

APPROVAL SHEET

Title of Thesis: APPROACH TO UNWRAP A 3D FINGERPRINT
TO A 2D EQUIVALENT FINGERPRINT

Name of Candidate: Ravikiran Ramdas Dighade
Master of Science, 2012

Thesis and Abstract Approved: _____

Dr. Marc Olano

Associate Professor

Department of Computer Science and

Electrical Engineering

CURRICULUM VITAE

Ravikiran Dighade

4381 148th Ave N.E. - 201,

Bellevue, WA 98007

EDUCATION

Dec 2012 M.S. in Computer Science
University of Maryland, Baltimore County

Aug 2006 B.E. in Computer Engineering
University of Mumbai, Veermata Jijabai Technological institute

ABSTRACT

Title of Document: **APPROACH TO UNWRAP A 3D
FINGERPRINT TO A 2D EQUIVALENT
Ravikiran Dighade, MS Computer Science 2012**

Thesis Directed By: **Dr. Marc Olano, Associate Professor
Department of Computer Science and
Electrical Engineering**

Fingerprints are the most widely used biometric feature for human identification because of their accuracy and uniqueness. Traditional fingerprint acquisition techniques are contact based and result in poor quality images. The new generation of non-contact based scanners capture high resolution and detailed 3D fingerprint scans, which addresses many of the problems of traditional fingerprint acquisition techniques. The majority of existing fingerprint databases available today are 2D, so there is a need for backward compatibility for the 3D scans captured. In order to solve this interoperability issue, I present an algorithm to unwrap the 3D fingerprint to its 2D equivalent image to be used in an Automatic Fingerprint Identification System.

APPROACH TO UNWRAP A 3D FINGERPRINT TO A 2D EQUIVALENT

Ravikiran Dighade

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, Baltimore County, in partial fulfillment
of the requirements for the degree of
Master of Science
2012

© Copyright by
Ravikiran Ramdas Dighade
2012

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Marc Olano, for his feedback, suggestions and help.

I am thankful to my committee for their comments. I would like to thank the VANGOGH lab for their time and support. I would also like to thank my parents who encouraged me to pursue my desires and for their love and support.

TABLE OF CONTENTS

CURRICULUM VITAE.....	III
ACKNOWLEDGEMENTS	VIII
TABLE OF CONTENTS	IX
LIST OF FIGURES	XI
LIST OF TABLES	XIII
CHAPTER 1. INTRODUCTION	1
1.1 Fingerprint as Biometric	1
1.2 Fingerprint Identification	1
1.3 Fingerprint Acquisition.....	2
1.3.1 Latent acquisition.....	2
1.3.2 Tenprint acquisition	3
1.4 Fingerprint Representation	3
1.5 Different approaches to fingerprint matching.....	4
1.5.1 Chain coded contour representations of fingerprint images	5
1.5.2 Fingerprint matching algorithm based on error propogation.....	6
1.5.3 Filterbank-based fingerprint matching.....	6
1.5.4 Filter-based feature extraction	7
1.5.5 Minutiae based fingerprint matching method.....	7

1.6 Thesis Contributions	9
CHAPTER 2. BACKGROUND AND RELATED WORK	10
2.1 3D Fingerprint Acquisition	10
Multi-view scanning	10
Structured light illumination scanning.....	11
2.2 Unwrapping	12
2.2.1 Parametric Approach	13
2.2.2 Non-parametric Approach	16
2.3 Comparison with our approach.....	18
CHAPTER 3. UNWRAPPING ALGORITHM	20
3.1 Fingerprint Data Acquisition	20
3.2 Normal Estimation	21
3.3 Curvature Estimation	25
3.4 Ridge Estimation.....	27
3.5 Convex Hull Construction	29
3.6 Texture Unwrapping	31
CHAPTER 4. RESULTS.....	34
CHAPTER 5. CONCLUSION AND FUTURE WORK.....	41
REFERENCES.....	42

LIST OF FIGURES

Figure 1.1: Ridge minutiae [2].....	2
Figure 1.2: Fingerprint features used in fingerprint matching, categorized as Level 1 (upper row), Level 2 (middle row) and Level 3 (lower row) features [2].	4
Figure 1.3: Minutiae points of sample fingerprint [8]	8
Figure 2.1: Fingerprint acquisition using a set of cameras surrounding the finger [11]...	10
Figure 2.2: Fingerprint acquisition obtained by combining a single line-scan camera and two side mirrors [11].....	10
Figure 2.3: Non-contact 3-D fingerprint acquisition using SLI technique [12]	12
Figure 2.4: Parametric unwrapping using a cylindrical model (top-down view) [11].....	14
Figure 2.5: Fingerprint unwrapping using the cylindrical model. Relative distances between points on the finger surface are not preserved after the unwrapping procedure [11].....	15
Figure 2.6: The 3D representation of finger. Vertices of the triangular mesh are naturally divided into slices [11].	17
Figure 3.1: Unwrapping Algorithm	20
Figure 3.2: Normal Variation for different values of k.....	24
Figure 3.3: Eight basic invariant surface types based on Mean and Gaussian curvature. 27	
Figure 3.4: Extracted ridges and valleys of 3D finger shape after curvature estimation (ridges are shown in blue and valleys are shown in red).....	28
Figure 3.5: Side views of 3D finger shape showing extracted ridges and valleys (ridges are in blue and valleys in red).....	29

Figure 3.6: Convex hull of 3D fingerprint	30
Figure 3.7: Unwrapped 2D fingerprint obtained after applying texture unwrapping technique (extracted ridges are shown in black).....	33
Figure 4.1 a) 3D fingerprint b) unwrapped 2D fingerprint (ridges in black).....	34
Figure 4.2 Color map showing the stretch involved after unwrap to 2D.....	35
Figure 4.3 a) Original 3D fingerprint surface b) unwrapped 2D using cylindrical unwrapping (ridges in black) c) unwrapped 2D using our approach (ridges in black)	36
Figure 4.4: Color map showing the stretch difference between cylindrical approach and our approach.....	36
Figure 4.5: Sample minutiae and distances considered for comparison	37
Figure 4.6: Stretch between minutiae points (measured in Euclidean) after unwrapping the fingerprint surface using cylindrical approach and our approach.....	38
Figure 4.7: Grid created over fingerprint surface to extract fingerprint points	40

LIST OF TABLES

Table 4.1 Distance between minutiae points (measured in Euclidean) in original 3D surface, unwrapped surface with cylindrical approach and our approach..... 38

CHAPTER 1. INTRODUCTION

Security and forensics drive a need to uniquely identify people with speed and accuracy. Other identification techniques like PIN identification are not unique and change over time. Biometrics, innate physical characteristics of the person being identified, address these problems and are now widely used for identification. Fingerprints are the most popular and studied biometric feature, their stability and uniqueness make them extremely reliable and useful for security applications.

1.1 Fingerprint as Biometric

There are nine different biometric techniques that are widely used: face, fingerprint, hand geometry, hand vein, iris, retinal pattern, signature, voice-print and facial thermograms [1]. Each of these biometric identification techniques can be used as evidence of identity, but fingerprints are considered to be most reliable and are generally accepted by experts due to their uniqueness, permanence, performance and simplicity in acquisition [2].

1.2 Fingerprint Identification

Extensive research is ongoing to create an Automatic Fingerprint Identification System in designing algorithms for effective and efficient fingerprint acquisition, feature extraction, fingerprint matching, and fingerprint classification. Ridge characteristics have been studied as a way to uniquely identify a fingerprint. The most prominent of local ridge characteristics are called minutiae points [3]. Minutiae are ridge endings and ridge

bifurcations that appear within fingerprints. A ridge ending is defined as a point where a ridge ends abruptly and a ridge bifurcation is defined as the point where a ridge forks or diverges into branch ridges. Figure 1.1 shows various minutiae features used to uniquely identify a fingerprint.

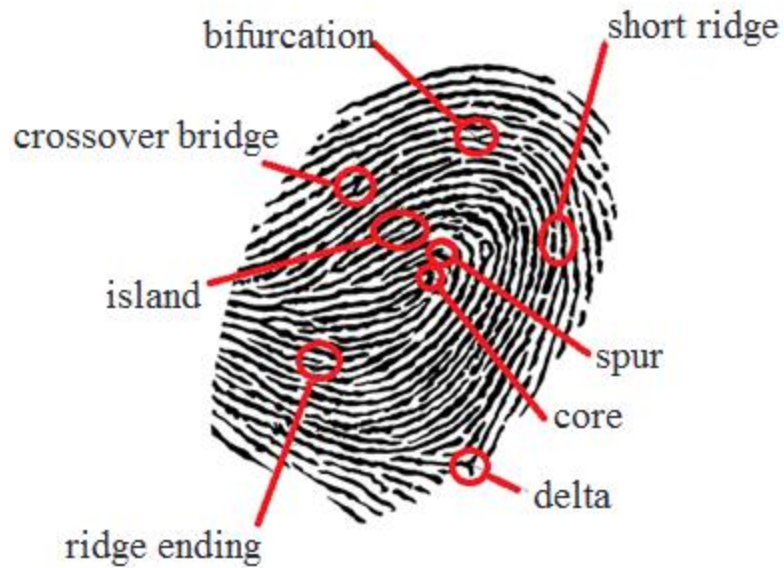


Figure 1.1: Ridge minutiae [2]

1.3 Fingerprint Acquisition

Various acquisition technologies are developed to acquire minutiae characteristics. They are broadly divided as Latent acquisition and Tenprint acquisition [2].

1.3.1 Latent acquisition

Latent or mark acquisition is used to obtain fingerprints from accidental impression left behind by a person on a surface. In the acquisition process, the residue left behind is

carefully lifted up or photographed and then processed using special treatments and imaging. The latents exhibits small portion of a finger surface, consist of very few number of features, and are of poor quality due to smudginess, distortion, and poor deposition. As a result, the fingerprint surfaces captured are less clear and include large distorted information. The actual patterns of ridges and grooves of a finger are not correctly captured.

1.3.2 Tenprint acquisition

A tenprint can be captured by using either off-line or on-line method. The off-line method includes ink-on-paper to acquire the fingerprint impression. The on-line method fingerprints are electronically scanned using optical scanners.

Optical scanners uses multispectral imaging (MSI) and touchless imaging to capture the fingerprint and are able to scan both the surface and sub-surface of a fingerprint using different wavelengths of light. This technology results in fingerprint acquisition free of smearing, slippage, and skin distortion. It produces a complete representation (from “nail to nail” and “tip to bottom”) preserving the “ground truth” of a fingerprint. They also produce higher resolution images with better quality and finer details and hence are widely used as a fingerprint sensing technology [4].

1.4 Fingerprint Representation

Once the minutiae features are acquired, it is important to be able to represent these salient and discriminatory features for fingerprint matching. The accuracy and efficiency

of fingerprint matching depends greatly on selection and extraction of features.

A large variety of features have been established based on the evidential value of fingerprints. The features are typically categorized into three levels - Level 1, Level 2, and Level 3. The higher the feature levels the finer the feature detail as shown in Figure 1.2. To extract the fingerprint features pre-compiled algorithms are used to determine the strength of ridge-valleys of fingerprint. The extracted features are then compared for matching. The more features that are matched, the better the accuracy in fingerprint identification system.

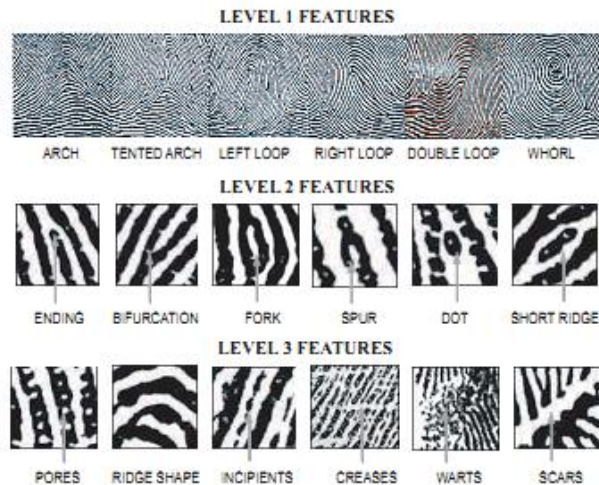


Figure 1.2: Fingerprint features used in fingerprint matching, categorized as Level 1 (upper row), Level 2 (middle row) and Level 3 (lower row) features [2].

1.5 Different approaches to fingerprint matching

Fingerprint Matching is a method of comparing two fingerprint images and finding the similarities between them. A large number of algorithms have been developed to achieve

matching and the choice of a matching algorithm depends on the representation of the fingerprint image and hence extracting the fingerprint features.

1.5.1 Chain coded contour representations of fingerprint images

The method focuses on extracting contours and global features of a fingerprint image. Initially, thinning was used for fingerprint image processing. But, thinning resulted in computationally expensive operation with less accuracy in representation. The Chain coded method addresses this problem and hence is also considered an efficient alternative for fingerprint image processing.

The first step is to threshold the fingerprint to a binary image. Then the average of the neighboring pixels are used to generate smooth chain codes without introducing spurious breaks in contours. The ridge flow field is estimated from a subset of selected chain codes. The original gray scale image is enhanced using a dynamically oriented filtering scheme together with the estimated direction field information. The enhanced fingerprint image hence forth can be used for all subsequent processing.

The orientation of the ridges and information on any structural imperfections such as breaks in ridges, spurious ridges and holes is given by the direction field estimated from chain code. The standard deviation of the orientation distribution in a block is used to determine the quality of the ridges in that block. A point where a ridge makes a sharp left turn is a candidate for a ridge ending point. Similarly when at a sharp right turn, the turning location marks a bifurcation point. Once it has all the features in fingerprint, those are compared to match the fingerprints [5].

1.5.2 Fingerprint matching algorithm based on error propagation

The method adopts ridge information and Hough transformation to find pairs of matching minutiae. The initial correspondences are used to estimate the common region and alignment of two fingerprints. A matched set is computed based on the correspondence and its surrounding minutiae matched pairs.

The subsequent matching process is guided by the concept of error propagation: For relevant neighbor minutiae, the matching error of each unmatched minutiae is estimated. The method adopts flexible propagation scheme to avoid being misguided by mismatched minutiae pairs [6].

1.5.3 Filterbank-based fingerprint matching

The method represents the fingerprint using a short, fixed length code called Finger Code. It utilizes both the global flow of ridge and valley structures and the local ridge characteristics to generate a short fixed length code for the fingerprints while maintaining high recognition accuracy. The proposed scheme of feature extraction tessellates the region of interest of the given fingerprint image with respect to a reference point. During matching only the Euclidean codes are compared and hence matching is very fast. As the representation is in the form of code, it is easy to store even in a smartcard.

A feature vector called Finger Code is the collection of all the features (for every sector) in each filtered image. This feature captures both the global pattern of ridges and valleys and the local characteristics. A feature vector is composed of an ordered

enumeration of the features extracted from the (local) information contained in each sub image (sector) specified by the tessellation. Thus, the feature elements capture the local information and the ordered enumeration of the tessellation captures the invariant global relationships among the local patterns [7].

1.5.4 Filter-based feature extraction

The method establishes many frames of reference based upon several landmark structures in a fingerprint to obtain multiple representations. For each reference point on the image, a Gabor filter is applied around the reference point. Then a feature vector also called a Finger Code is defined by computing the average absolute deviation of individual sectors. Fingerprint matching is based on finding the Euclidean distance between the corresponding Finger Codes. The method offers robust matching performance. Translation is handled by a single reference point location during the feature extraction stage [7].

1.5.5 Minutiae based fingerprint matching method

Minutiae based fingerprint matching method is considered the fastest, simplest and most robust method available. Minutiae are ridge endings and ridge bifurcations that appear within fingerprints. A ridge ending is defined as a point where a ridge ends abruptly and a ridge bifurcation is defined as the point where a ridge forks or diverges into branch ridges. A sample fingerprint with minutiae points marked is shown in Figure 1.3.

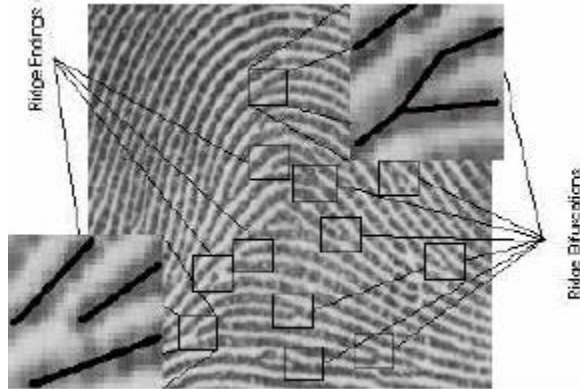


Figure 1.3: Minutiae points of sample fingerprint [8]

A few issues need to be considered while using this method for fingerprint matching. First, the skin elasticity; two corresponding minutiae may not be in the same place in two sets. One must find some sort of common reference point for two set of fingerprints. Also, false minutiae points can be introduced during the minutiae extraction and true minutiae might not be introduced at all. The method also needs to address rotation of fingerprint images.

Though the method faces issues with the poor quality images where minutiae cannot be extracted well it is widely used technique for fingerprint matching due to its simplicity and efficiency. We have incorporated this technique for fingerprint matching. Our thesis provides a method to unwrap the 3D fingerprint and convert into a 2D image which is then compared with the other fingerprint image using this technique [3].

1.6 Thesis Contributions

Offline and live scan techniques of fingerprint acquisition are contact based which has several disadvantages. These techniques lead to deformation of fingerprint features and have low capture area. They produce poor quality images and also have issues because of improper skin conditions and worn ridges. Also, the process is very slow, not user friendly and requires the supervision of a technician to capture the fingerprint images. To address these issues, there is an increase demand for non-contact based fingerprint acquisition technique. For example, Flashscan3D produces a device that generates 3D representation of a fingerprint. The scanners are high speed and very accurate in acquiring the fingerprint [2]. However, 3D touchless fingerprints need to be compatible with the legacy rolled images used in automated fingerprint identification systems (AFIS).

In this thesis, we have addressed the above challenges and developed an algorithm to achieve interoperability between 3D touchless and 2D legacy rolled fingerprints. The algorithm based on parametric unwrapping is developed to simulate a virtual rolling process to unfold 3D touchless fingerprints as 2D rolled images. Methods to evaluate and enhance the image quality of touchless fingerprints are also proposed. The resulting touchless fingerprints are “rolled-equivalent” and quite compatible with legacy rolled fingerprints.

CHAPTER 2. BACKGROUND AND RELATED WORK

2.1 3D Fingerprint Acquisition

The most recent technology for fingerprint acquisition is 3-D live scan, uses more than one camera that surround the finger for acquisition of a 3-D fingerprint. This new technology addresses the problems with the touch-based technology, which were introduced due to the contact between a finger and the surface.

Multi-view scanning

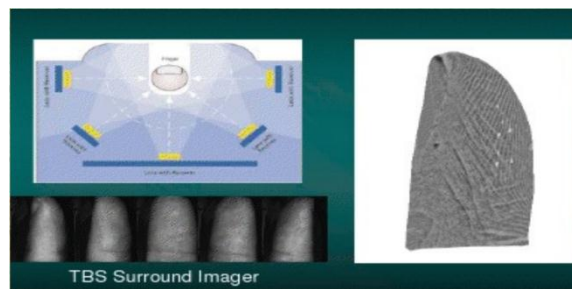


Figure 2.1: Fingerprint acquisition using a set of cameras surrounding the finger [11]

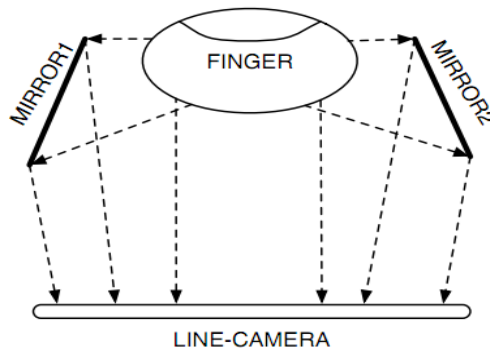


Figure 2.2: Fingerprint acquisition obtained by combining a single line-scan camera and two side mirrors [11]

Multi-view scans use views from several cameras (Figure 2.1), or views projected onto a single image plane using mirrors (Figure 2.2). For each acquired image from camera, a silhouette is obtained. This silhouette is then combined to form a 3D shape for the finger also called as shape-from-silhouette technique. The drawback of this technique is even though the shape of fingerprint is captured, detailed ridge information is lost in the process. If this information is used for unwrapping, a lot of deformation is added which is difficult to control. The obtained prints are also affected by surface color, surface reflectance, and geometric factors.

Structured light illumination scanning

The other finger scanning system is developed by Flashscan3D and the University of Kentucky which are based on Structured light illumination (SLI). The Flashscan device uses multi-pattern SLI (shown in Figure 2.3) and has the advantage of being low cost, having fast data acquisition and processing, and achieving high accuracy with dense surface reconstructions [10]. This system uses multiple, high-resolution, commodity digital cameras and utilizes SLI for acquiring 3-D scans of the fingers. These scanners enable to capture of entire 3D fingerprint (full nail-to-nail) and provides usable fingerprint area. I have used a Flashscan fingerprint dataset as the basis of this work, though the work could apply as well to other 3D fingerprint scanning technologies.

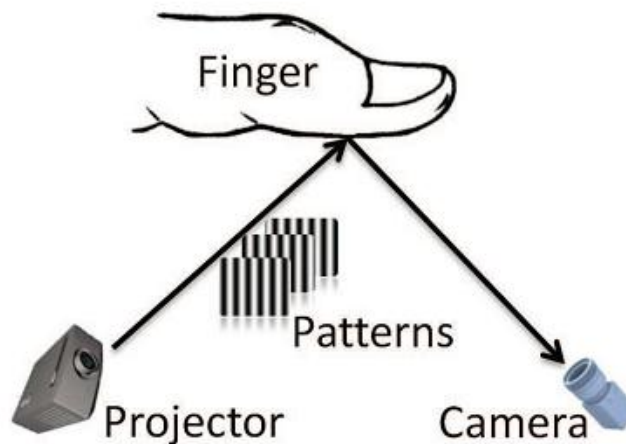


Figure 2.3: Non-contact 3-D fingerprint acquisition using SLI technique [12]

2.2 Unwrapping

Different methods have been studied to achieve interoperability between the 3D fingerprint scan and a 2D fingerprint image. Minutiae features need to be extracted. Shafaei and Yongchang's method initially unwraps the 3D fingerprint to 2D and then extracts the minutiae features from the 2D fingerprint image [10, 12], but in these methods important ridge and valley features are lost due to deformation involved in converting 3D to 2D. Our method extracts ridges and valleys in the 3D fingerprint scan itself and then it is converted to 2D. Many of the important ridge and valley information is preserved even after converting to 2D.

In general, there are two main types of unwrapping methods, parametric and non-parametric.

1. Parametric unwrapping

This method involves projecting 3D points onto a defined parametric model, for example, a cylinder or cone [11]. Once projected the 3D model is then unwrapped using simple transformations. The chosen parametric model needs to fit closely to the shape of a 3D shape that needs to be unwrapped to avoid large distortions after the unwrap.

2. Non-parametric unwrapping

This method involves mapping 3D points onto 2D while preserving the relative distances or angular relations. This method is usually used for irregular shape models. As compared to the parametric approach, non-parametric methods are more computationally intensive [12].

2.2.1 Parametric Approach

Cylinder Parametric Approach

Human fingers vary in size and shape. The thumb is usually wider than the rest of the fingers. The middle finger is often more cylindrical than the thumb. But the fingers closely resemble a cylinder in shape. While applying the parametric approach on a 3D fingerprint this approach considers cylinder as a parametric model. Chen's Cylinder based unwrapping method involves projecting 3D fingerprint points onto a cylinder and then flattening the cylinder to obtain the 2D fingerprint [11].

Mathematically, let the origin be positioned at the bottom of the finger, centered at the principal axis of the finger. Let T be a point on the surface of the 3D finger,

$$T = (x, y, z)^T \quad (2.1)$$

This 3D point is then projected (transformed) onto the cylindrical surface to obtain the corresponding 2D coordinates

$$S = (\Theta, z), \text{ where } \Theta = \tan^{-1}(x / y) \quad (2.2)$$

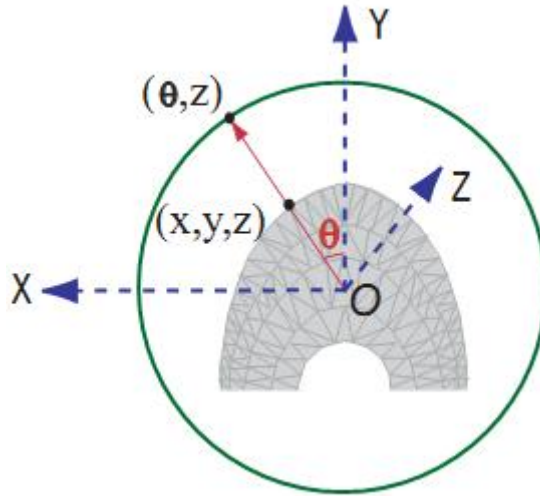


Figure 2.4: Parametric unwrapping using a cylindrical model (top-down view) [11].

Point (x, y, z) on the 3D finger is projected to (Θ, z) on the 2D plane. A top-down view of the unwrapping model is shown in Figure 2.4, where the axis points outward from the origin. The method involves constructing a triangular mesh over the 3D points. Each vertex on the triangle is then projected onto a cylinder. The points in between vertices of the triangle are mapped using linear interpolation.

This approach of unwrapping has the advantage of being simple and

straightforward, but it does possess some disadvantages. It does not preserve the relative distance between the points on the finger surface. This can be illustrated in the Figure 2.5.

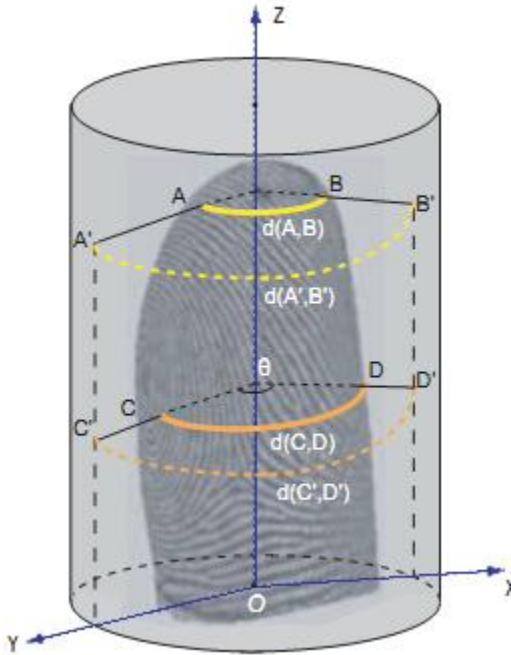


Figure 2.5: Fingerprint unwrapping using the cylindrical model. Relative distances between points on the finger surface are not preserved after the unwrapping procedure

[11]

The surface distance $d(A, B)$ between points A and B at the fingertip is much smaller than the distance $d(C, D)$ between points C and D near the middle of the finger. However, since they both correspond to the same angle Θ , the unwrapped distances $d(A', B')$ and $d(C', D')$ become equal. In general, each cross section of the finger, big or small, is projected into a fixed-length row in the projected 2D image. As a result, horizontal distortion is introduced as the fingerprint will be noticeably stretched, especially at the

fingertip.

The distance between two feature points needs to be preserved and is very important for matching two fingerprint images. As the cylinder parametric approach involves considerable stretch this distance is not maintained.

Fit Sphere Approach

Wang's method introduces a fit-sphere algorithm that uses a sphere as a parametric model [12]. The method fits a sphere on 3D point cloud fingerprint data points by mapping 3D points on fingerprint to 3D points on sphere. The distortion caused by unwrapping is minimized by controlling the local distance between neighboring points which is achieved by creating non-linear maps.

2.2.2 Non-parametric Approach

Due to the stretch involved in parametric approach research has been done on unwrapping fingerprint using non-parametric method. Non-parametric approach aims at unwrapping the fingerprint locally and preserving the geodesic distance between two points on 3D surface [11].

Equidistant Unwrapping

Chen has proposed a non-parametric unwrapping method called 'equidistant unwrapping' which involves dividing the 3D fingerprint into multiple thin parallel slices, orthogonal to

principal axis of the finger as shown in Figure 2.6 [11]. Each slice is then unfolded without stretching. Each slice is kept thin to ensure smooth unwrapping of the fingerprint.

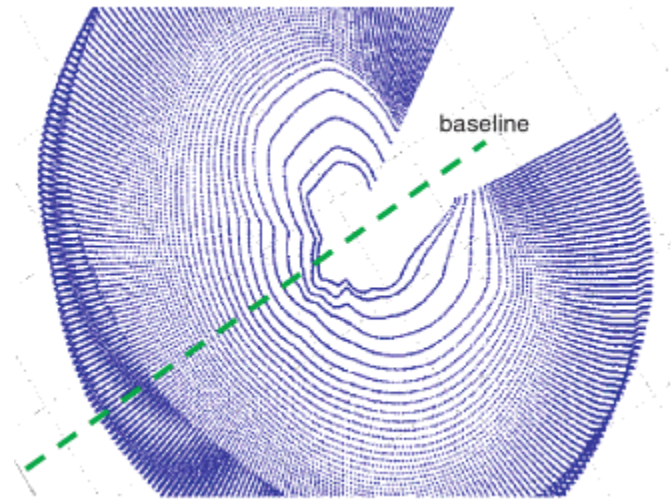


Figure 2.6: The 3D representation of finger. Vertices of the triangular mesh are naturally divided into slices [11].

In the non-parametric approach both inter-point surface distances and scale are preserved to a maximum degree.

Unwrapping using Springs Algorithm

Fatehpuria et al. proposed a ‘Springs Algorithm’ to unwrap the 3D fingerprint to 2D [13]. The algorithm allocates a certain mass to each the point of 3D fingerprint. It assumes a virtual spring between a point and its neighbors on the 3D surface. When unwrapping is added, the algorithm computes the displacement of neighboring points by minimizing the spring energy and bringing springs to their relaxed length by compressing or stretching

them. The algorithm follows an iterative approach and at each iteration the displacement of points is computed over all the 3D fingerprint points. Hence the algorithm is computationally intensive compared to other methods.

2.3 Comparison with our approach

The parametric and non-parametric approaches studied and developed to unwrap a 3D fingerprint face some challenges. Cylindrical, fit sphere or other parametric unwrapping methods have a problem of adding distortion to the unwrapped image. The image is stretched and the relative distance between extracted fingerprint features is also not maintained which is very important for fingerprint matching using a minutiae based method. Non-parametric methods are based on iteration and are computationally intensive which inculcate a performance problem in real-time applications. Our approach combines the simplicity and efficiency of the parametric approach and, at the same time, reduces the deformation added to an unwrapped fingerprint by not considering a defined model, example, cylinder or sphere.

Our approach computes a convex hull over the 3D fingerprint and uses it as a parametric model and applies texture unwrapping to unwrap the 3D fingerprint. We have used texture unwrapping via multi-dimensional scaling technique proposed by Zigelman to preserve the relative distance between extracted features [14]. As the parametric approaches which are focused on preserving angles to avoid deformation while unwrapping, our approach incorporates the Multi-Dimensional Scaling algorithm and preserves the geodesic distance between fingerprint features. The output image obtained

has minimal stretch deformation and hence obtains higher accuracy in matching over other proposed algorithms.

CHAPTER 3. UNWRAPPING ALGORITHM

Our unwrapping algorithm is divided into six major steps (shown in Figure 3.1). The first step is fingerprint data acquisition where we scan the finger to get a point cloud dataset. From second step we start processing the point cloud dataset to extract fingerprint features and eventually unwrap the 3D fingerprint while preserving the features extracted.

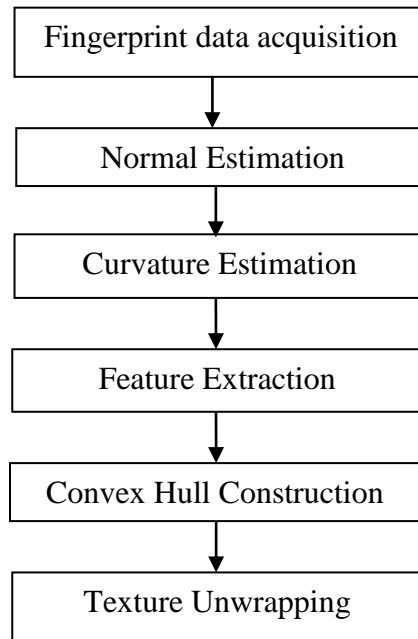


Figure 3.1: Unwrapping Algorithm

3.1 Fingerprint Data Acquisition

Fingerprints acquired using traditional acquisition methods face many drawbacks due to direct contact between sensor and skin: The fingerprint image is distorted due to

illumination, environmental factors and skin conditions, partial and degraded images due to uneven finger pressure on the scanner.

The multi-camera touchless scanners provide a 3D representation of the fingerprint. Due to the lack of contact between sensor and skin, the fingerprint image captured is of good quality and is less deformed. This scanner uses the shape-from-silhouette technique, due to which important ridge information is not captured.

We have addressed this problem by using data from FlashScan3D LLC. The scanners use multiple, high-resolution, commodity digital cameras and utilize structured light illumination (SLI) as the means of acquiring 3-D scans of the fingers. SLI are low cost and allows fast data acquisition and processing.

3.2 Normal Estimation

The 3D point cloud dataset obtained from the scanner consists of uneven surfaces consisting of ridges and valleys. To determine the orientation of the 3D surface we need to find the surface normal at each point. Given a set of point samples on the real surface, the surface normal is given by a vector perpendicular to the surface in that point. There are two possibilities to compute this vector:

1. Create a mesh over the 3D point cloud dataset using the surface meshing techniques, and compute the surface normal from the mesh.
2. Obtain the surface normal directly from the point cloud dataset using a least square plane fitting estimation algorithm.

In computer graphics, 3D objects are usually represented by triangular meshes which involve heavy processing for generation of mesh. We used a least square fitting estimation algorithm instead for normal estimation which reduces the time and storage complexity. The normal can then be computed by analysis of eigenvectors and eigenvalues of a covariance matrix created from the approximate nearest neighbors of a query point.

Approximate Nearest Neighbors

For each point P_i , in 3D space, we compute its k nearest neighbors by orthogonal decomposition of the space. The nearest neighbors are efficiently searched in huge point cloud dataset by constructing a k-d tree data structure [24]. Euclidean distance is used as the distance metric.

The Euclidean distance $\text{dist}(p, q)$ between two points' p and q is defined as

$$\text{dist}(p, q) = \left(\sum_{0 \leq i < d} (p_i - q_i)^2 \right)^{1/2} \quad (3.1)$$

Once the k-d tree is constructed over the dataset, we can find the exact k nearest neighbors for a point by searching for k closer points to the search point, but this process is time consuming. So, for further optimization, we set an upper bound on the number of points to be examined in the tree to get an approximate nearest neighbors. The advantage is that the search is stopped when k points are obtained in the upper bound. The

neighbors obtained may not be the exact nearest neighbors but they are near enough for effective covariance matrix computation and result in improved speed and memory saving. The approach provides a linear preprocessing time while taking a storage space which is moderately larger than the underlying dataset [20, 24].

Covariance Matrix

For each point \mathbf{P}_i , a covariance matrix \mathbf{C} is then computed as:

$$\mathbf{C} = \frac{1}{k} \sum_{i=1}^k \cdot (\mathbf{p}_i - \bar{\mathbf{p}}) \cdot (\mathbf{p}_i - \bar{\mathbf{p}})^T, \mathbf{C} \cdot \mathbf{v}_j = \lambda_j \cdot \mathbf{v}_j, j \in \{0, 1, 2\} \quad (3.2)$$

where k is the number of point neighbors considered in the neighborhood of \mathbf{P}_i . $\bar{\mathbf{P}}$ represents the 3D centroid of the nearest neighbors, λ_j is the j^{th} eigenvalue of the covariance matrix, and \mathbf{v}_j the j^{th} eigenvector. The eigenvalue corresponding to the smallest eigenvalue of the covariance matrix provides the normal at the point.

The value of k is decided based on the density of the point cloud data. The number should be large enough to correctly define a surface and small enough to exclude any noise. In general, the number of neighbors should result in a region larger than the noise scale but smaller than the ridge or valley scale. We can evaluate a particular choice for k by looking at the variance in the dot product between nearby normals. If k is too small, this measure will be dominated by surface noise. At the appropriate k we capture the smooth surface of each ridge or valley, leading to a low variance. When k is just a little too large, it beats with the ridge/valley undulation, so increases again. At very large

k, the variance will drop again as we capture the total surface orientation of the finger. To understand the normal variation with values of k we computed values of normal variation for different values of k and plotted a graph (Figure. 3.2).

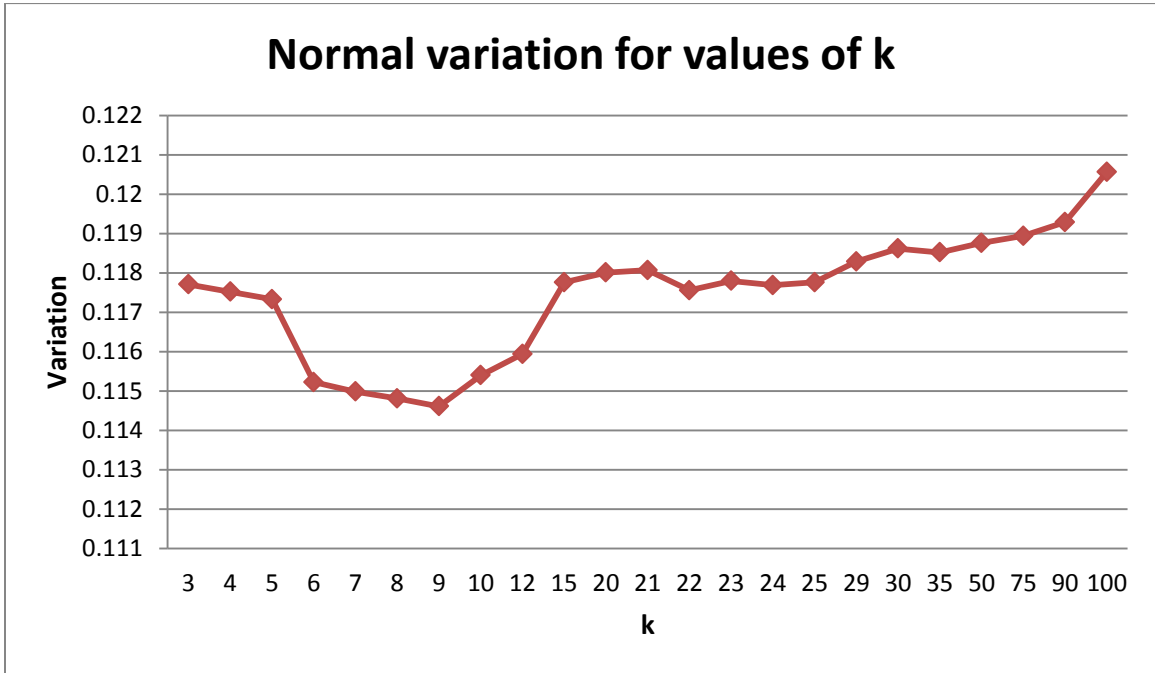


Figure 3.2: Normal Variation for different values of k.

From the graph we see that the variation of normal with k neighbors in the range of six to twelve is minimal. As the number of nearest neighbors used to compute normal increases, noise points are included in normal estimation, and from the graph, we observe that the normal variation increases gradually. If the number of nearest neighbors chosen is very small, we don't have enough points to define correct normal, and normal variation also increases as shown in graph. Based on the point cloud density of our input dataset and from the graph results we found nine neighbors ideal for our computation. This

evaluation would need to be repeated for scans with different point density, but will remain valid for all scans using the same technology and point density.

Normal Orientation

The normals estimated using the least square fitting algorithm do not consider the orientation of each surface. To orient all the normals, we choose a viewpoint \mathbf{V}_p below the nail of the fingerprint. All the normals are oriented away from the viewpoint by satisfying the following condition:

$$\vec{\mathbf{n}}_i \cdot (\mathbf{v}_p - \mathbf{p}_i) > 0 \quad (3.3)$$

3.3 Curvature Estimation

The curvature of a surface is a fundamental descriptor for shape analysis. To extract the ridges and valleys of 3D fingerprint surface we estimate curvatures at each of the points to find if a point is part of ridge or a valley. Our approach of the curvature estimation avoids mesh reconstruction and just uses neighbor points and normals to estimate a curvature. Principal curvatures and principal directions are estimated through the least square fitting of all normal curvatures related to k neighbor points.

Principal component analysis of the covariance matrix created from the nearest neighbors of a query point gives us three vectors with smallest vector in the direction of normal and two vectors tangent to the surface. This marks the local reference axis for a surface.

Biquadratic fit over the surface is given by the polynomial:

$$z = f(x, y) = ax^2 + bxy + cy^2 \quad (3.4)$$

where coordinates (x, y) are measured in tangent directions and z is measured along the normal direction, a , b and c are the coefficients of the polynomial. We find the biquadratic fit for each of the k neighboring points and use a least square method to compute the three coefficients.

The Weingarten matrix can then be constructed as:

$$W = \begin{bmatrix} a & b \\ b & c \end{bmatrix} \quad (3.5)$$

This matrix has real eigenvalues, k_1 and k_2 (principal curvatures) and eigenvectors v_1 and v_2 (principal directions).

The mean curvature H is then computed as

$$H = \frac{1}{2}(\kappa_1 + \kappa_2). \quad (3.6)$$

and Gaussian curvature K is computed as

$$K = \kappa_1 \kappa_2. \quad (3.7)$$

3.4 Ridge Estimation

Once we have the mean and Gaussian curvature of the surface at a point p , we can determine the surface types. Figure 3.2 shows the different surface types based on the sign on H and K .

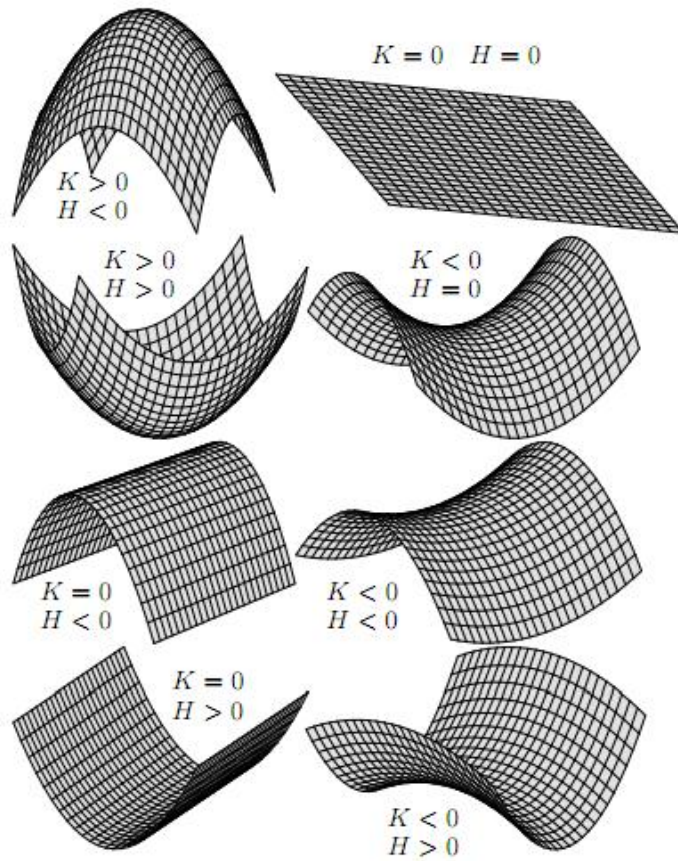


Figure 3.3: Eight basic invariant surface types based on Mean and Gaussian curvature.

For the values of $H < 0$ the surface is more convex and for $H > 0$ the surface is more concave [15, 16]. The convex surfaces on the 3D fingerprint surface are identified as ridges and concave surfaces are identified as valleys. Minutiae features of the fingerprint surface are defined by its ridges. We therefore extract just the ridges of the 3D fingerprint surface based on the sign of the mean curvature. Figure 3.3 shows the result obtained after extracting ridges and valleys after curvature estimation. Figure 3.4 shows the side views of extracted ridges and valleys.

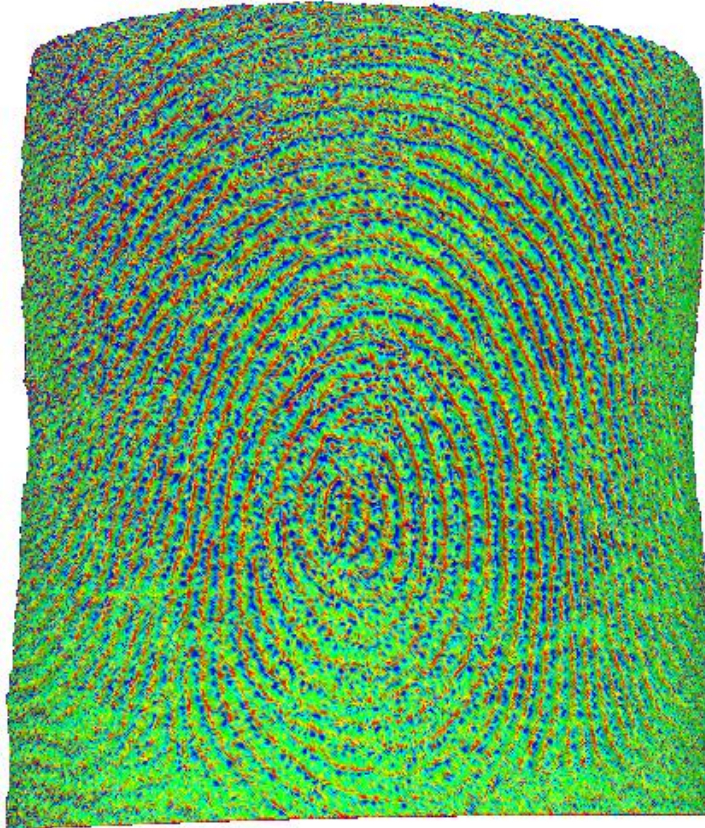


Figure 3.4: Extracted ridges and valleys of 3D finger shape after curvature estimation (ridges are shown in blue and valleys are shown in red)

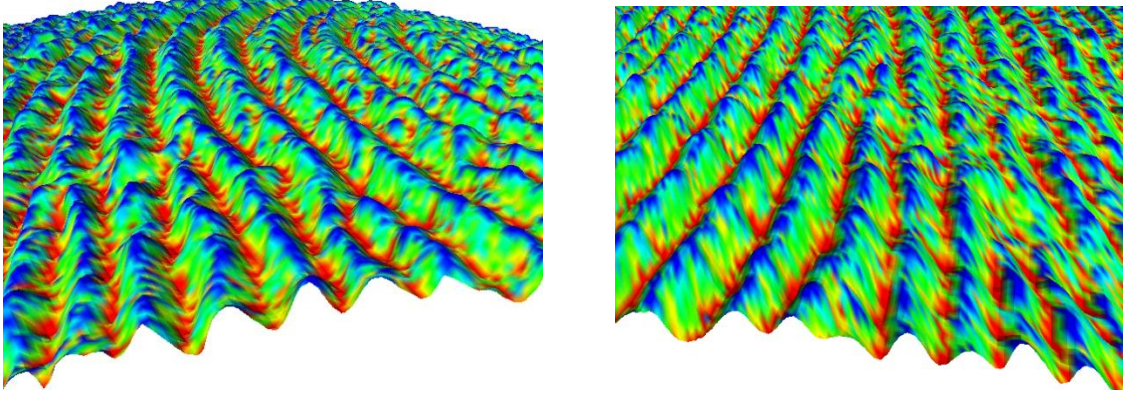


Figure 3.5: Side views of 3D finger shape showing extracted ridges and valleys (ridges are in blue and valleys in red)

3.5 Convex Hull Construction

The cylinder based parametric approach or fit sphere based approach stretch the unwrapped output of the 3D fingerprint. These approaches do not preserve the relative distance between the minutiae features required for matching algorithms. To preserve the distance between the ridges, it is important to minimize distortion. The convex hull gives an overall estimate of the 3D finger shape. Using this estimate we apply texture unwrapping to unwrap the 3D fingerprint. The convex hull is the convex model which best fits the 3D fingerprint surface.

The convex hull of a set of points is the smallest convex set that contains the points [19]. The convex hull of X can be characterized as the set of all of the convex combinations of finite subsets of points from X : that is, the set of points of the form

$$\sum_{j=1}^n t_j x_j \quad (3.8)$$

where n is an arbitrary natural number, the numbers t_j are non-negative and sum to 1, and the points x_j are in X [18].

Different algorithms are available for creating a convex hull. The Quickhull algorithm is the ideal choice for 3D fingerprint convex hull creation. The Quickhull algorithm for creating a convex hull consumes less space, includes non-extreme points and is comparatively faster than other algorithms for creating a convex hull [19]. Figure 3.5 shows the convex hull obtained after applying the Quickhull algorithm to 3D fingerprint.

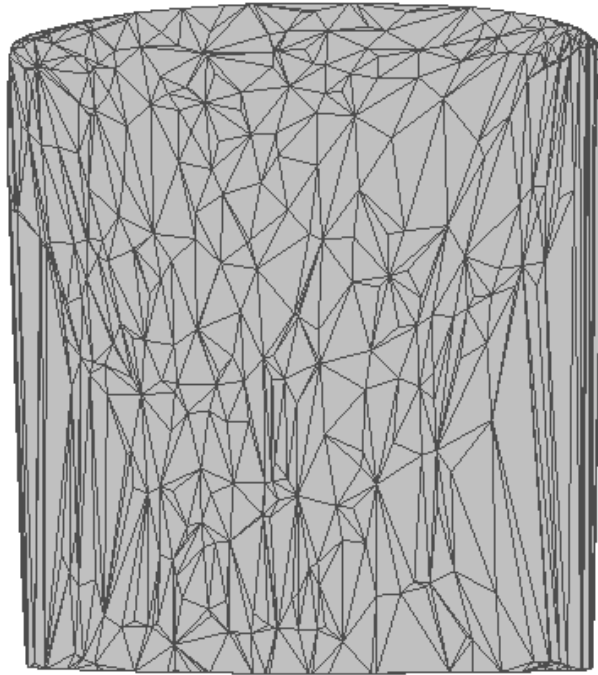


Figure 3.6: Convex hull of 3D fingerprint

3.6 Texture Unwrapping

The convex hull provides the estimate of 3D fingerprint surface and also the connectivity between the extracted ridges. The next step is unwrapping the mesh obtained by the convex hull. Various parameterization techniques are proposed for unwrapping a mesh with minimal distortion. Based on the distortion they preserve these algorithms are mainly classified as:

1. Angle preserving
2. Area preserving
3. Distance preserving

Angle preserving and area preserving mesh parameterization techniques minimize angular and area distortion respectively but result in heavy distortion in distance. But, minutiae based 2D fingerprint matching algorithms depends on minutiae distance for matching fingerprints. So, we have incorporated a distance preserving mesh parameterization technique to preserve relative distance between minutiae features.

Zigelman proposed a distance preserving technique for unwrapping a 3D model [14]. The technique uses a fast marching method and multidimensional scaling for texture mapping. MDS (multi-dimensional scaling) is a set of mathematical techniques which allow finding Geometric structure of the 3D surface. MDS computes geodesic distances between every two points on the surface which is later used for flattening the 3D surface. For efficiency, fast marching method is used to find the geodesic distance. The algorithm

then aims to preserve these computed geodesic distances while flattening the 3D surface. The geodesic distance M between vertices of the mesh is computed as:

$$M = \left(\text{dist}^2(\mathbf{x}_i, \mathbf{x}_j) \right)_{n \times n} \quad (3.9)$$

The implementation of the algorithm is explained in greater detail in Zigelman [14]. Using the algorithm, we obtain an unwrapped 2D fingerprint with minimal stretch. The extracted ridges are mapped to the 2D surface to get the 2D map. Figure 3.6 shows the unwrapped fingerprint obtained by texture unwrapping. We observe that the algorithm successfully minimizes the distance distortion with more accuracy in distance between extracted features.

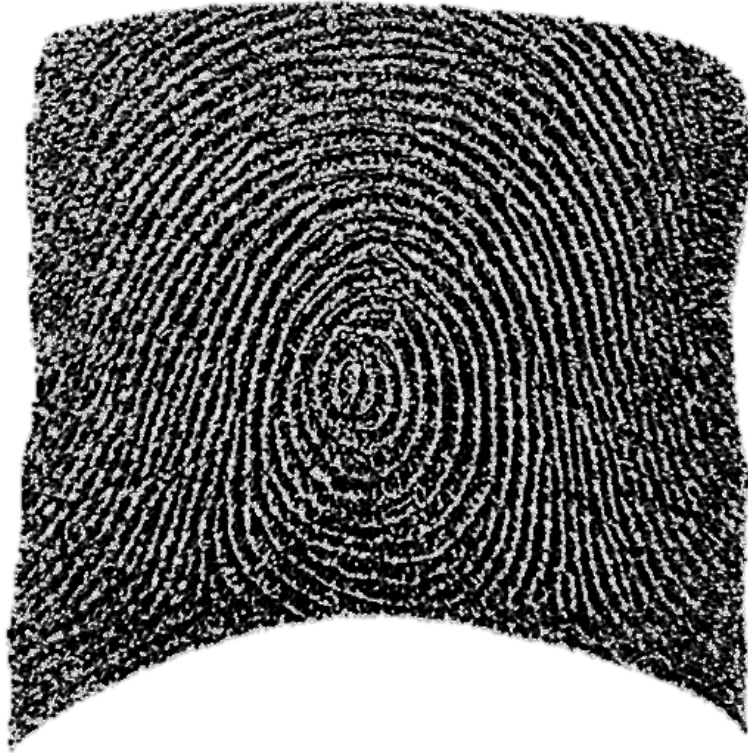


Figure 3.7: Unwrapped 2D fingerprint obtained after applying texture unwrapping technique (extracted ridges are shown in black)

CHAPTER 4. RESULTS

We applied our approach to a single scanned 3D fingerprint data obtained from FlashScan3D. The data consisted of approximately 220,000 points. Figure 4.1 shows the 3D fingerprint shape and the unwrapped image obtained after applying the unwrapping algorithm.



Figure 4.1 a) 3D fingerprint b) unwrapped 2D fingerprint (ridges in black)

As we are using a sample dataset, we do not have the corresponding 2D legacy fingerprint, and consequently are not able to make direct 3D to legacy print comparisons. Instead, we evaluate our algorithm results through comparison to the simpler 3D cylindrical unwrapping algorithm and by analyzing the stretch introduced in distances between minutiae.

The algorithm aims at reducing the distortion involved in the unwrapping process while preserving the relative distance between minutiae points. The stretch involved

during the unwrapping process can be computed by the difference between 3D fingerprint data points and the corresponding 2D points of the unwrapped fingerprint surface. We compute a color map showing the stretch.

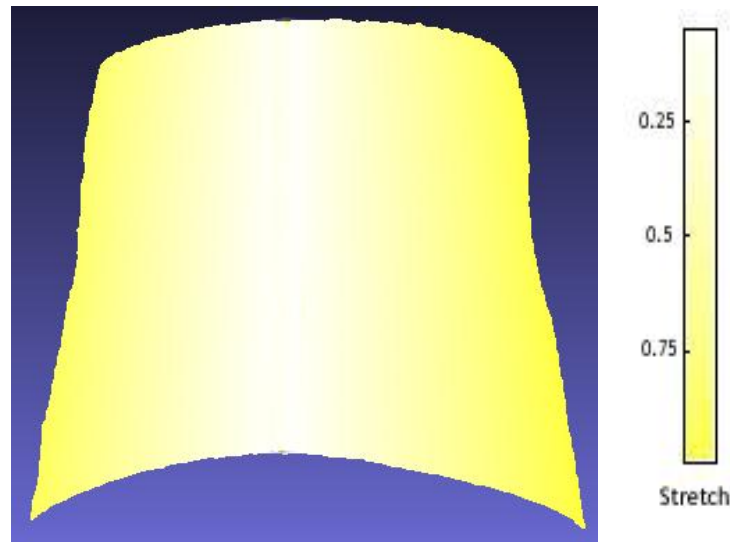


Figure 4.2 Color map showing the stretch involved after unwrap to 2D

From the color map shown in Figure 4.2, we observe that the color gradually changes from lighter to dark from center towards the end. This signifies that the stretch involved at the center of the fingerprint surface is minimal and slowly increases towards the sides. The distortion around the central region of the fingerprint is less and hence the relative distance between minutiae points is preserved. From the color map we can deduce that our algorithm preserves relative distance and lessens the distortion in the unwrapping process.

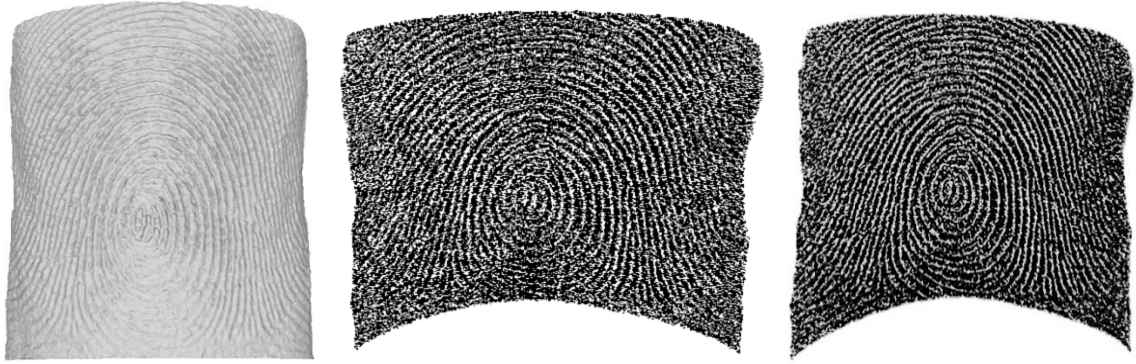


Figure 4.3 a) Original 3D fingerprint surface b) unwrapped 2D using cylindrical unwrapping (ridges in black) c) unwrapped 2D using our approach (ridges in black)

To compare our results, we computed a cylindrical unwrap of the 3D fingerprint surface. From the Figure 4.3, we can observe that the fingerprint is considerably stretched in the cylindrical unwrapping distorting the minutiae distance. We obtain a color map shown in Figure 4.4 showing the stretch difference between cylindrical and our approach.

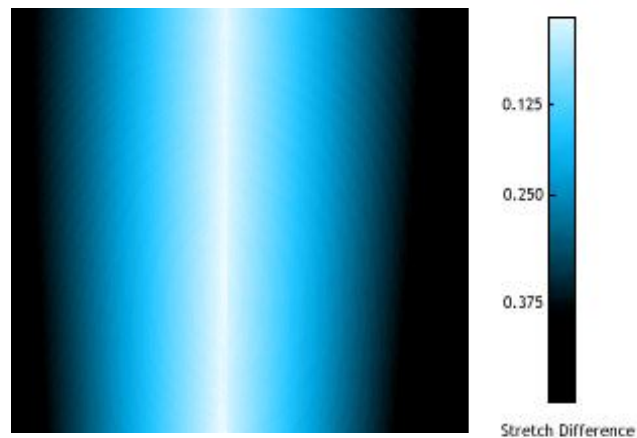


Figure 4.4: Color map showing the stretch difference between cylindrical approach and our approach

The lighter color signifies less stretch and dark signifies more stretch. From the color map, we observe that the 3D fingerprint while unwrapping is greatly stretched towards the end. Even in the mid region the unwrapping adds some stretch which distorts the relative distance between minutiae. From the results we observe that our approach preserves the stretch involved in the 3D fingerprint unwrapping to a great scale compared to the cylindrical approach.

To effectively compute the distance distortion, we extracted a few minutiae points as shown in Figure 4.5 and plotted a table and graph showing the stretch involved during unwrapping using the cylindrical and our approach.

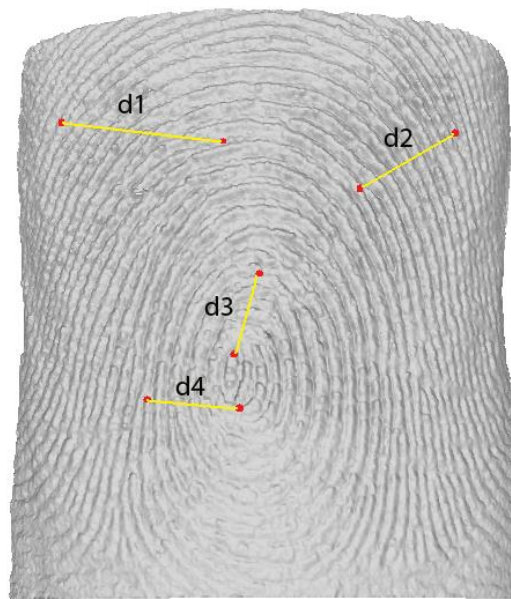


Figure 4.5: Sample minutiae and distances considered for comparison

Distances	3D	Cylindrical	Our approach
d1	4.5476	7.3189	5.2744
d2	3.4293	4.0333	3.8916
d3	1.9526	2.1757	1.8947
d4	2.3347	3.0401	2.4000

Table 4.1 Distance between minutiae points (measured in Euclidean) in original 3D surface, unwrapped surface with cylindrical approach and our approach

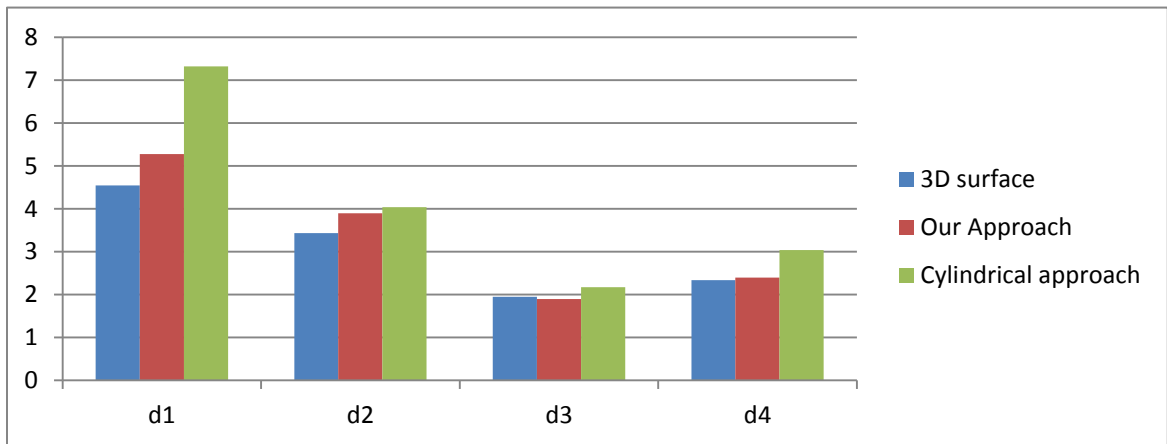


Figure 4.6: Stretch between minutiae points (measured in Euclidean) after unwrapping the fingerprint surface using cylindrical approach and our approach.

In the 2D fingerprint acquisition process, the center region tightly touches the sensor and so has no or very little distortion, whereas due to elasticity of the skin and application of force while pressing the finger, the finger surface is stretched towards the sides and corners [9]. From Table 4.1 and Figure 4.6, we observe that the minutiae distance d_3 and d_4 which signifies the stretch at the center of the fingerprint surface is significantly less with minimal distortion, whereas these distances are stretched in the cylindrical approach. The distance d_1 is significantly larger in cylindrical approach compared to our approach, which imply that the cylindrical approach has large stretch towards the corners and sides. Our approach has preserved this stretch to a larger scale and hence is less error prone during the fingerprint matching process. From the results obtained, we observe that our algorithm is successful in preserving the relative distance between minutiae and hence is very effective for unwrapping a 3D fingerprint surface to its 2D equivalent.

The relative error E in unwrapped distance can then be computed as:

$$E = \frac{|P_m - P_t|}{P_t} \quad (4.1)$$

where P_m is the measured unwrapped distance and P_t is the true 3D fingerprint distance. The relative error also signifies the amount of overall stretch involved during the unwrapping process. For the four minutiae distances considered in Figure 4.5 the relative error computed is 8.72% for unwrapping 3D fingerprint using our method and 30.04% using cylindrical unwrapping method. To have a better estimate of the relative error, we plotted a grid as shown in Figure 4.7, to extract 2209 points of the fingerprint surface.

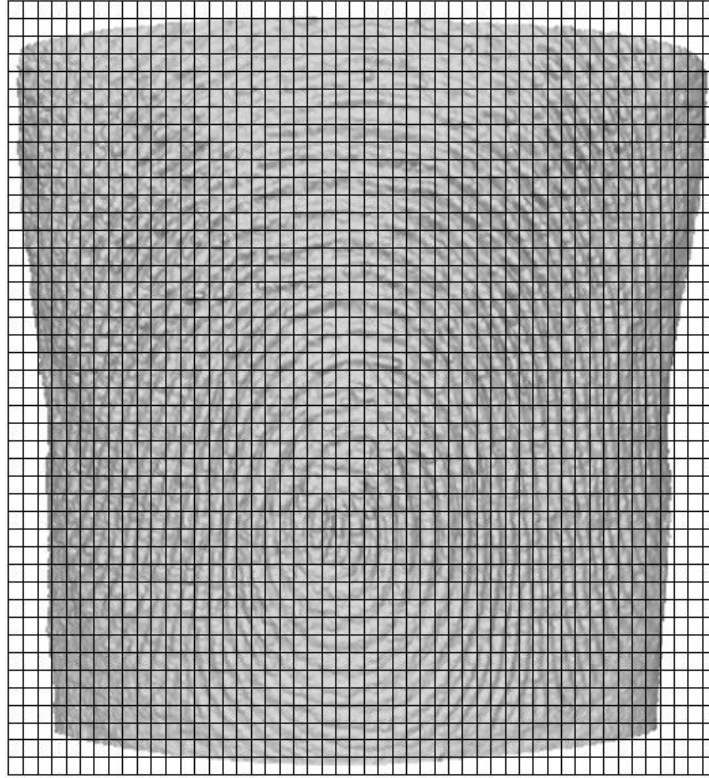


Figure 4.7: Grid created over fingerprint surface to extract fingerprint points

The relative error obtained over these 2209 points is 11.55% using our approach and is 29.89% using cylindrical approach. Comparing the relative error values, we can identify that our algorithm have less relative error compared to cylindrical approach

The algorithm takes 8.2743 sec over Intel Core 2 Duo 2.13 GHz processor to compute the 2D unwrapped image from the 3D scan obtained. The algorithm is part of initial processing of the fingerprint. Once we have the 2D unwrapped fingerprint, the real time 2D fingerprint matching algorithms will be responsible for identification.

CHAPTER 5. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a parametric algorithm based on curvature analysis of the 3D surface to unwrap the 3D fingerprint. We initially extract the ridges and valleys of the 3D fingerprint using the curvature analysis and then convert the 3D surface to the 2D unrolled surface.

We compute principal curvatures from the 3D point cloud dataset to identify the ridges and valleys of the fingerprint. Once features are identified we used the parametric approach to unwrap the fingerprint. This approach projects a 3D fingerprint to the parametric model and then unwraps the model. A convex hull is used as a parametric model and is created on the 3D fingerprint surface to get the best fit plane for texture unwrapping. Results shows that the unwrapped 2D image obtained has minimal distortion and preserves relative distance between minutiae points which is important for the fingerprint matching algorithms.

In our future work we want to compare our results with actual 2D scans of the same person to validate the accuracy of the method. We also want to utilize other methods of curvature analysis of 3D surface to obtain even more accurate results.

REFERENCES

- [1] D. Sarat, J. Anil, Fingerprint Based Recognition, *Technometrics*, Volume 49, Number 3, August 2007, pp. 262-276.
- [2] Y. Chen, Extended Feature Set and Touchless Imaging for Fingerprint Matching, Dissertation, PhD, Michigan State University, 2009. pp. 27-29.
- [3] H. Lin, W. Yifei, and J. Anil, Fingerprint image enhancement: Algorithm and Performance Algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, May 1998. pp. 777–789.
- [4] R. Rowe, S. Corcoran, K. Nixon, and R. Ostrom, Multispectral Imaging for Biometrics. In *Proc. SPIE Conference on Spectral Imaging: Instrumentation, Applications, and Analysis*, volume 5694, March 2005, pp. 90–99.
- [5] Z. Shi and V. Govindaraju, A Chaincode Based Scheme for Fingerprint Feature Extraction. *Pattern Recog. Lett.* 27, 2006, pp. 462-468.
- [6] Y. Hao, T. Tieniu, W. Yunhong, Fingerprint Matching Based on Error Propagation, *International Conference on Image Processing*, Sep. 2002, pp. 273-276.
- [7] J. Anil, S. Prabhakar, L. Hong, and S. Pankanti, FingerCode: A Filterbank for Fingerprint Representation and Matching, *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 2, 1999, pp. 187-193.
- [8] L. Xiping, T. Jie and W. Yan, A Minutia Matching Algorithm in Fingerprint Verification, vol 4, 2000, pp. 833-836.
- [9] K. Rohr, M. Fornefett, and H. S. Stiehl, Approximating Thin-Plate Splines for Elastic Registration: Integration of Landmark Errors and Orientation Attributes. In *Proceedings of the 16th International Conference on Information Processing in Medical Imaging*, vol 1613, 1999, pp. 252-265.

- [10] S. Shafaei, T. Inanc, and L. G. Hassebrook, A New Approach to Unwrap a 3D Fingerprint to a 2D Rolled Equivalent Fingerprint, IEEE Int. Conf. Biometrics Theory, Application and Systems, 2009.
- [11] Y. Chen, G. Parziale, E. Diaz-Santana, and J. Anil, 3D Touchless Fingerprints: Compatibility with Legacy Rolled Images, Biometr. Consort. Conf. Biometr. Symp., 2006, pp. 1–6.
- [12] Yongchang Wang, Daniel L. Lau and Laurence G. Hassebrook. Fit-sphere Unwrapping and Performance Analysis of 3D Fingerprints, Applied Optics, Vol. 49, Issue 4, Feb. 2010, pp. 592-600,
- [13] A. Fatehpuria, D. L. Lau and L. G. Hassebrook, Acquiring a 2-D Rolled Equivalent Fingerprint Image from a Non-Contact 3-D Finger Scan, Biometric Technology for Human Identification, Orlando, Florida, vol. 6202, 2006, pp. 62020C-1 to 62020C-8.
- [14] G. Zigelman, R. Kimmel, and N. Kiryati, Texture Mapping Using Surface Flattening via Multi-dimensional Scaling. IEEE Transactions on Visualization and Computer Graphics 9, no. 2, 2002, pp. 198-207.
- [15] B. O’Neill. Elementary Differential Geometry. Academic Press, Inc., 1966.
- [16] A. Pressley. Elementary Differential Geometry. Springer-Verlag London, 2001.
- [17] Y. Ohtake, Y. Belyaev and S. Seidel, Ridge-valley Lines on Meshes via Implicit Surface Fitting, ACM Transactions on Graphics 23, Aug 2004, pp. 609–612.
- [18] A. Andrew, Another Efficient Algorithm for Convex Hulls in Two Dimensions, Information Processing Letters, 9, 1979, pp. 216-219.
- [19] Barber, C.B., D.P.Dobkin, and H.T.Huhdanpaa, The Quickhull Algorithm for Convex Hulls, ACM Transactions on Mathematical Software, 22, 1996, pp. 469-483.

- [20] P. Yianilos, Data Structures and Algorithms for Nearest Neighbor Search in General Metric Spaces, in Fourth ACM-SIAM Symposium on Discrete Algorithms, 1993, pp. 311-321.
- [21] D. Boyanapally, Merging of Fingerprint Scans Obtained from Multiple Cameras in 3D Fingerprint Scanner System, Master Thesis, University of Kentucky, Feb. 2008.
- [22] Y. Wang, L. G. Hasebrook, and D. L. Lau, Data Acquisition and Processing of 3D Fingerprints. IEEE Transactions on Information Forensics and Security, 5(4), 2010, pp. 750–760.
- [23] C. B. Atkins, J. P. Allebach, and C. A. Bouman, Halftone Postprocessing for Improved Rendition of Highlights and Shadows, J. Elec. Imaging 9, 2000, pp. 151–158.
- [24] K. Hajebi, Y. Abbasi-Yadkori, H. Shahbazi, and H. Zhang, Fast Approximate Nearest-Neighbor Search with k-Nearest Neighbor Graph, 2011, pp. 1312-1317.

