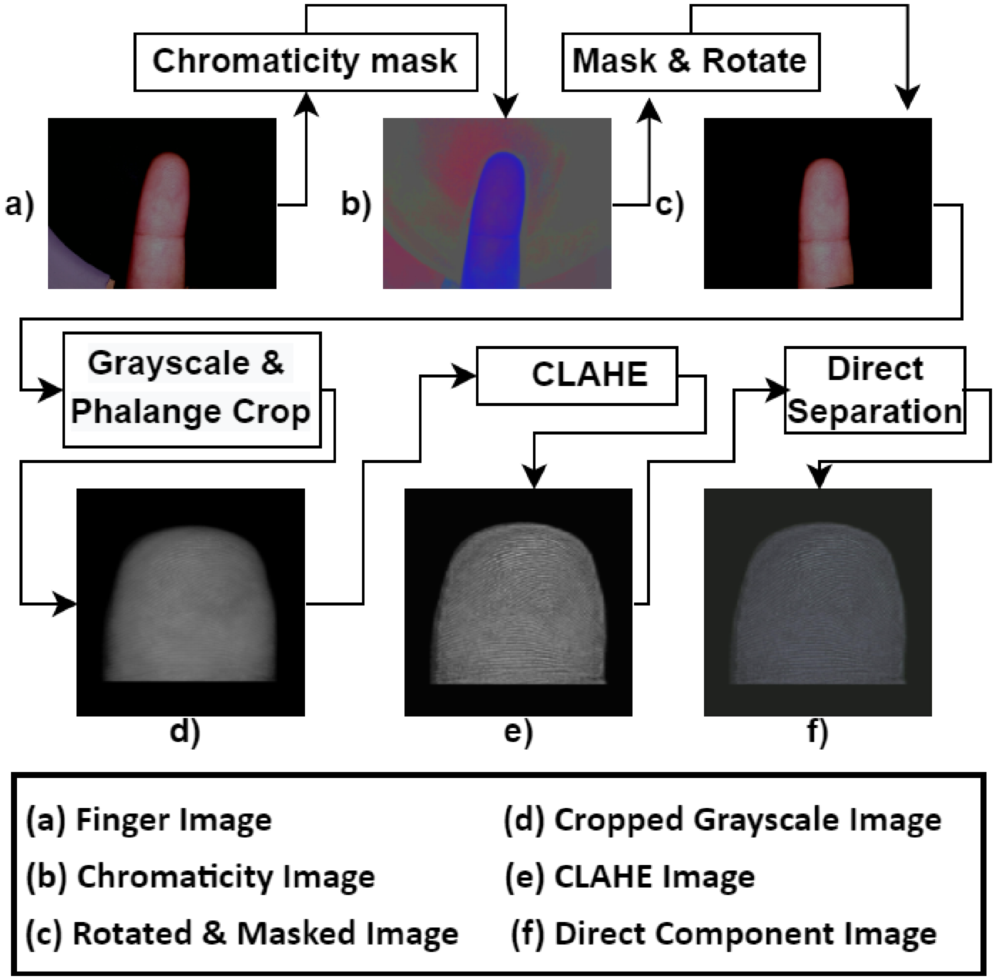


Figure 3.12 (a)-(d) Preprocessing steps to obtain the grayscale image of the first phalange segmented at a zero-degree yaw angle, (e)-(f) Steps to reduce the non-lambertian nature of the finger image.

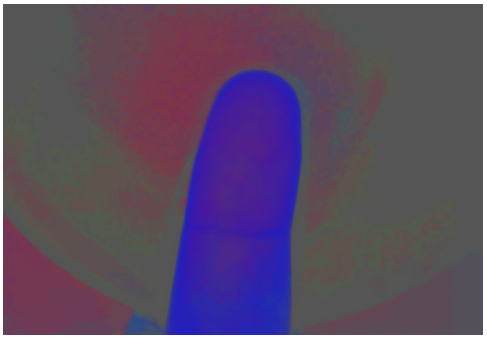


Figure 3.13 The Chromaticity image of a finger captured using fingrprint setup.

$$I_{min-avg} = \frac{1}{x * y} \sum_{x} \sum_{y} I_{min}(x, y) \tag{3.60}$$

$$I_{MSF} = I - I_{min} \tag{3.61}$$

where I_{MSF} is modified specular free image.

$$I_{MSF_sum} = \sum_{i} I_{MSF}(x, y, i) \tag{3.62}$$

$$I_{Chromaticity}(x, y, k) = I_{MSF}(x, y, k)/I_{MSF_sum} \tag{3.63}$$

Here k represents R,G and B channel. Further we applied Gaussian blurring on the chromaticity image followed by *k-means clustering* with k being two. The results of the *k-means clustering* separated the finger from the background very sharply as shown in the figure 3.14. Still if due to some properties of the finger or illumination condition we obtained some spurious components in the mask image then we first programatically determined the spurious region. We did this by detecting the uneven height in the image much lower than the finger height and further we detached such regions from the finger image. After detachment, we cleared them by finding out the largest connected component of the mask image, which is the finger itself. Hence, we obtain a very clear and crisp mask for the finger by the above algorithm.

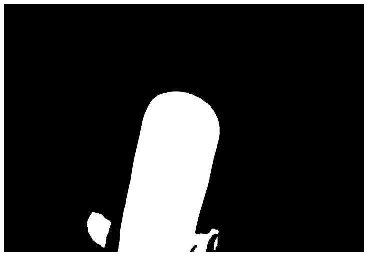


Figure 3.14 The noisy raw mask image obtained from the setup using Chromaticity image.

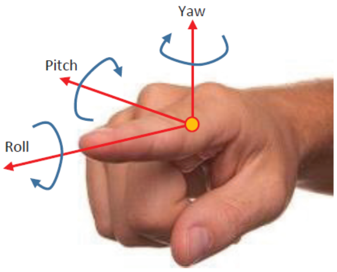


Figure 3.15 The angles to identify the position of a finger. The figure is taken from [6]

2. Orientation Correction The orientation of the finger can be determined using three types of angles *viz. roll, pitch, yaw angles.* These three angles are shown in figure 3.15. The roll and pitch angle can be controlled by asking the person to keep the finger rested on a support, facing towards the orthographic camera. But the major challenge remains in obtaining a 0° yaw angle. We designed as simple but effective algorithm to correct the yaw orientation of the fingers to zero 0°. The 0° orientation of the finger helps in future reconstruction and matching processes. Also, it provides uniformity in the collected data.

To correct the yaw orientation of the finger to 0°, the first step needed is to detect the existing orientation of the finger and then rotate the finger anticlockwise by the same amount. To detect the existing angle of the finger, we utilized the mask obtained in the previous step. Here we obtained the zero crossing points for multiple rows of the mask. Now for each of the obtained pairs of zero-crossing we obtained mid-points. Utilizing these mid points, we fit a first degree polynomial *i.e.* a line. Now for this line we calculated its orientation with respect to the vertical axis. This orientation gives us a good estimate of the current yaw orientation of the finger. We rotate the mask image in the opposite direction by the same amount of angle, hence we obtain the finger at 0° yaw angle.

3. Phalange Segmentation

The phalange segmentation is again required to provide the uniformity in the data collected. Also, at the time of data collection participants keep variable length of the finger exposed to the camera, making it difficult to match the fingers and do the identification at later stage. Phalange segmentation aims to detect the first vein of the finger from the top and mask out the finger below it. Hence, we designed an algorithm to detect the first vein of the finger. The algorithm needs to be independent of the complexion of the finger skin and also it should detect the first vein even if the second vein is visible. Following are the steps of the phalange segmentation algorithm:

Step 1 First of all we obtained the average gray image of the fingers.

Step 2 By performing repeated histogram equalization followed by Gaussian blurring we obtain an approx, threshold-ed image of the finger with darker region around the finger vein as compared to the other finger region. We binarize this image to obtain a mask for the threshold-ed image.

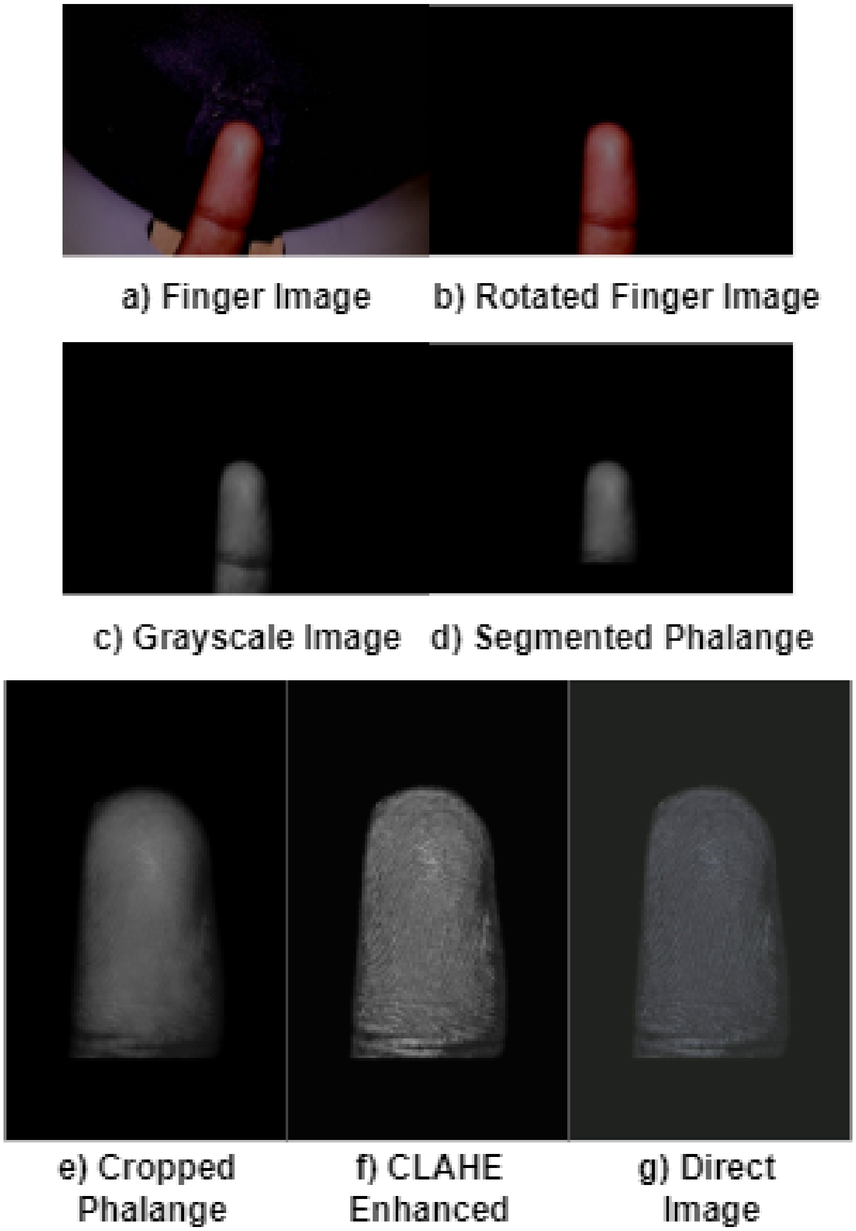


Figure 3.16 The image shows the various stages of fingerprint preprocessing and for reducing Lambertian nature of the finger.

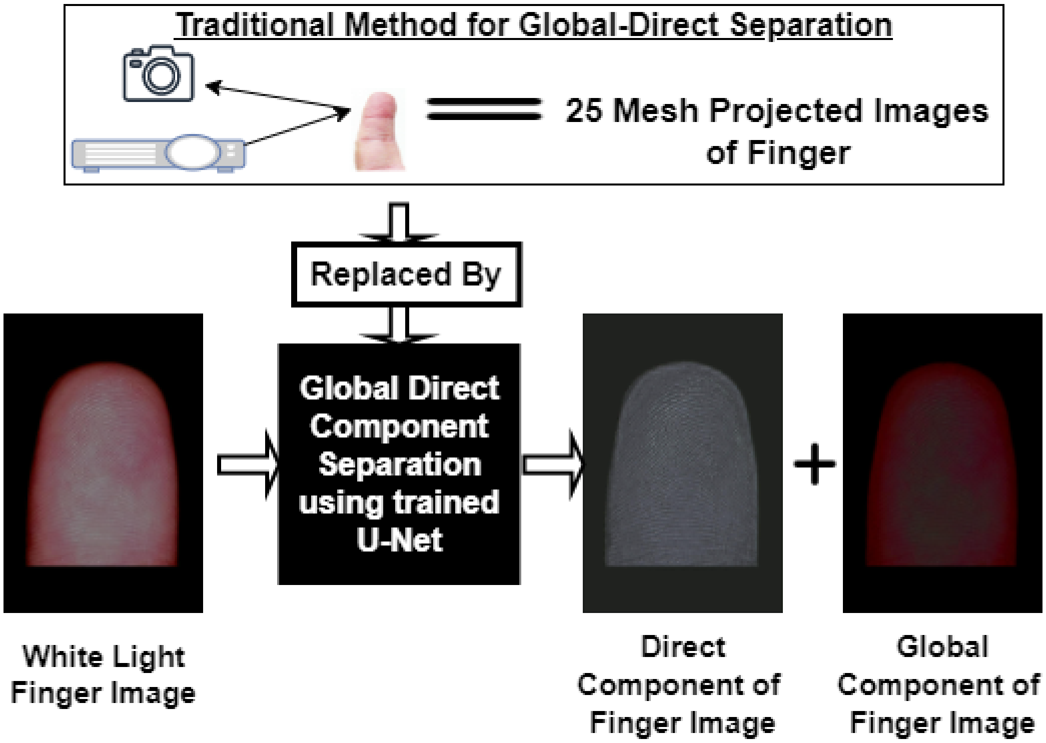

Traditional Method for Global-Direct Separation
25 Mesh Projected Images of Finger
Replaced By
Global Direct Component Separation using trained U-Net
White Light Finger Image
Direct Component of Finger Image
+
Global Component of Finger Image

Figure 3.17 To reduce the non-Lambertian nature of the finger image, we need the Direct Component of the finger image. Traditionally, the Global-Direct Separation is done using 25 mesh images captured with the help of a camera and projector [23]. We replaced the hardware setup with a trained U-Net to provide the Direct Component of the finger image. The Global Component is obtained by subtracting the Direct Component from the white light finger image.

Step 3 Using the obtained mask image we obtain a bounding box around the finger structure and tried to detect the first point where the number of non-white pixels is minimum in a row while traversing from the top region of the bounding box. This provides us an approximate range of the values for the existence of the finger vein.

Step 4 We also designed a verification and correction system where if two such possibilities exist, as in the case of two finger veins, the code automatically judges the better choice considering the height of the finger and the location of the two veins.

Step 5 After ascertaining the position of the first vein of the finger we mask out the rest of the finger and obtain the first finger phalange.

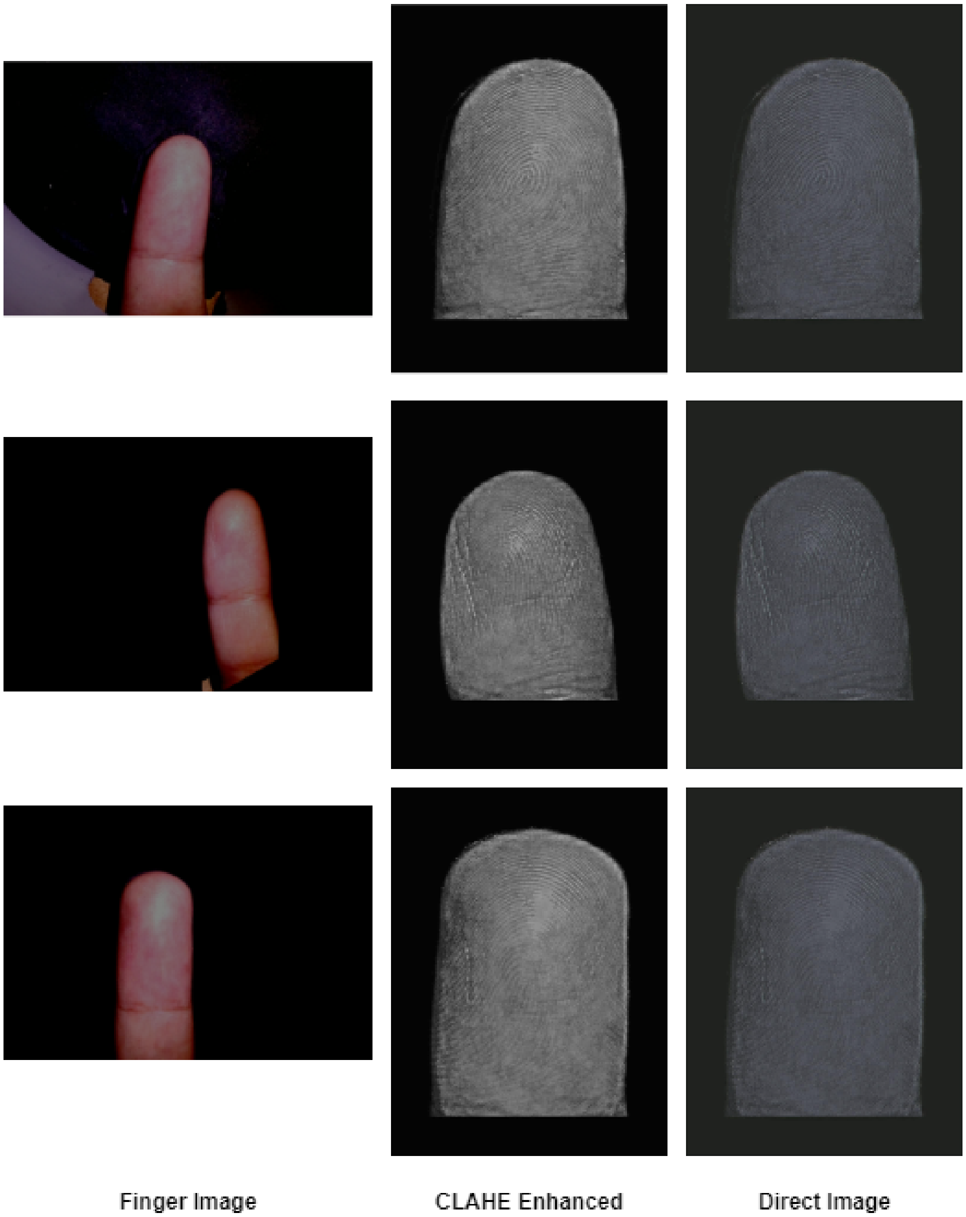


Figure 3.18 The image depicts the steps to reduce the Non-Lambertian nature of the finger. They include the CLAHE enhancement of the finger phalange and obtaining the direct image.

3.3.1.2 Reducing the Non-Lambertian Nature of the finger image

We performed CLAHE Enhancement followed by Global-Direct component separation to reduce the non-Lambertian nature of the finger image.

3.3.1.3 CLAHE Enhancement:

Contactless fingerprint images are relatively low contrast between ridges and valleys compared with contact fingerprint images. The enhancement step has to be chosen so that it does not entirely distort the lighting direction information stored as pixel intensities. In this regard, we chose adaptive histogram equalization to enhance the contrast based on neighborhood pixel information. However, adaptive histogram equalization amplifies the noise in the homogenous region. To fix this, we applied Contrast Limited Adaptive Histogram Equalization (CLAHE), which clips the slope of the transformation function, thus preventing the noise amplification [28].

CLAHE (Contrast Limited Adaptive Histogram Equalization) is a technique used for image enhancement, particularly for improving the contrast of images. It is an extension of the traditional Histogram Equalization (HE) method, which is a global image enhancement technique that works by redistributing the pixel intensities of an image to increase its contrast. It divides the image into small tiles and applies Histogram Equalization (HE) to each tile individually. The size of the tiles can be specified by the user and typically ranges from $32x32$ to $256x256$ pixels. The HE method redistributes the pixel intensities in each tile to increase the contrast of the image, making it easier to see the details and structures. However, one of the problems with traditional HE is that it can over-enhance the contrast of the image and create artifacts or noise amplification. To avoid this, CLAHE introduces a clip limit, which is a threshold value that controls the amount of contrast enhancement in each tile. Pixels with intensities outside this clip limit are redistributed to the neighboring tiles, which helps to preserve the overall image quality and prevent artifacts from appearing. The clip limit can be specified by the user and typically ranges from 2 to 4 times the mean intensity of the image. Higher clip limits can result in greater contrast enhancement, but also increase the risk of artifacts and noise amplification. After the contrast enhancement has been applied to each tile, the image is reassembled by stitching the tiles back together to form the final enhanced image. The result is an image with improved contrast and better visibility of the details and structures, without the risk of over-enhancement or artifact creation. Figure 3.18 shows the enhancement obtained in the fingerprint after applying CLAHE over the 64x64 neighborhood windows of the grayscale fingerprint images.

3.3.1.4 Global-Direct component separation using trained U-Net:

After obtaining the enhanced image of the fingerprint, we proceeded to obtain the direct component of the image. This is done to remove all the global component of radiation and specially get rid of the sub-surface scattering component coming from beneath the finger surface. As mentioned in the section 3.2.5, we explored multiple methods for global-direct component separation and then following the checkerboard illumination shift method mentioned in [23], we separated the global-direct components for 50 palm images. We took 25 images with the shifted checkerboard pattern of each palm surface. The direct image is obtained as follows:

$$L_{direct} = L_{max} - L_{min} \tag{3.64}$$

where L_{direct} is the direct image of the palm surface, L_{max} and L_{min} are the maximum and the minimum image out of 25 checkerboard images of a single palm. L_{max} is obtained by taking the maximum brightness of each pixel across its 25 checkerboard images. Similarly, we obtain L_{min} for a given palm image. This process of obtaining the global direct component separation is not feasible during the capture while the reconstruction and matching of the $3D$ fingerprints. The requirement of the projector will make the system bulky and capturing 25 images under shifted checkerboard patterns for direct component separation will increase the setup time duration. Hence, we trained a neural network to separate the direct component of an image using the above data. We chose palm because the skin on the palm and fingers are the same, so we get to separate the direct component of the same skin. Also, capturing a palm image will provide more pixels for direct global separation and hence more data for training the U-Net. Jacobson *et al.* [10] have trained an Encoder-Decoder-based neural network to separate the direct component. There are multiple other [34, 24] global- direct component separation network, but many of them are not present in the public domain. Also, these networks are trained over the generic scene data and none has been trained on finger/palm data. As our image domain was limited to palm images,we model the direct component segregation problem as a machine learning problem for a specific pattern and surface. We used the direct image dataset created using the checkerboard method. The global image can be further obtained by subtracting the direct image from the RGB finger image as we assume that any component of light other than the direct component in an RGB image constitutes global component.

U-Net for Global Direct Separation: The network trained for the Global Direct component separation generally follow the encoder-decoder architecture [34, 24, 10]. Also we observed that obtaining a direct image from the RGB image is a style transfer problem in deep learning context. Moreover, we wanted a neural network which takes care of global as well as the local details of the image, because fingerprint data is highly detailed and prone to distortion. Based on all above observations and requirements we chose U-Net [29], specifically for separating the direct component of the palm surface. U-Net possess the encoder-decoder architecture, takes care of the global and local details of the image via skip connections and can be used for the task of style transfer. The figure 3.17 the concept of applying U-Net for the Global-Direct component separation.

U-Net model training details: We chose U-Net with a depth of 3 encoder-decoder blocks with 45 layers as shown in the figure 3.19. We replaced the segmentation layer of the U-Net with the regression layer of mean squared error loss to generate the direct component of the RGB finger images. U-Net is also a fully convolutional network that means there is no restriction on the size of the input

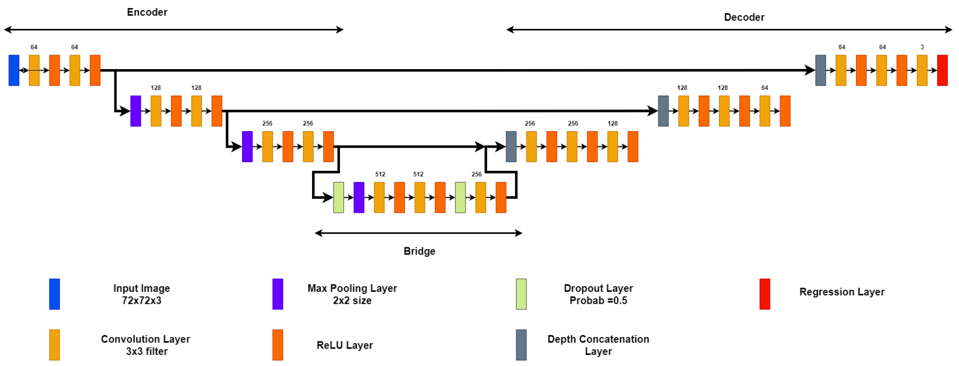


Figure 3.19 Trained U-Net model architecture for Direct Component generation. The number over the top of layers depicts the number of filters. We used image patch of size $72x72x3$ to train the network and used image of size $1528x936x3$ at inference time.

and output image size due to the absence of fully connected layers in the network. As the size of the palm images was $1018x768$ pixels, hence training directly on such a big size image would have been computationally expensive and required more data. Also, the palm images consisted of black masked region which would have wasted the pixels for the task of direct component separation. Hence, we extracted 750 patches of size $72 \times 72 \times 3$ from every image, leading to a dataset of $37,500$ direct image patches. These patches were randomly generated from the palm image and the fully black patches were discarded. These patches were then augmented with random rotation and translation to create variations in the input dataset. Following are the hyperparameters used for the training, we used adam optimizer, learning rate of 0.001, batch size of 32 and trained it for 50 epochs. This gave us the trained U-Net model for separating the direct component of an image.

U-Net model inference details: In the $3D$ reconstruction pipeline, we deployed the trained U-Net model as code block. Here we padded the image to be of size $1528x936$ pixels which is a multiple of $2^{EncoderDepth}$. Further in the model we gave the CLAHE-enhanced image as the input and obtained the direct images as shown in the figure 3.18. The output was then used to generate the input data for the $3D$ reconstruction using Photometric Stereo, discussed in the next section.

3.3.1.5 Photometric Stereo for obtaining surface normals of fingerprint

For applying the Photometric Stereo to the direct image of the finger phalange, we need a $7x3$ matrix containing pre-computed vectors of 7 illumination direction for each LED light. Along with this, we need to provide the mask Image for the finger area for which the Photometric Stereo has to be applied for the $3D$ reconstruction. We also need to provide the 7 images of finger phalange in each illumination direction. Before calculating the surface normals at every pixel, we refine the mask image and make it to be zero at the location where the pixel brightness is not greater than a pre-chosen threshold for all the image channels. This helps in removing the noise in $3D$ reconstruction due to uneven brightness. After this the surface normal can be calculated along with the albedo for any channel of the image. We chose grayscale version of the 7 images for the surface normal calculation. Then by solving the equation 3.65 at each pixel location available in the refined mask we obtain, surface normal and the albedo map as shown in the Figure 3.6.

$$N = \frac{1}{k} \cdot (L^T L)^{-1} \cdot (L^T I) \tag{3.65}$$

3.3.1.6 Obtaining Point Cloud of finger ridge-valley using Shapelet Reconstruction

The Shapelet Reconstruction method converts the surface normals into the $3D$ point cloud. In the first component, the algorithm reduces the non-Lambertian nature of the finger image by applying CLAHE enhancement and separating the direct component of the finger image. It helps in obtaining the fingerprint of the entire finger surface. However, the global shape of the finger gets distorted due to specular reflection, and the image processing applied to reduce the non-Lambertian nature of the finger image. So, Shapelet reconstruction extracts the ridge-valley pattern from the distorted finger structure. As mentioned earlier that Shapelet reconstruction is summation of the correlation of the shapelet function with the surface gradient in terms of slant and tilt. To perform the shapelet reconstruction using the above calculated surface normals, first we calculate the gradient of the surface normals. Then from the obtained gradient we calculate the slant and tilt value for the whole gradient field. The obtained slant and tilt is correlated with the shapelet function and summed to obtain the point cloud. The correlation function has three hyperparameters to be tuned for obtaining a proper $3D$ point cloud reconstruction. The first hyperparameter is about number of scales of the shapelet function which decides the spatial frequency which will be filtered out and reconstructed from the gradient field. The second hyperparameter deals with the standard deviation for the smallest shapelet scale, this factor decide the unit span of the shapelet function and hence the fundamental spatial frequency. The third hyperparameter decided the scaling factor between the successive shapelet function, this decides for the frequency gap between two consecutive shapelet scales. We chose number of scales of the shapelet function $= 3$, standard deviation for the smallest shapelet scale $= 1$ and the scaling factor between the successive shapelet function $= 2$. To determine the correct number of shapelet scales for extracting the ridge-valley pattern, we performed a 6 ridge-valley point cloud matching experiment for 170 fingers recorded over 2 sessions. Each of the six experiments had a different shapelet scale varying from 2 to 7. As shown in Figure 3.21, the highest 90.5%GAR @ 0.001FAR is obtained for shapelet scale 3. It shows that the initial 3 shapelet scales produce the highest quality ridge-valley pattern among all the shapelet scales. Hence, for the Split-and-Knit algorithm, we fixed the shapelet scale to 3. Figure 3.20 also shows a visual representation that as the shapelet scale increases from $n = 2$ to $n = 7$, there is an increase in low-frequency shape information, distorting the overall finger shape.

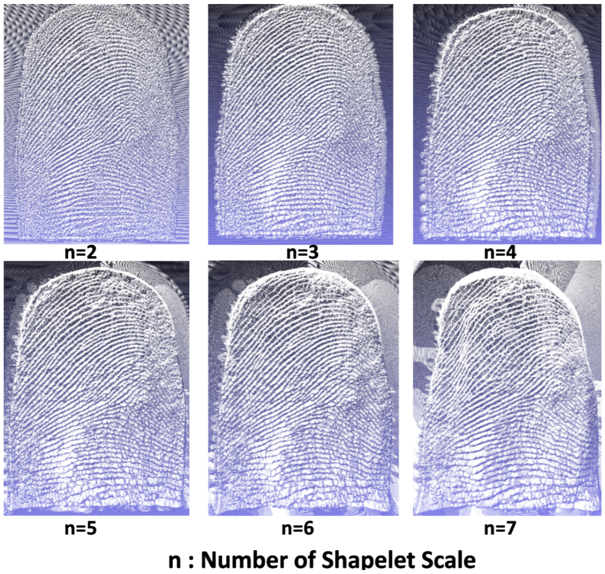


Figure 3.20 With increase in the number of shapelet scale(n), the lower frequency information increase which distort the global finger shape. We found n=3 as the optimal number of shapelet scale.

3.3.2 Finger Shape Reconstruction

3.3.2.1 Overview

To obtain the finger shape using a single camera, we need to apply Photometric Stereo. The Photometric Stereo is suited for high-frequency surface geometry and is susceptible to the slightest illumination noise in pixel values, specifically specular reflection. The finger shape is low-frequency information that gets distorted due to specular reflection and applied image processing to reduce the Lambertian nature of the finger image while obtaining the ridge-valley point cloud. So we reconstruct the shape separately. We propose two novel methods for finger shape reconstruction using a single camera:

1. Reconstruction of finger shape using a single camera under IR illumination This method belongs to the category of *computational illumination*, where we observe the object under different frequency of EM radiation to capture the desired information.

2. Reconstruction of finger shape using a camera, mirror and a *single image* The second method is about using a mirror along with the camera. This is equivalent to having a virtual camera.

Both of them are discussed in detail in the below subsections.

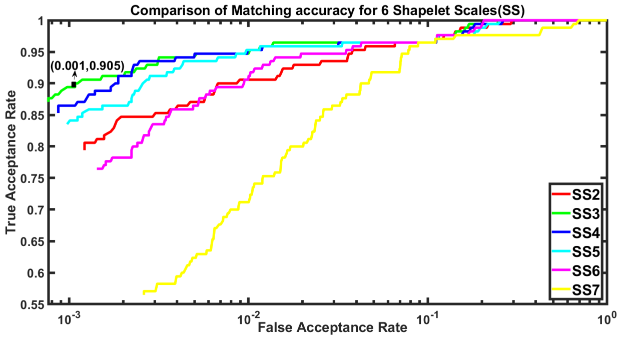


Figure 3.21 ROC curves for matching ridge-valley point cloud for fingers captured over two sessions with different shapelet scales varying from 2 to 7. The highest 90.5% GAR @ 0.001 FAR is obtained for shapelet Scale 3(SS3). Hence, using 3 shapelet scales, we obtain the highest quality ridge-valley pattern.

3.3.2.2 Reconstruction of finger shape using a single camera under IR illumination:

As mentioned in section 3.2.3, to obtain the finger shape we need to overcome not only the sub-surface scattering and specular reflection but also the high-frequency ridge-valley information. As per the equation 3.33, the RGB image has the shape information intermingled with all the above three components.

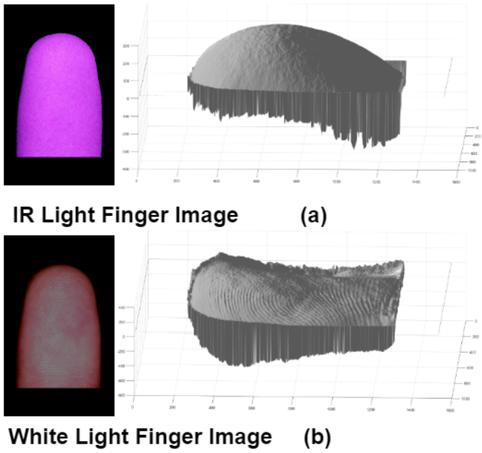
To obtain the finger shape separately we made the following observation regarding the IR light and skin interaction:

1. The finger skin consists of three layers, Epidermis, Dermis, and Subcutaneous tissue layer. The white light suffers sub-surface scattering and specular reflection due to the epidermis and dermis layers of the skin. The NIR rays penetrate deeper into the skin to the subcutaneous layer and are absorbed by the water, and fat tissue in this layer [41]. It leads to the removal of sub-surface scattering and specular reflection coming from the epidermis and dermis layers. This is depicted by the Figure 3.3.

2. Also, the ridge-valley structure resides in the epidermis layer, which is not visible in NIR illumination; hence we filter out the high-frequency information using the computational illumination method.

3. The finger surface appears homogeneous under NIR illumination. This homogeneous appearance reveals the un-distorted overall shape as depicted in figure 3.4.

4. The same camera can be used to observe visible and NIR spectra.


IR Light Finger Image (a)

White Light Finger Image (b)

Figure 3.22 Finger shape obtained using photometric stereo under (a) NIR light illumination, (b) White light illumination. Distortion in shape under the white light is due to the non-lambertian nature of the finger surface.

Figure 3.22 compares the shape of the finger obtained under white light and the NIR light using photometric stereo.

3.3.2.3 Reconstruction of finger shape using a single image and a mirror:

Using a mirror along with the camera as shown in the figure 3.23 allows us to visualize the portion of the fingers which get occluded due to the structure of the finger in an orthographic view by a single camera. We proposed to keep the mirror at 45 ° tilt beside the finger in all the three directions. A mirror is equivalent to a virtual camera, but there will be a single camera in practice.

For obtaining the shape of the finger only one single mirror kept on the side of the finger at a tilt of 45 ° is needed. Many previous researchers have obtained the finger shape using multiple cameras and Stereo Vision technique. This is for the first time we are obtaining the finger shape using a single camera and a single image. Here we obtain the *parametric modelling* of the finger using ellipses.

Steps to Obtain finger shape via parametric modeling:

1. Once we obtain the front view and the lateral view of the finger, we mask them separately using the chromaticity based masking method mentioned in section 3.3.1.1

2. Both the lateral and the frontal masks now need to be aligned in terms of rotation so that both the views have the fingers making same angle w.r.t. y-axis and in terms of the translation so that the tips of the finger in both the views are aligned and have same starting point.

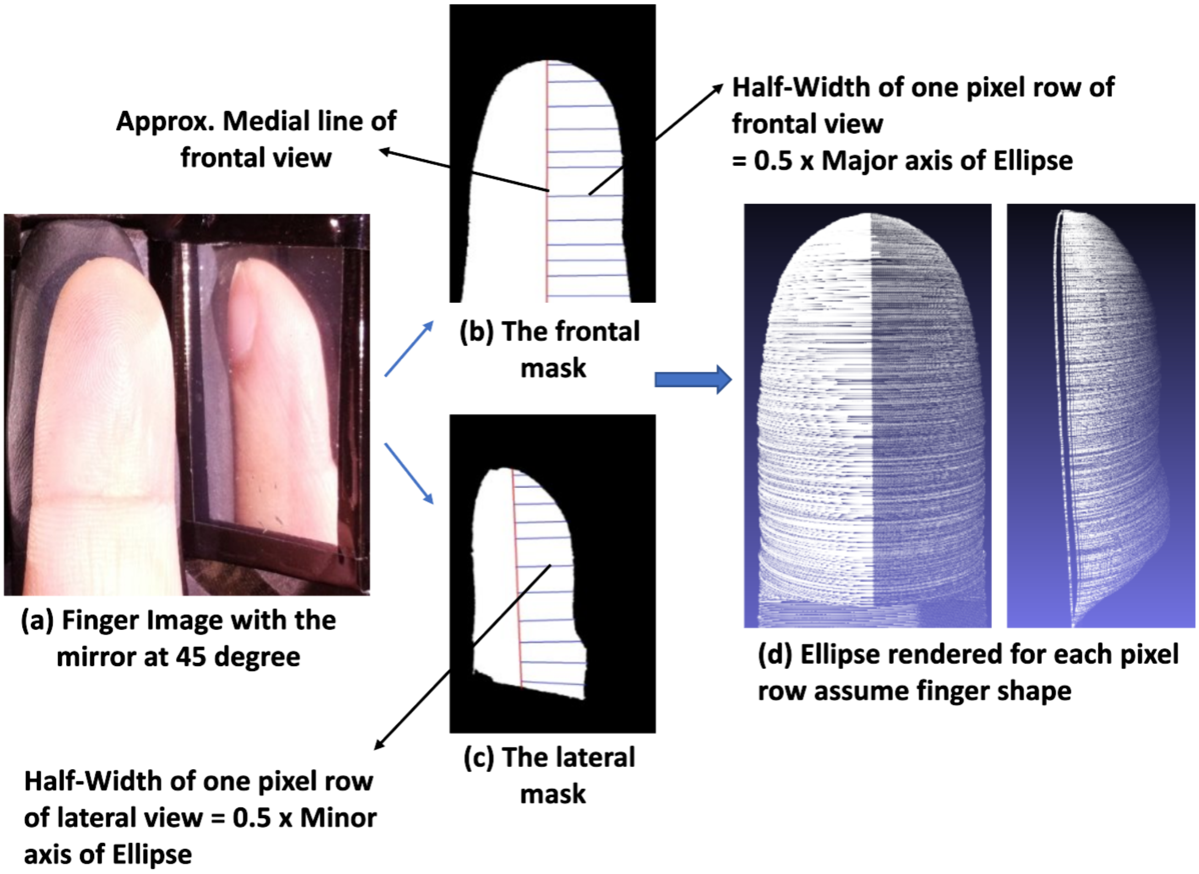

Approx. Medial line of frontal view
Half-Width of one pixel row of frontal view = 0.5 x Major axis of Ellipse
(a) Finger Image with the mirror at 45 degree
(b) The frontal mask
(c) The lateral mask
Half-Width of one pixel row of lateral view = 0.5 x Minor axis of Ellipse
(d) Ellipse rendered for each pixel row assume finger shape

Figure 3.23 Alternative method to obtain finger shape with a *single image* obtained using a mirror, camera, and parametric modeling of the finger with ellipses. This method *does not require* photometric stereo and IR illumination.

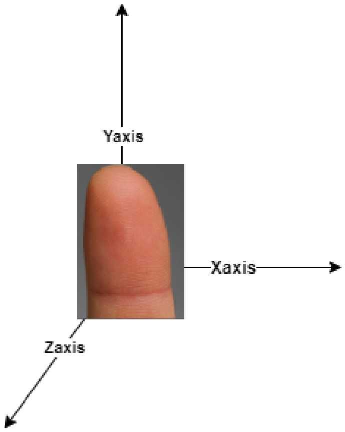


Figure 3.24 Coordinate axis assumed for the parametric modeling

3. To achieve the rotation and translation alignment for the fingers we obtained the medial line of the fingers for the masks of both the views. The medial line was obtained by the same method as mentioned in the pre-processing section 3.3.1.1 for the finger orientation correction.

4. After obtaining the medial lines as shown in the figure 3.25. The rotation and translation correction was obtained for the lateral view with the help of the medial lines and the same were applied on the lateral view to align it with the frontal view as shown in the figure 3.25.

5. For modelling the finger, we consider the y-axis across the length of the finger. The x-axis along the width and the z axis to be along the thickness of the finger as shown in the Figure 3.24. Now for each common pixel row, along the y-axis, between the two views we try to obtain one ellipse which models the finger at that particular point.

6. Based on the medial line, we calculate the half-width of the finger for each pixel row in each view. Let us suppose the two views have N corresponding pixel rows. We model an ellipse for a given pixel row with the frontal view half-width and the lateral view half-width as half of the major and minor axis, respectively. Hence, we obtain the overall finger shape by rendering the ellipses for each N-pixel row. Further, we cut the ellipses by half along the minor axis,along our supposed z axis, to resemble the finger phalange.The whole process is shown in the Figure 3.23.

7. As we know the equation of ellipse for each pixel row, we can render the x,y,z coordinates in the space and can obtain the finger shape from the elliptical parametric model. The final rendered finger shape is shown in the figure 3.26

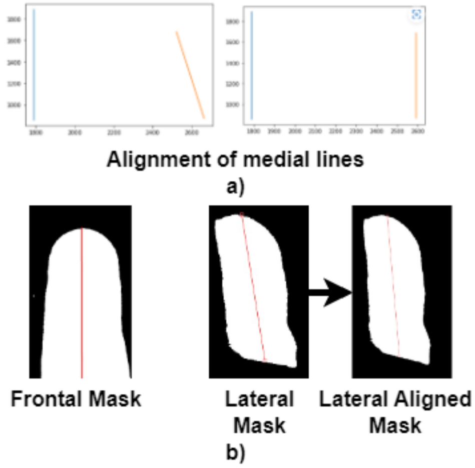


Figure 3.25 Alignment of medial axis of frontal and lateral view of the finger and rotation and translation of lateral mask based on the above calculation. This helped to get both the views in the same reference frame.

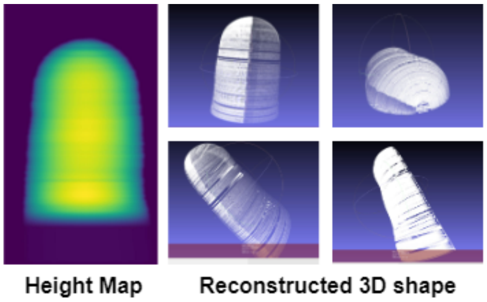


Figure 3.26 Finger Shape obtained using a single mirror and parametric modeling

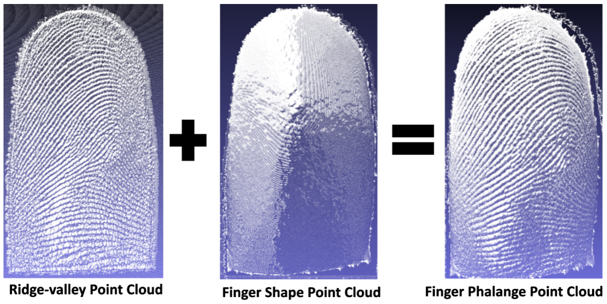


Figure 3.27 Pixel-wise addition of ridge-valley point cloud with finger shape point cloud yield finger phalange point cloud

3.3.3 3D Finger Phalange Reconstruction

We obtain the ridge-valley point cloud and finger shape point cloud separately. These represent the higher and lower frequency bands of the finger's range image. The $3D$ finger phalange is obtained by pixel-wise addition of the range image of the fingerprint and the finger shape. As we have already mentioned that both the finger shape and the fingerprint are corrupted by the noise outside their original frequencies due to sub-surface scattering, specular reflection and image processing. But when we filter out the high frequency ridge-valley pattern using the shapelets and obtain low frequency shape via the method of IR, it removes the medium frequency spatial noise, which causes finger shape distortion.

The pixel-wise addition has become possible because we are reconstructing the finger shape via IR illumination and are capturing the IR and Visible light image of the finger in one go, leading to the pixel wise correspondence. For the case of obtaining finger shape via mirrors, we need to execute separate mapping algorithm among the pixels of fingerprint and the finger shape to obtain the final finger phalange. Figure 3.27 shows the finger phalange reconstructed via the pixel-wise addition of the fingerprint and the finger shape range images.

Chapter 4

3D Fingerprint acquisiton setup

4.1 Setup Design

As mentioned earlier, the Photometric Stereo method is sensitive to the external illumination condition. This generates a need for the design of a closed setup for fingerprint reconstruction. Designing a hardware setup from scratch requires lots of initial analysis to obtain a perfectly carved system. We carried out the anaysis for designing a Photometric Stereo Setup in three steps:

1. Simulation Experiments: These were conducted to decide upon the measurement parameters of the setup, e.g. Shape of the setup, Height of LED and Camera, camera resolution and power of LEDs and their relative positions.

2. Prototype setup: After deciding the parameters in a simulation based setup we needed to consider the real world factors and adjustments needed in the setup.

3. 3D printed setup: This is the final design of the setup which is obtained after the final confirmation of each and every parameter of the setup.

4.1.1 Simulation Experiment

For determining the exact locations of LEDs, the height of the LED and camera, resolution of the camera, the brightness of the LED, and angular orientation of the LED, we simulated the Photometric Stereo on Blender [1]. We sculpted a finger phalange using the blender by modifying an ellipsoid and patterning it with the fingerprint. Then we simulated the LED light from multiple direction. For letting the uniform intensity of light fall on the finger irrespective of the direction of the illumination, we arranged LEDs in a circle centered around the finger. The placement of the each LED was also done at the same height for the same reason. Then we placed the camera at the same height as the LEDs to automatically satisfy one of the assumptions of the Photometric Stereo. The assumption requires that no direct component of light from LEDs should reach the camera, i.e. the LEDs should not be directly visible from the camera. We took three variable parameters, the height of the LEDs and the Camera, power of LEDs and the radius of the LED circle. Then for each of the parameter combination, we clicked the image of the finger under seven illumination condition. From each seven image collection we reconstructed the 3D finger with the Photometric Stereo. The best parameters were obtained for the minimum Hausdroff distance between the sculpted phalange and the reconstructed phalange. The best parameters came out to be as shown below:

1. Height of LEDs and Camera = $14\ cm$
2. Radius of LEDs = $5.25\ cm$
3. Power of each of the white light LEDs = 1W

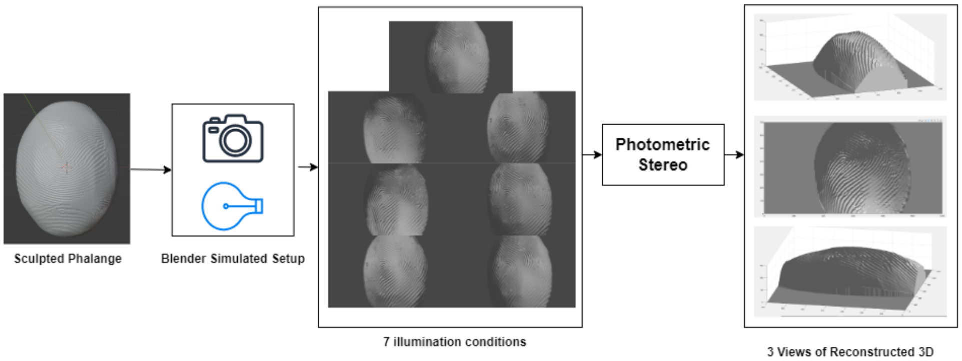


Figure 4.1 Image depicts the process of simulation of Photometric Stereo in Blender for reconstructing a sculpted finger. The sculpted finger is designed in the Blender and then imaged under 7 simulated illumination directions. On applying Photometric Stereo to theses images, we can obtain the reconstructed 3D for the sculpted finger.

4.1.2 Prototype Setup

After obtaining the parameters from the simulated setup, we needed to create a lab prototype to obtain the reconstruction of a real finger. For that we needed a cylindrical box of radius $5.25\ cm$ and height $14\ cm$ with a grove for the finger at the bottom and placement of Camera, LED and their control circuit on its roof. To design such a box we took a PVC pipe of the desired dimension. The figure 4.2 shows the setup and the schematic of different parts of the setup. Following are the take-aways after experimenting with the prototype setup:

1. **Calibration:** This setup is used to carry out the basic experiments of camera calibration and LED Light Direction Calibration, explained in the next chapter.

2. **Finger Groove Position and dimension:**Initially we kept a clay support for keeping the finger but after fixing the camera and LED positions we replaced it with a fixed size groove in the $3D$ printed setup. The details of the finger groove is given in the next section.

3. **Ergonomics:** We learned that keeping the finger in the bottom of the setup pose ergonomic difficulty hence we designed the $3D$ printed setup in a way that data can be collected in both ways, i.e. keeping the finger at the bottom and camera at the top or keeping the camera at the bottom and showing the finger from the top.

4. **LED Circuit Robustness** We also observed that the LEDs are prone to disconnection over slight movement hence we designed special inclined LED grooves to make the circuit connections robust to the movement after calibration.

5. **Finger Backgrounds** We also observed the finger under multiple backgrounds as the quality of the finger image is highly affected by the image background [3]. After obtaining the $3D$ fingerprint under green, blue, purple and black background, we found the best fingerprint to be captured with black background.

4.1.3 3D printing the setup

After the finalization of all setup parameters, we came up with the following parts for the setup as shown in the figure 4.2. Starting from the bottom, we have a groove for the finger placement. Initially we did not know at what position we can best focus the LEDs and till what extent any generic finger can be inserted in the setup. After collecting the data for many individuals and considering the average human finger dimensions across males and females, we decided the length of the finger groove and positioned it in the center of the semicircular hole where finger was inserted. Just above the finger groove plate a cylindrical structure of $14cm$ was placed with a semi-circular opening for the finger insertion. This part was also responsible for separating out the Camera and LED from the finger by $14\ cm$. Above the cylindrical structure, a frame with arrangements for camera and LED was kept. This frame was specifically designed with 14 holes at the boundary for inserting the LEDs. The holes were designed at an inclination of 20° from the vertical. This was done to ensure that the LEDs are focused towards the center of the box where the finger phalange had to be placed. In the center of the frame has a groove for the raspberry pi camera. The camera was placed in a way that the image plane of the camera became parallel to the plane of the finger, giving an orthographic viewpoint. Above the frame, we kept the circuit box. The circuit box was created with a mount for the raspberry pi and the LED control circuit. The PCB was mounted above the raspberry pi as shown in the Fig. 4.3. The circuit box also had a hole near the top to provide space for the connection wires for providing power supply to the circuit. After the circuit box. Above the circuit box, lid was placed to prevent external illumination to interfere with the process of Photometric Stereo.

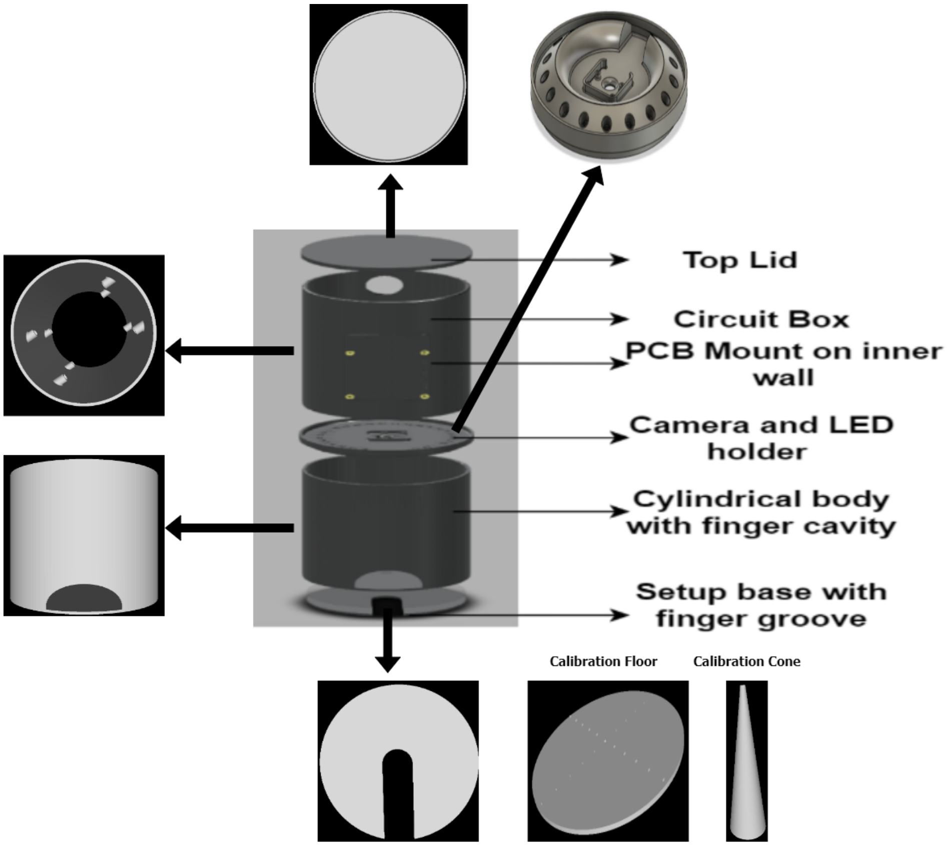


Figure 4.2 The figure details the components of the $3D$ printed setup for fingerprint capture and for further reconstruction using Photometric Stereo.

Apart from the setup, two extra calibration parts were $3D$ printed. We got a circular plate to be kept on the floor replacing the groove plate. The calibration plate was marked with $1cm$ dots to provide help with calibration tasks for LED. The conical structure was used to trace the shadow for LED light calibration.

4.2 Camera-LED Control Circuit

To control the camera and the 7 LEDs in sync together we utilized a simple raspberry pi module. We chose Raspberry Pi No-IR Camera Module v2, which has an 8 megapixel $SonyIMX219$ image sensor custom-designed add-on board for Raspberry Pi, with a fixed focus lens. The chosen camera can capture both the NIR and visible light spectrum. The LEDs and the camera were controlled using a Raspberry pi $4B$ model, and a separate PCB was designed to power the LED circuit. The LED used for white illumination is of specification $4V$ and 100 mA. The LED for IR illumination is 1.6 V and 100 mA. These specifications were finalized after the simulation experiment on Blender. The 100 mA

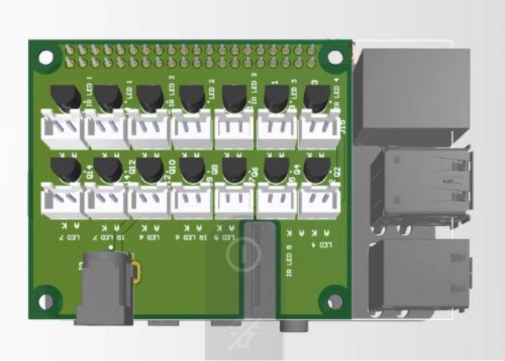


Figure 4.3 The figure shows the design circuit of the PCB for 14 LED-transistor circuit mounted on the top of the raspberry pi 4B. Together they make the control circuit for the synchronization of camera and LEDs.

current is very high to be directly generated from raspberry pi and will lead to the short circuit of the raspberry pi circuit. Hence, we designed transistor based amplifier circuit for the LEDs. So we designed 7 amplifier circuits for white LEDs and 7 amplifier circuits for the IR LED on a separate PCB as shown in the Figure 4.3.The single unit of each amplifier circuit is shown in the Figure 4.4. We used raspberry pi control signal as the base current for each amplifier. We mounted this separate PCB on the raspberry pi to obtain a complete control circuit for the Camera and LEDs. The camera was synced with the LEDs using the python code on the raspberry pi. The camera was programmed to click the image of the resolution $3200x2200$ whenever a new LED brightens up. The same camera was enabled to capture both white light and NIR images. So for reconstructing a single finger we took 14 images in the duration of 7 seconds.

4.3 Calibration

Two types of Calibration were required to be done for the setup before capturing the images for 3D reconstruction:

1. Camera Calibration

2. Illumination Direction Calibration

Both are discussed in the below subsections.

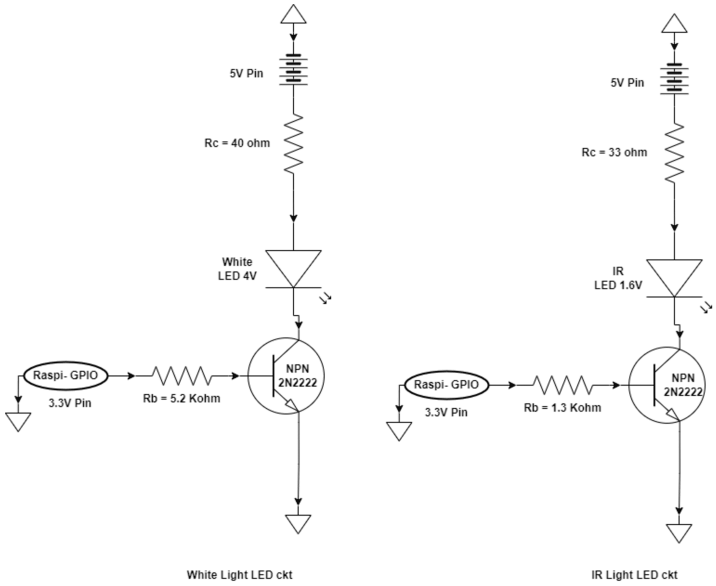


Figure 4.4 The figure shows the design circuit of the PCB for 14 LED-transistor circuit mounted on the top of the raspberry pi 4B. Together they make the control circuit for the synchronization of camera and LEDs.

4.3.1 Camera Calibration

Camera calibration is the process of determining the intrinsic and extrinsic parameters of a camera. Intrinsic parameters are the internal characteristics of the camera that do not change with respect to the camera's position and orientation, such as the focal length, principal point, and lens distortion. Extrinsic parameters are the external characteristics of the camera that change with respect to the camera's position and orientation, such as the camera's position and orientation in 3D space. The calibration process involves taking images of a known calibration pattern, such as a checkerboard, from different viewpoints and orientations. The images are then analyzed to extract the coordinates of the calibration pattern in the image plane. By comparing the known coordinates of the calibration pattern in the real world with the extracted coordinates in the image plane, the intrinsic and extrinsic parameters of the camera can be determined. Once the intrinsic and extrinsic parameters are determined, they can be used to correct for lens distortion and project 3D points onto the 2D image plane accurately.

Zhang's method, also known as the Zhang's camera calibration algorithm, is a popular algorithm for camera calibration [45]. The Zhang's method requires a calibration pattern, such as a checkerboard, with known dimensions. The calibration pattern is placed in front of the camera, and a series of images of the pattern are captured from different angles and positions. The algorithm then uses the image coordinates of the pattern points and their corresponding real-world coordinates to estimate the camera's intrinsic and extrinsic parameters, including the focal length, principal point, and lens distortion.

The key idea behind Zhang's method is to express the mapping between the image plane and the 3D world in terms of a set of linear equations that can be solved using standard linear algebra techniques. This allows the algorithm to estimate the camera's intrinsic and extrinsic parameters using a set of known equations that relate the image coordinates of the calibration pattern points to their corresponding 3D coordinates.

The major challenge for obtaining the calibration that the setup was of small size. Hence to obtain the calibration of the camera in a small FOV we printed out smaller checkerboards and used them to get the camera calibration as shown in the figure 4.5. We needed the camera calibration for the below reasons:

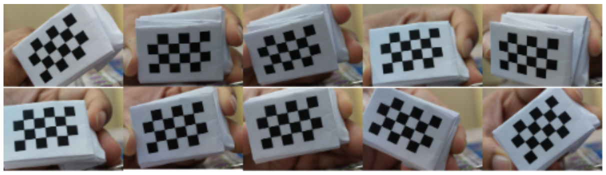


Figure 4.5 Checkerboard patterns printed on a smaller area to calibrate the setup camera

1. To focus camera on the plane of the fingerprint. Also the calibration was required to verify the focal length of the camera to be focused at the same distance as the height of the camera in the setup.

2. Calibration also led us ascertain that the plane of the camera is parallel to the base of the setup. This we ascertained via mapping the 3D point to the pixel point and verifying it in the image manually.

3. Lastly, we wanted to adjust the dof of the camera such that it contains the thickness of the finger.

4.3.2 Illumination Direction Calibration

The Illumination Direction Calibration is about finding the vector for the LED light ray. The knowledge of illumination direction is required as a pre-requisite for the Photometric Stereo algorithm to compute the surface normal of an object. There are several methods to calibrate the illumination direction. We followed the method using a simple nail as mention in [14].

The basic philosophy of this method is to keep a nail in the FOV of camera and inside the LED circle. Both the camera and the circle are placed at a height and nail is kept at a bottom surface where the object is to be imaged later. Under the illumination of a single LED, the camera will capture the shadow of the tip of the nail. If we know the location of the the tip of the shadow and the nail, then we obtain a ray, say ”shadow-ray”. Now if we trace the ray from shadow tip to the nail location and furter make it intersect with the LED circle, whose radius is known, then we obtain the x,y location of the LED. Given

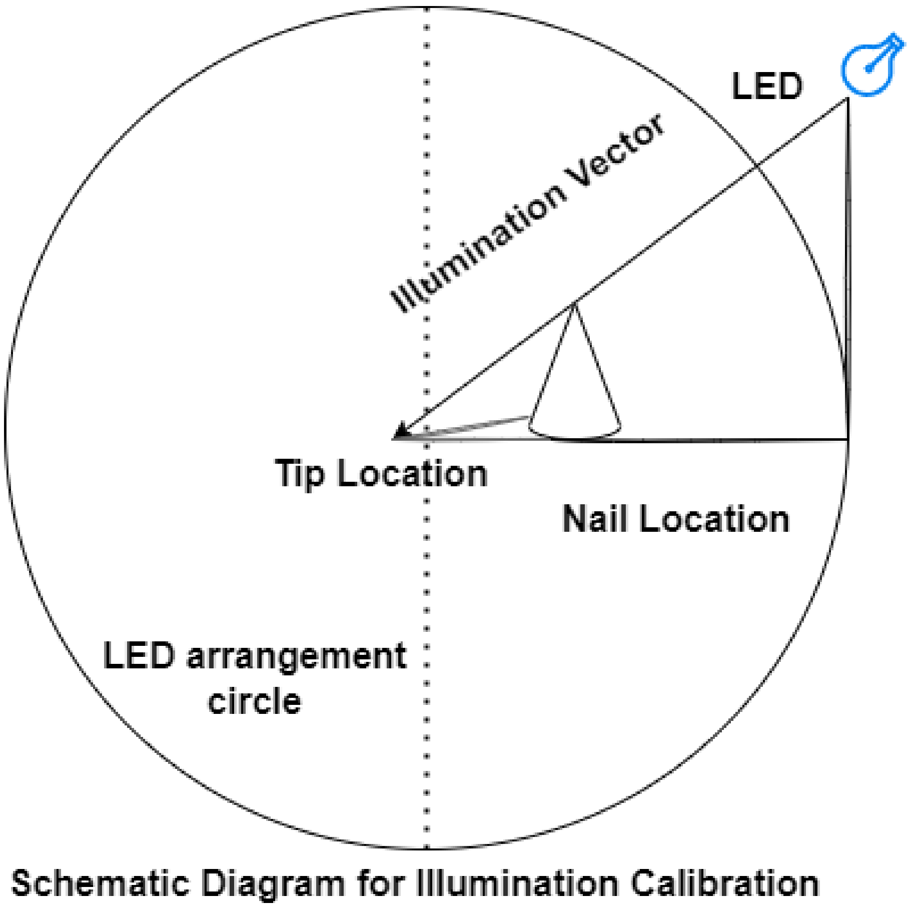


Figure 4.6 For illumination calibration various parameter are recorded offline to obtain the illumination vector for each of the LED position. a) The schematic diagram depicting the various parameters of a calibration.

that we know the height of the LED circle, we can obtain the z coordinate. This is possible because the shadow tip of the nail, the intersection point of shadow ray with the LED circle and the LED itself form a right angle triangle as shown in the Figure 4.6. Now for each of the LED we can click images of the nail under their illumination and can obtain their respective locations. The ray from the LED location to the center of the LED circle will provide the illumination direction for each LED.

Steps for Light Calibration:

1. Obtain the pixel to cm ratio for the camera by capturing the image of 1cm line drawn on a surface and storing it as ”pix2cm” variable.

2. Record the LED height (L_h) and LED circle radius (L_r) in cm and convert them to pixels by multiplying it with pix2cm. Store the variable L_{h-pix} and L_{r-pix}

3. Note the resolution of the image captured by the camera as MxN, where M is number of pixel rows and N is number of pixel columns

4. Note the pixel locations of the LED circle center,Nail position and the shadow tip, then convert them to the camera coordinate system by the following equation:

$$LED_{center-cam} = [LED_{center}(1) - M/2, LED_{center}(2) - N/2] \tag{4.1}$$

$$Nail_{loccam} = [Nail_{loc}(1) - M/2, Nail_{loc}(2) - N/2] \tag{4.2}$$

$$ShadowTip_{loc-cam} = [ShadowTip_{loc}(1) - M/2, ShadowTip_{loc}(2) - N/2] \tag{4.3}$$

5. Obtain the intersection (x,y) of the LED circle and the line segment between $LED_{center-cam}$ and $ShadowTip_{loc-cam}$. The z coordinate is L_{h-pix}. Save this as LED_{pos}.

6. Convert the obtained coordinate back to pixel coordinate system by subtracting $M/2$ and the obtained position LED_{pos} from LED center pixels.

7. Normalize the vector between the LED_{pos} and LED center. This gives the illumination direction for one particular LED. Repeat the same for all LEDs.

8. To avoid the errors we calculated the illumination direction for each LED by keeping nails at 2 positions and taking out the average of the two illumination vectors as the final vector.

Chapter 5

Experiments and Results

In this chapter we will explain the various experiments which were conducted on the reconstructed fingerprint to prove the quality of the fingerprint. In the below section we will first detail the fingerprint capture procedure using the setup designed. Then we will discuss about various fingerprint features and their matching approaches in detail followed by qualitative and the quantitative results.

5.1 Fingerprint Capture Procedure

To capture the fingerprint, we need to perform calibration of the camera and LEDs offline and then we can capture the fingerprint using the setup.

1. First of all using the Zhang's method, calibrate the camera and record the camera matrix. The focal length of the camera should be set to $14\ cm$ so that it focuses on the fingerprint plane.

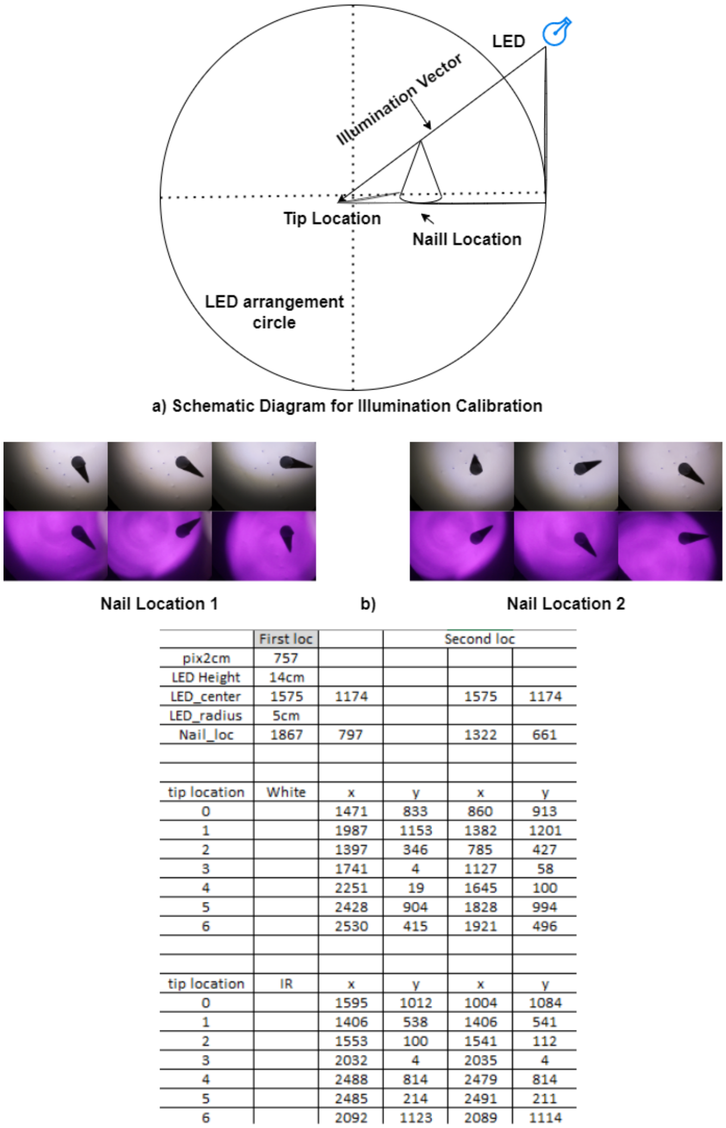
2. Capture the image of the calibration disk shown in the Figure 4.2. From the captured image, we measure the number of pixels between the marking of $1\ cm$ on the disk. This gives the pixel-to-cm ratio for the camera.

3. Keep the conical calibration structure in the field of view of the camera within the setup and capture seven images under white LED and seven images under IR LEDs. Repeat this process by keeping the conical structure at another position.

4. From the two sets of 14 images, note down the pixel location of LED center, position of cone in both the sets and also the pixel value of the tip of the shadow for the conical structure under each illumination for the two positions. Few images of the conical structure and the collected data format is shown in the figure 5.1

5. Using the collected data calculate the direction of the incident illumination for all 14 LEDs.

6. Capture the 7 images of the finger under while illumination and 7 images under IR illumination using the raspberry pi based control circuit.



a) Schematic Diagram for Illumination Calibration

b)

	First loc			Second loc	
pix2cm	757				
LED Height	14cm				
LED_center	1575	1174		1575	1174
LED_radius	5cm				
Nail_loc	1867	797		1322	661
tip location	White	x	y	x	y
0		1471	833	860	913
1		1987	1153	1382	1201
2		1397	346	785	427
3		1741	4	1127	58
4		2251	19	1645	100
5		2428	904	1828	994
6		2530	415	1921	496
tip location	IR	x	y	x	y
0		1595	1012	1004	1084
1		1406	538	1406	541
2		1553	100	1541	112
3		2032	4	2035	4
4		2488	814	2479	814
5		2485	214	2491	211
6		2092	1123	2089	1114

c) Recorded values for 7 illumination direction for White and IR images for two nail location

Figure 5.1 For illumination calibration various parameter are recorded offline to obtain the illumination vector for each of the LED position. a) The schematic diagram depicting the various parameters of a calibration. b) The sample images of the calibration cone and the shadow tip locations for White and IR light at two different nail locations. c) Values recorded for illumination calibration using the above Calibration setup.

7. From the captured white light images calculate the mean image of the finger. This mean image is used to calculate the chromaticity mask for the finger.

8. Using the mask obtained for the finger, separate out the finger from the background for all the 14 images.

9. Using the mask image, rotation and orientation is corrected for the finger and then phalange segmentation is performed.

10. The direct component for the segmented phalange is used to perform Photometric Stereo and obtain $3D$ ridge-valley point cloud. And the segmented IR phalange is used to obtain the $3D$ finger shape. These two are merged together to obtain the $3D$ fingerprint reconstruction for a finger.

We collected the data for 17 individuals for all of their 10 fingers in four sessions and used the collected 680 samples for matching experiment. We discarded some fingers with the roll angle mismatch, where the portion of the finger occluded in one session becomes visible in the other session leading to poor matching results. Partial contact-less fingerprints can be obtained using deep networks [19], which can help alleviate roll angle mismatch.

5.2 Matching Approaches

The basic purpose of capturing the fingerprint is biometric identification which is performed by matching the query fingerprint with the fingerprints stored in a given database. The fingerprint matching can be performed in both $3D$ format and $2D$ format. The traditional way of matching the fingerprint has been carried out in form of minutiae matching and this has been the most reliable method till date. Due to its popularity, the $2D$ minutiae algorithm has been extended to the $3D$ minutiae matching. After the reconstruction of $3D$ fingerprint we had two choices to consider the matching of the fingerprints in $3D$ format a) $3D$ minutiae matching and (b) $3D$ spatial feature matching. We chose to perform spatial matching instead of $3D$ minutiae matching as in the minutiae matching we will be matching only few points of the fingerprint. Matching few points do not assure the good quality of reconstruction. So we decided to go for the spatial feature matching using the LBP (Local Binary Pattern) feature. We chose LBP feature out of multiple other features like HOG, Gabor and shape context. We found the performance of LBP feature in capturing the textural pattern to be the best among the three. Also, as mentioned by [6] the LBP feature is more sensitive to the small changes in the texture and captures the depth information of the fingerprints more distinctively.

5.2.1 LBP feature matching

LBP stands for Local Binary Pattern, which is a feature extraction method used in computer vision and image processing. It was first introduced by Ojala *et al.* [25]. In the LBP feature extraction

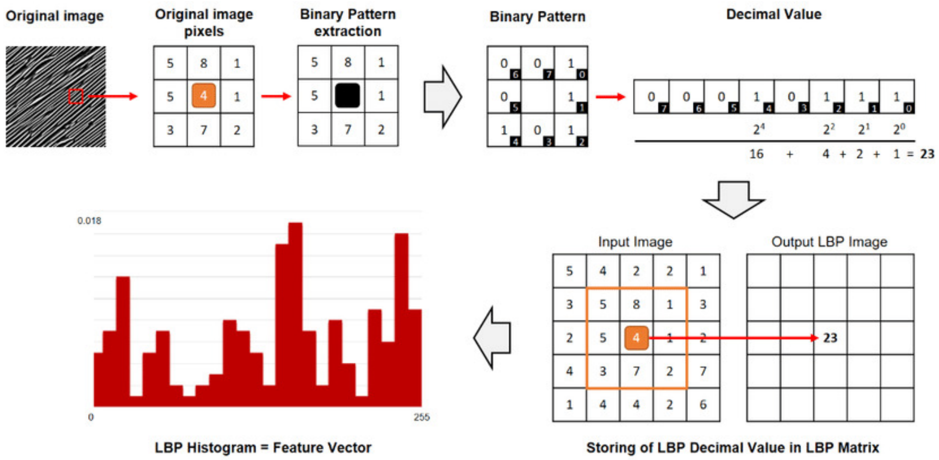


Figure 5.2 The figure illustrates the process of LBP feature extraction from a given image. The image is taken from [22]

process, each pixel in an image is compared to its surrounding pixels. Based on the comparison, a binary value (0 or 1) is assigned to each pixel in the neighborhood. The resulting binary pattern is then used to represent the local texture of the image. LBP features have several advantages, including their simplicity, robustness to noise, and computational efficiency. They have been used in a wide range of applications, such as facial recognition, object detection, and texture classification. Figure 5.2 shows the process of LBP extraction. The example figure is taken from [22] To extract LBP features, a window of a fixed size is moved over the image, and for each pixel in the window, a binary pattern is computed based on the values of its surrounding pixels. These binary patterns are then concatenated to form a feature vector that represents the texture of the image within the window. The resulting feature vector can then be used for further analysis, such as classification or clustering. The basic steps involved in calculating the LBP feature are as follows:

1. Neighborhood definition: The first step is to define a circular neighborhood around each pixel in the image. The size of the neighborhood is usually set to a fixed value, such as 3x3 or 5x5.

2. Thresholding: Next, a threshold value is determined based on the intensity of the central pixel. This threshold is used to compare the intensity values of the surrounding pixels.

3. Comparison: For each pixel in the neighborhood, its intensity value is compared to the threshold. If the intensity value is greater than or equal to the threshold, a binary value of 1 is assigned; otherwise, a binary value of 0 is assigned.

4. Binary pattern generation: The binary values for each pixel in the neighborhood are concatenated in a clockwise or counterclockwise order to form a binary pattern. This binary pattern is used to represent the texture of the image in the local neighborhood.

5. Histogram generation: Once the binary patterns have been generated for all the pixels in the image, a histogram of the patterns is computed. The histogram counts the number of occurrences of each unique binary pattern in the image.

6. Feature vector generation: The histogram is then used to generate a feature vector, which represents the texture of the image. Within a single image we can have multiple histograms and concatenating them all provides the final LBP feature vector for the image.

5.3 Matching Experiment

As mentioned before, we collected 170 fingers over the 4 sessions. We performed $4C2 = 6$ matching among the four sessions taking two at a time. For performing the matching experiment using the LBP features, the most important factor was alignment between the query and the database finger images. When the mask of the finger was created and phalange segmentation was performed then the area of pixels covered by a particular phalange was not same across the session. Also, as the finger samples were recorded on different days and after a gap, there were orientation differences among the sample despite orientation correction. Moreover, the difficulty in matching the 10 fingers for different individuals have an overlap issue. In case of matching the little finger with the thumb of certain individual even after alignment, there might be the case that the smaller finger (say little finger) matches with a patch of the bigger finger (say thumb). To deal with all the above issues we designed an alignment algorithm for the query and database finger images to calculate the score for the LBP feature matching.

5.3.1 Alignment Algorithm Details

The alignment algorithm tried to ensure that the same finger regions of two fingers are compared with each other for the LBP features. This helped to correctly identify the matching fingerprints via LBP features. Following are the steps for aligning the fingers and calculating LBP feature vector and comparing them:

1. The query finger mask q_{mask} and database finger mask db_{mask} are aligned by overlapping the bounding boxes around the finger phalange. The pixel location for the starting point of each finger tip is obtained and a translation amount is decided for the q_{mask} to get it aligned. After this step we obtain the translated mask for $q_{translated-mask}$

2. We calculate the union pixel area and intersection pixel area for the db_{mask} and $q_{translated-mask}$. Using this we obtain
$$ratio = area_{intersection}/area_{union} \tag{5.1}$$

3. If the ratio is less than 0.9 then this ascertains that the two masks belong to highly different fingers and they are not further matched. The matching score assigned to such case is $exp^{(2-ratio)}$

4. If the ratio is greater than or equal to 0.9, then we obtain the bounding box across db_{mask} and q_{mask}. We reshape the bounding box of q_{mask} to match the size of the db_{mask}. This step ensures that we obtain constant length for the LBP feature vector.

5. As the matching has to be performed for surface normals and $3D$ point cloud, we perform the translation and reshaping over each channel of the surface normal and point cloud.

6. For each of the channel of surface normal and point cloud we extracted LBP feature with neighborhood = 8, CellSize = $64x64$. Then the LBP feature vector is extracted for each channel.

7. The euclidean distance between feature vector of the corresponding channels is calculated as $dist1$, $dist2$, $dist3$.

8. The final score is calculated as

$$score = exp^{-(ratio)^4/(dist1+dist2+dist3)} \tag{5.2}$$

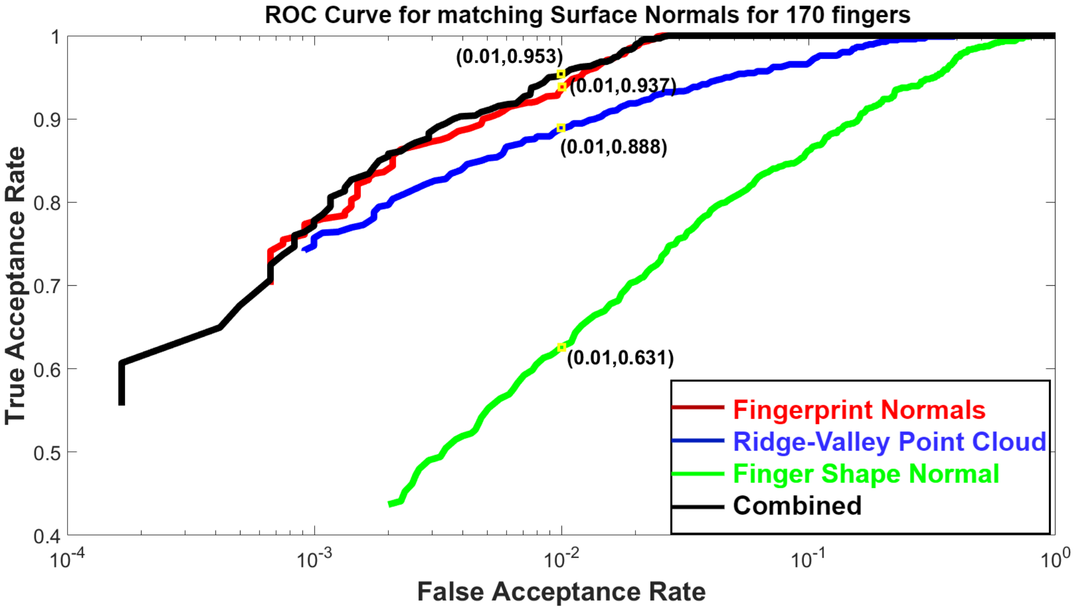


Figure 5.3 ROC curves for matching normalised LBP score for fingerprint surface normals, ridge-valley point cloud, finger shape surface normals, and combination of above attributes over 4 sessions for 170 unique fingers.

5.3.2 3D fingerprint attribute distinctiveness

To portray the discriminating nature of the $3D$ finger attributes, we matched fingerprint surface normals, ridge-valley point cloud, and finger shape surface normal using LBP features. We used 680 fingers collected over the 4 sessions for matching and then averaged out the scores obtained over the four sessions. Figure 5.3 shows the ROC curve for the three attributes and their linear combination. The combined 95.3% GAR @ 1% FAR is obtained using only $3D$ features. The results prove high-quality $3D$ reconstruction and high discriminating ability of $3D$ fingerprints as we match the whole phalange instead of a patch or minutiae points. As the shape of the first phalange of the finger does not carry highly discriminating information, we obtain low matching accuracy for the shape. However, it helps in improving overall matching accuracy. Also, in an end-to-end fingerprint classification system, the finger shape attribute can be utilised as a weak classifier. Hence in our further matching experiments for performing the comparison among our method and the existing methods, we only consider taking the ridge-valley point cloud and surface normals for their superior discriminating powers.

5.4 Comparative Analysis

We performed qualitative and quantitative comparative studies to display the improvements in the $3D$ fingerprint reconstruction by *Split-and-Knit Algorithm (SnK)* over the *Existing Algorithms (EA)* [15] based on the photometric stereo. *We considered the following three cases:*

EA: Existing Algorithms: $3D$ finger phalange obtained from gray images of the finger, assuming the finger to be lambertian. The reconstruction is obtained by photometric stereo and Frankot-Chellappa reconstruction [5].

PR: Partial Result: $3D$ finger phalange obtained after CLAHE enhancement and direct component separation of finger image. It is an intermediate result of SnK before extracting ridge-valley point cloud and overall shape addition. Hence, it has a distorted overall shape but a detailed fingerprint pattern.

SnK: Split-and-Knit Algorithm: $3D$ finger phalange obtained after CLAHE enhancement and direct component separation, followed by the addition of extracted ridge-valley point cloud and the finger shape.

5.4.1 Qualitative Comparison

Figure 5.9 shows the qualitative comparison of the $3D$ point cloud for all three cases: *SnK, PR, EA*. We conclude the following from Figure 5.9:

$1)$ The output of *EA:* Existing Algorithms based on the photometric stereo shows its inability to reconstruct the non-lambertian finger surface. It can't capture the fine details of the entire finger surface and the global shape. In comparison, the output of *SnK:* Split-and-Knit algorithm shows the fingerprint of the entire finger surface with the undistorted global shape. The zoomed patches of fingers obtained from SnK and EA in Figure 5.9 (b) show the superior quality of the ridge-valley point cloud captured by SnK over EA.

2) On comparing the *PR* (Partial Results) and the *EA* (Existing Algorithms), the reconstruction for the *PR* captures the finer fingerprint details for the larger finger area. It proves the utility of applying the direct component separation and the CLAHE enhancement.

3) On comparing the *SnK* (Split-and-Knit) and the *PR* (Partial Results), the PR reconstruction has the medium frequency noise mixed in shape and the fingerprint, which distorts the finger structure. The improved result of SnK over PR shows the utility of finger shape addition to the extracted ridge-valley point cloud.

5.4.2 Quantitative Comparison

For obtaining the quantitative comparison among the three cases: *SnK (Split-and-Knit)*, *PR (Partial Results)*, *EA (Existing Algorithm)*, we reconstructed and matched 170 fingers captured over 4 sessions using LBP features for each of the three cases. We matched SnK, PR, and EA using surface normals to show the improvement in fingerprint reconstruction quality as shown by the ROCs in Figure 5.4. We also matched the ridge-valley point cloud obtained for SnK and EA to show the improvement in pattern capture as shown by the ROCs in Figure 5.5.

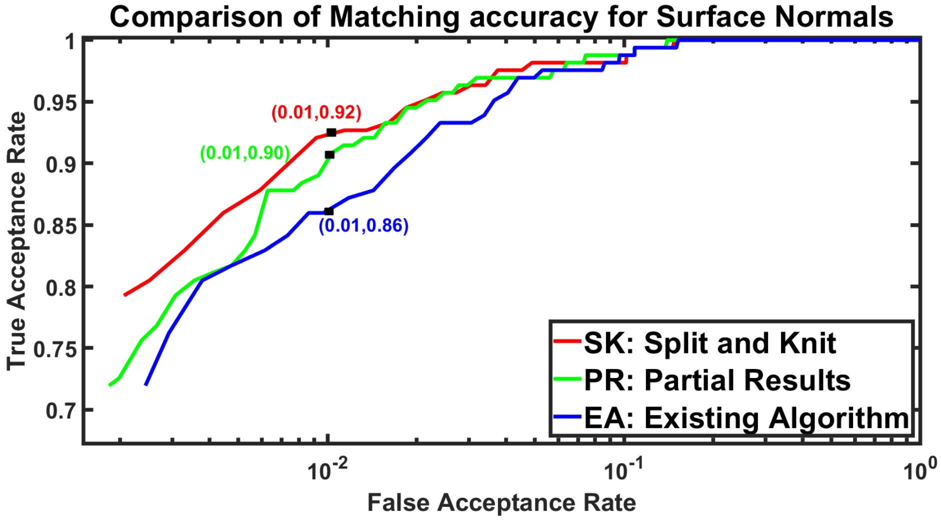


Figure 5.4 ROC curves for matching fingerprint surface normals for *SnK:* Split-and-Knit algorithm, *PR:* Partial Result after reducing the non-lambertian nature of finger image with the distorted overall shape, and *EA:* Existing Algorithms output based on the photometric stereo. The best ROC curve and 92%TAR@0.01FAR are obtained for *SnK*, displaying its highest reconstruction quality.

1) The 6% improvement in TAR@0.01 FAR for the surface normal matching of *SnK* as compared to the *EA* displays the better quality reconstruction by SnK, as shown in Figure 5.4. The 11.7% improvement in TAR@0.001 FAR and 2% improvement in TAR@0.01 FAR for ridge-valley point cloud

matching of SnK as compared to EA displays better capturing of the fingerprint details, as shown in Figure 5.5.

2) The better ROC of surface normal matching of PR over EA in Figure 5.4 shows detailed pattern capture for PR due to the reduction in the non-lambertian nature of the finger image.

3) The better ROC of surface normal matching of SnK over PR in Figure 5.4 shows the improvement in the $3D$ finger due to the addition of the overall finger shape to the ridge-valley point cloud.

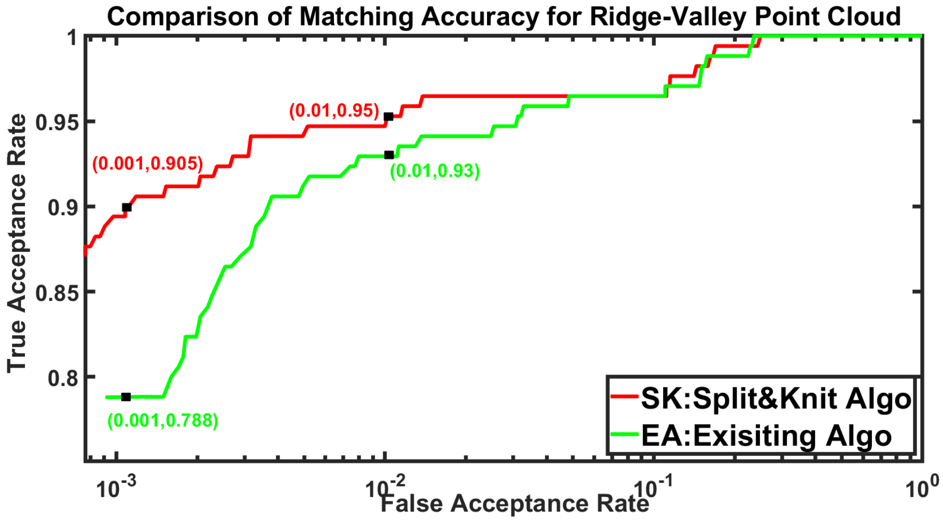


Figure 5.5 ROC curves for matching ridge-valley point cloud obtained from *SnK:* Split-and-Knit algorithm and *EA:* Existing Algorithms based on the photometric stereo. The 95% TAR@0.01 FAR and 90.5% TAR@0.001 FAR of SnK as opposed to 93% TAR@0.01 FAR and 78.8% TAR@0.001 FAR of EA proves the high quality of the ridge-valley pattern by SnK.

The Bar-Graph in Figure 5.6 summarizes the quantitative comparison, which again justifies the conclusions of the qualitative comparison, i.e., an improvement for SK of 11.7%GAR@0.001 FAR in ridge-valley matching, 6%GAR@0.01 FAR in surface normal matching, and 2%GAR@0.01 FAR in ridge-valley matching over EA.

5.5 3D-2D Conversion

Need for $3D$ to $2D$ conversion: The above method of spatial matching in the $3D$ domain was performed to prove the quality of the $3D$ reconstruction. But in an end-to-end fingerprint matching system, we can't go for the spatial matching. As this will be computationally expensive and will not be real-time. Also, the minutiae matching is an established fingerprint matching technique and is state of the art for existing fingerprint matching system. Hence to make the $3D$ reconstruction algorithm back-compatible with the $2D$ fingerprint matching technique we need to convert the $3D$ fingerprints

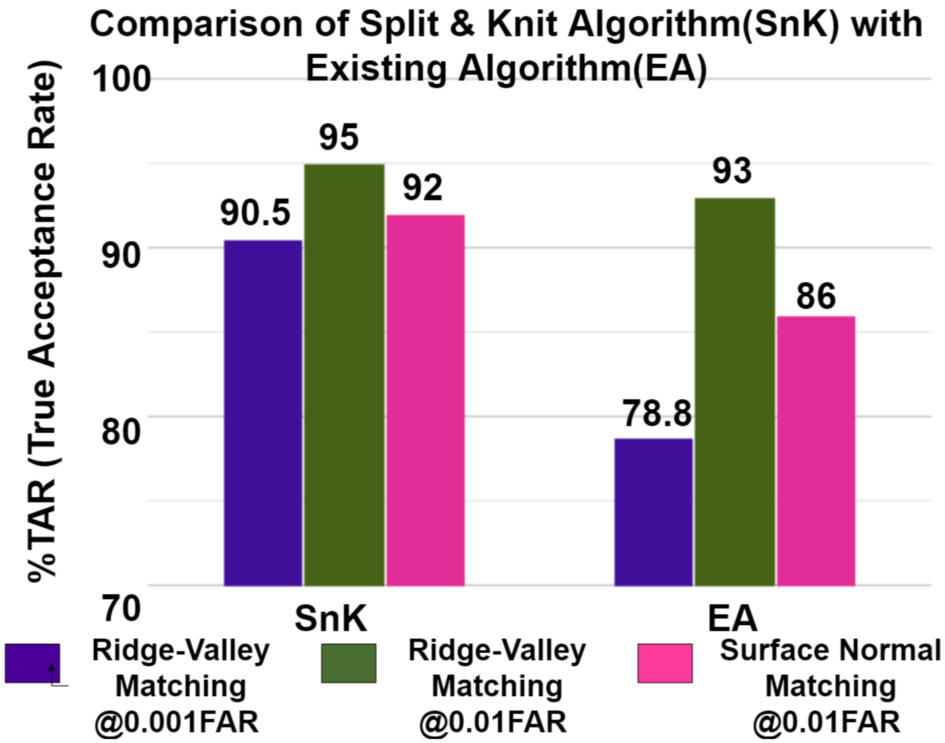


Figure 5.6 The above Bar-Graph compares the matching accuracies of *SnK:* Split-and-Knit algorithm output and *EA:* Existing Algorithms output based on the photometric stereo. The higher % TAR of SnK over EA for ridge-valley point cloud matching and surface normal matching for all cases show the higher quality of 3D phalange produced by the SnK.

into the $2D$ fingerprints. Also, even though the $3D$ minutiae matching also exists but it is less accurate and more computationally expensive. To convert the $3D$ fingerprint into a $2D$ fingerprint, there exist following methods:

1. Unrolling the $3D$ fingerprint
2. Enhancing the fingerphoto image
3. Curvature based conversion of $3D$ fingerprint
4. Novel way of obtaining $2D$ fingerprint using Split and Knit Algorithm

We will discuss each of the conversion methods briefly one by one.

Unrolling the $3D$ fingerprint: Unrolling a 3D fingerprint into a 2D fingerprint involves projecting the 3D fingerprint onto a 2D plane. This can be achieved using a mathematical transformation such as a perspective or orthographic projection. One approach to unrolling a 3D fingerprint is to use a cylindrical projection. This involves placing the 3D fingerprint on a virtual cylinder and then unwrapping the cylinder to create a 2D image. This approach can preserve the relative sizes and shapes of the features in the fingerprint. Another approach is to use a planar projection, which involves projecting the 3D fingerprint onto a flat plane. This approach can result in distortion, as the features on the edges of the fingerprint will be more stretched out than those in the center. But the method of unrolling the fingerprint has following shortcomings: 1) Distortions: The process of unrolling a 3D fingerprint can introduce errors or distortions, and the resulting 2D fingerprint may not be an exact representation of the original 3D fingerprint. 2) Loss of information: When unrolling a 3D fingerprint into a 2D representation, some information can be lost in the process. This can include information about the depth or curvature of the fingerprint, which can be important for certain applications such as fingerprint recognition. 3) Inconsistencies in feature size and shape: Unrolling a 3D fingerprint can result in inconsistencies in the size and shape of the features in the fingerprint. For example, features in the center of the fingerprint may appear larger or more distorted than those on the edges. 4) Dependency on projection method: The choice of projection method used to unroll a 3D fingerprint can have a significant impact on the resulting 2D fingerprint. Different projection methods can result in different levels of distortion and can preserve or lose different types of information. 5) Processing time and complexity: Unrolling a 3D fingerprint into a 2D fingerprint can be a computationally intensive process that requires specialized software and hardware. This can make it difficult to process large numbers of fingerprints quickly and efficiently.

Enhancing the fingerphoto image: Enhancing a finger photo to obtain a 2D fingerprint involves applying image processing techniques to enhance the ridges and valleys of the fingerprint and suppress the noise and background. There are myriads of methods proposed in literature for the enhancement. Here are some generic steps that can be followed to enhance a finger photo: 1) Preprocessing: The finger photo should be preprocessed to remove any noise or artifacts. This can involve applying filters to remove blurring or uneven lighting, or cropping the image to focus on the fingerprint region. 2) Segmentation: The fingerprint region should be segmented from the rest of the image. This can be done using techniques such as thresholding or edge detection. 3) Ridge detection: The ridges and valleys of the fingerprint should be detected. This can be done using techniques such as the Canny edge detector or the Sobel operator. Ridge thinning algorithms can then be applied to thin the ridges and reduce the width of the valleys. 4) Minutiae extraction: The minutiae (i.e., the points where ridges end or bifurcate) should be extracted from the thinned ridges. This can be done using techniques such as the skeletonization algorithm or the Harris corner detector. 5) Post-processing: The resulting fingerprint image should be post-processed to remove any remaining noise or artifacts. This can involve applying filters or smoothing techniques to the image. The major shortcoming of this method is that the quality of the resulting 2D fingerprint image will depend on the quality of the original finger photo and the effectiveness of the image processing techniques used. In some cases, it may not be possible to obtain a high-quality 2D fingerprint image from a low-quality finger photo.

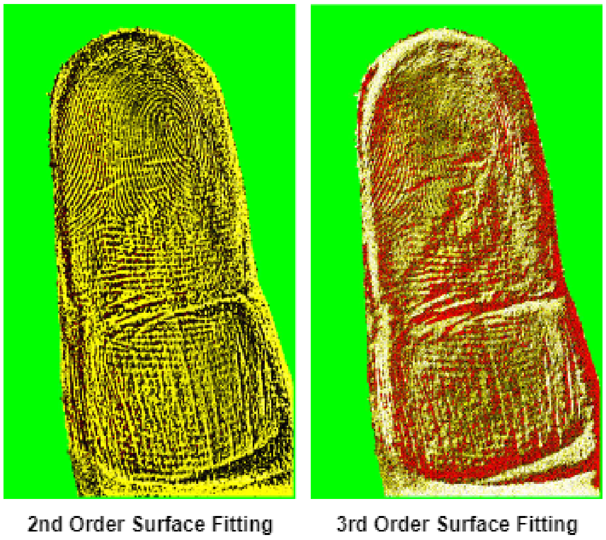


Figure 5.7 The figure compares the curvature map obtained by fitting the second order *i.e.* quadratic and the third order *i.e.* cubic surface fitting. We obtain this for a neighborhood of 15 pixels. Also the result of a quadratic surface fitting is better than the cubic fitting as the quadratic surface is more immune to noisy curvature due to resonstruction.

Curvature based conversion of $3D$ fingerprint: The differential geometric properties of a $3D$ surface remains invariant with the geometrical transformations. This information of local differential $3D$ surface geometric properties like principal curvature and surface directions can act as features for the $3D$ fingerprint and upon visualization it can provide $2D$ fingerprint. This method is not essentially for $3D$ to $2D$ conversion but this can be used to extract features from the $3D$ fingerprint and further use them for fingerprint matching. In order to find the surface curvature at a given point in the 3D space, we try to fit a surface for a specific neighborhood around a point. Using the coefficients of the fitted surface we need to construct the Weingarten curvature matrix W. The eigen values of the W matrix are the max(k_{max}) and min(k_{min}) curvature of the surface at the point where we have fit the surface. The shape index c_i at a point can be calculated by

$$C_i = \frac{1}{2} - (\frac{1}{\pi}) \cdot tan^{-1}(\frac{k_{max} + k_{min}}{k_{max} - k_{min}}) \tag{5.3}$$

Using the above method we obtained the fingerprint features by fitting cubic and quadratic surface for the neighborhood of 15 pixels. We chose the neighborhood to be 15 by observation. We found that a better feature extraction was possible by fitting the second order surface. Figure 5.7 shows the $2D$ visualization of the features obtained using quadratic and cubic surfaces, respectively. We can observe better results for the second order surface fitting as the third order surface fitting is more susceptible to the noisy structures.

The major shortcoming of this method is that apart from the local curvature at a point, we observe that global curvature of the fingerprint changes sharply at the edges as compared to center of the finger. Hence fitting the same kind of surface at both the places lead to the noise at the edges and hence loss of ridge-valley information at the edges. Also, this method is quite computationally expensive as we need to fit a surface at every pixel location to obtain the curvature index at that pixel.

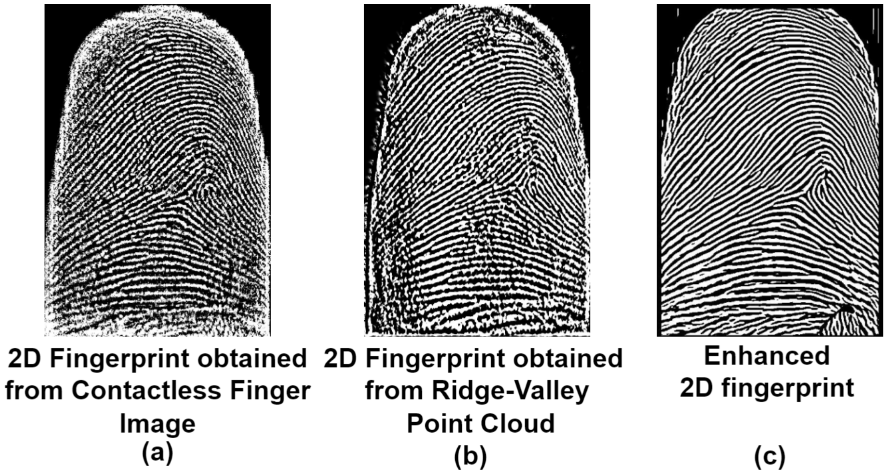

2D Fingerprint obtained from Contactless Finger Image
(a)
2D Fingerprint obtained from Ridge-Valley Point Cloud
(b)
Enhanced 2D fingerprint
(c)

Figure 5.8 2D fingerprints obtained from (a) Contact-less finger image, and (b) Ridge-Valley range image (c) Enhanced 2D fingerprint. The similarity between a) and b) proves the high-quality ridge-valley point cloud reconstruction. (c) The enhanced fingerprint assures compatibility with 2D fingerprint-matching algorithms.

Novel way of obtaining $2D$ fingerprint using Split and Knit Algorithm The Split and Knit algorithm provides an implicit way to obtain the $2D$ fingerprint. We obtain the $2D$ fingerprint in a novel way directly from the range image of the ridge-valley point clouds without any image processing on the point cloud. This method is excellent because no extra step of unrolling or image processing is required for the $2D$ fingerprint. There is no extra computation involved and we can obtain a high quality $2D$ fingerprint. Further, we obtained the enhanced fingerprint using Gabor filter-based enhancement.

The $2D$ fingerprint obtained from visualizing the ridge-valley range image appears similar to the $2D$ fingerprint obtained by processing the contact-less fingerprint, as shown in Fig 5.8. It again proves the excellent quality of the $3D$ reconstruction of the ridge-valley point cloud by SnK. Also, obtaining the $2D$ fingerprints makes the system use the Split-and-Knit algorithm(*SnK*) *compatible* with $2D$ fingerprint matching algorithms.

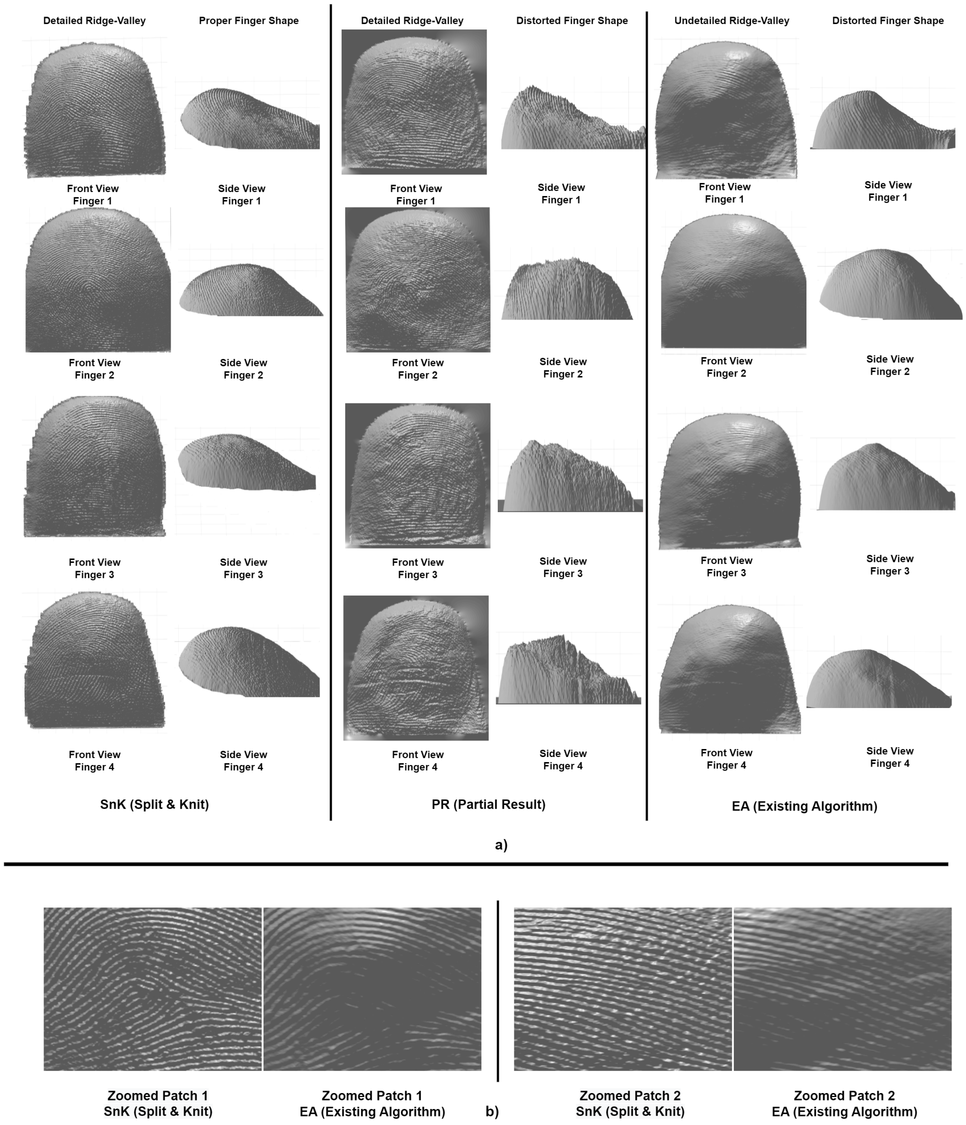


Figure 5.9 Qualitative Comparison: a) Comparison of the front view and side view of the phalange point cloud produced by ***SnK: (Split-and-Knit)***, ***PR: (Partial Result)***, ***EA: (Existing Algorithm)***. The EA [15] considers fingers to be lambertian and reconstructs using grayscale images leading to undetailed fingerprints and distorted global shape. The PR is the intermediate result obtained after reducing the non-lambertian nature before extracting the ridge-valley and adding global shape leading to a detailed fingerprint but distorted global shape. The SnK gives the best result with a detailed fingerprint and proper global shape. b) Comparison of zoomed ridge-valley point cloud for SnK and EA. SnK retrieves superior quality ridge-valley point cloud.

Chapter 6

Conclusions and Future Work

We have presented a novel algorithm that leverages photometric stereo to achieve high-quality 3D reconstruction of finger surfaces. Our approach involves a novel method for obtaining finger shape using a single camera, which is combined with ridge-valley point cloud to obtain a 3D finger phalange. Through qualitative and quantitative analysis, we have demonstrated that our algorithm produces superior results compared to existing methods, including improved matching accuracy using LBP features and direct conversion of ridge-valley point cloud into 2D fingerprint image.

However, our algorithm is just a step towards developing an end-to-end portable 3D fingerprint recognition system. Further experimentation is necessary to refine the capture setup design, camera specifications, and minutiae matching algorithms to make it more portable and user-friendly. Moreover, reducing the computational complexity of the photometric stereo method can be a future research area.

Finally, we suggest exploring the possibility of rolled-equivalent fingerprint reconstruction using a single camera to improve over roll-angle mismatch during data capture. Overall, we believe that our algorithm provides a solid foundation for future work in the field of 3D fingerprint recognition.

Related Publications

- Split and Knit: 3D Fingerprint Capture with a Single Camera
  Apoorva Srivastava, Anoop M. Namboodiri, *Thirteenth Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP'22)*, December 8–10, 2022, Gandhinagar, India

Bibliography

[1] B. O. Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.

[2] L. N. Darlow, S. S. Akhoury, and J. Connan. Internal fingerprint acquisition from optical coherence tomography fingertip scans. In *2015 Third International Conference on Digital Information, Networking, and Wireless Communications (DINWC)*, pages 188–191, 2015.

[3] M. Derawi, B. Yang, and C. Busch. Fingerprint recognition with embedded cameras on mobile phones. volume 94, pages 136–147, 01 2012.

[4] R. Donida Labati, A. Genovese, V. Piuri, and F. Scotti. Toward unconstrained fingerprint recognition: A fully touchless 3-d system based on two views on the move. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 46(2):202–219, 2016.

[5] R. Frankot and R. Chellappa. A method for enforcing integrability in shape from shading algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(4):439–451, 1988.

[6] J. Galbally, L. Beslay, and G. Böstrom. 3d-flare: A touchless full-3d fingerprint recognition system based on laser sensing. *IEEE Access*, 8:145513–145534, 2020.

[7] A. Genovese, E. Muñoz, V. Piuri, F. Scotti, and G. Sforza. Towards touchless pore fingerprint biometrics: A neural approach. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 4265–4272, 2016.

[8] K. Hoyle, M. Hsiao, L. Chair, E. Abbott, and Fox. Minutiae triplet-based features with extended ridge information for determining sufficiency in fingerprints. 04 2023.

[9] S. Huang, Y. Zhao, J. Dai, C. Chen, Y. Xu, E. Zhang, and L. Xie. 3d fingerprint imaging system based on full-field fringe projection profilometry. *Optics and Lasers in Engineering*, 52:123–130, 01 2014.

[10] A. Jacobson, Z. Duan, J. C. Bieron, and P. Peers. Deep separation of direct and global components. 2020.

[11] A. K. Jain, K. Nandakumar, and A. Ross. 50 years of biometric research: Accomplishments, challenges, and opportunities. *Pattern Recognition Letters*, 79:80–105, 2016.

[12] J. Khodadoust, M. A. Medina-Pérez, R. Monroy, A. M. Khodadoust, and S. S. Mirkamali. A multibiometric system based on the fusion of fingerprint, finger-vein, and finger-knuckle-print. *Expert Systems with Applications*, 176:114687, 2021.

[13] P. Kovesi. Shapelets correlated with surface normals produce surfaces. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 994–1001 Vol. 2, 2005.

[14] A. Kumar. *Contactless 3D Fingerprint Identification*. Edition 1, Springer Cham, 2018.

[15] A. Kumar and C. Kwong. Towards contactless, low-cost and accurate 3d fingerprint identification. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3438–3443, 2013.

[16] A. Kumar and Q. Zheng. A method and device for contactless biometrics identification. `https://patents.google.com/patent/US20160070980A1/en`, Jul 2017. US Patent No. 14798573.

[17] R. D. Labati, A. Genovese, V. Piuri, and F. Scotti. Fast 3-d fingertip reconstruction using a single two-view structured light acquisition. *2011 IEEE Workshop on Biometric Measurements and Systems for Security and Medical Applications (BIOMS)*, pages 1–8, 2011.

[18] R. D. Labati, A. Genovese, V. Piuri, and F. Scotti. Accurate 3d fingerprint virtual environment for biometric technology evaluations and experiment design. *2013 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, pages 43–48, 2013.

[19] C. Lin and A. Kumar. Contactless and partial 3d fingerprint recognition using multi-view deep representation. *Pattern Recognition*, 83:314–327, 2018.

[20] C. Lin and A. Kumar. Tetrahedron based fast 3d fingerprint identification using colored leds illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):3022–3033, 2018.

[21] F. Liu and D. Zhang. 3d fingerprint reconstruction system using feature correspondences and prior estimated finger model. *Pattern Recognition*, 47(1):178–193, 2014.

[22] M. Müller, D. Britz, L. Ulrich, T. Staudt, and F. Mücklich. Classification of bainitic structures using textural parameters and machine learning techniques. *Metals*, 10:630, 05 2020.

[23] S. Nayar, G. Krishnan, M. Grossberg, and R. Raskar. Fast separation of direct and global components of a scene using high frequency illumination. *ACM Trans. Graph.*, 25:935–944, 07 2006.

[24] S. Nie, L. Gu, A. Subpa-Asa, I. Kacher, K. Nishino, and I. Sato. A data-driven approach for direct and global component separation from a single image. In *Asian Conference on Computer Vision*, 2018.

[25] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[26] S. Pankanti, S. Prabhakar, and A. Jain. On the individuality of fingerprints. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24:1010– 1025, 09 2002.

[27] G. Parziale, E. Diaz-Santana, and R. Hauke. The surround imager™: A multi-camera touchless device to acquire 3d rolled-equivalent fingerprints. In *Proceedings of the 2006 International Conference on Advances in Biometrics*, ICB'06, page 244–250, Berlin, Heidelberg, 2006. Springer-Verlag.

[28] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. T. H. Romeny, and J. B. Zimmerman. Adaptive histogram equalization and its variations. *Comput. Vision Graph. Image Process.*, 39(3):355–368, sep 1987.

[29] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. volume 9351, pages 234–241, 10 2015.

[30] A. Sankaran, A. Malhotra, A. Mittal, M. Vatsa, and R. Singh. On smartphone camera based fingerphoto authentication. In *2015 IEEE 7th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–7, 2015.

[31] H.-L. Shen, H.-G. Zhang, S.-J. Shao, and J. H. Xin. Chromaticity-based separation of reflection components in a single image. *Pattern Recognition*, 41(8):2461–2469, 2008.

[32] C. Sousedik, R. Breithaupt, and C. Busch. Volumetric fingerprint data analysis using optical coherence tomography. In *2013 International Conference of the BIOSIG Special Interest Group (BIOSIG)*, pages 1–6, 2013.

[33] C. Stein, C. Nickel, and C. Busch. Fingerphoto recognition with smartphone cameras. In *2012 BIOSIG - Proceedings of the International Conference of Biometrics Special Interest Group (BIOSIG)*, pages 1–12, 2012.

[34] A. Subpa-Asa, Y. Fu, Y. Zheng, T. Amano, and I. Sato. Direct and global component separation from a single image using basis representation. In *Asian Conference on Computer Vision*, 2016.

[35] R. Tan and K. Ikeuchi. Separating reflection components of textured surfaces using a single image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(2):178–193, 2005.

[36] C.-w. Tsai, P. Wang, and J.-J. Yeh. Compact touchless fingerprint reader based on digital variable- focus liquid lens. 08 2014.

[37] Y. Wang, L. G. Hassebrook, and D. L. Lau. Data acquisition and processing of 3-d fingerprints. *IEEE Transactions on Information Forensics and Security*, 5(4):750–760, 2010.

[38] A. Weissenfeld, B. Strobl, and F. Daubner. Contactless finger and face capturing on a secure handheld embedded device. In *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 1321–1326, 2018.

[39] R. Woodham. Photometric method for determining surface orientation from multiple images. *Optical Engineering*, 19, 01 1992.

[40] J. C. Wu. Statistical analysis of widths and heights of fingerprint images in terms of ages from segmentation data october 15 , 2008. 2010.

[41] S. Xie, Y. Lu, S. Yoon, J. Yang, and D. Park. Intensity variation normalization for finger vein recognition using guided filter based singe scale retinex. *Sensors*, 15:17089–17105, 07 2015.

[42] W. Xie, Z. Song, and X. Zhang. A novel photometric method for real-time 3d reconstruction of fingerprint. In *Proceedings of the 6th International Conference on Advances in Visual Computing - Volume Part II*, ISVC'10, page 31–40, Berlin, Heidelberg, 2010. Springer-Verlag.

[43] V. Yalla and L. Hassebrook. Very high resolution 3-d surface scanning using multi-frequency phase measuring profilometry. volume 5798, pages 44–53, 01 2005.

[44] W. Yang, Z. Chen, J. Huang, L. Wang, and W. Kang. Lfmb-3dfb: A large-scale finger multi-biometric database and benchmark for 3d finger biometrics. In *2021 IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–8, 2021.

[45] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.

[46] Q. Zheng, A. Kumar, and G. Pan. Contactless 3d fingerprint identification without 3d reconstruction. *2018 International Workshop on Biometrics and Forensics (IWBF)*, pages 1–6, 2018.