MASTER

3D fingerprint detection in ancient museum sculptures from CT data

Sanders, S.J.C.

Award date:
2019

Link to publication
3D Fingerprint Detection
In Ancient Museum
Sculptures From CT Data

Master Thesis

S.J.C. Sanders

Supervisors:
Prof. Robert van Liere
Dr. Andrei Jalba

Eindhoven, October 2019
Abstract

The identification of a creator of a piece of art has both cultural and economical values. Cultural in a sense that it helps in shaping human history and economical in a sense that it gives an art buyer the confidence in purchasing an authentic art object. The type of art object studied in this thesis are ceramic sculptures. These sculptures are normally highly polished on the outside while the inside is unpolished. Fingerprints, which can be used to identify a single person, are therefore more likely to be found on the inside of a hollow sculpture rather than the outside. Current fingerprint scanning methods need the fingerprint to be located on the outside of the scanned object. This thesis is the first to use computerized tomography (CT) data to obtain fingerprints which can be located on either the outside or inside of an object. It is qualitatively shown that fingerprint identification with CT data is feasible.
Preface

I started my higher education career in Biomedical Engineering. During my studies, it became obvious that programming was an area I highly enjoyed. This resulted in a switch to Computer Science after obtaining my bachelor in Biomedical Engineering. Especially the application of Computer Science of algorithms and systems which solve problems in the real world was an important reason for this switch. Likewise, this is an important factor for my master project being on fingerprint extraction from CT data. The implemented algorithm serves as a first step into fingerprint extraction of fingerprints which are located at places which could not be scanned before.

I want to thank Dr. Andrei Jalba for the detailed discussions we had on the project and particularly on steering me to explore voxel space for a large part of the algorithm. Furthermore, I want to thank Prof. Robert van Liere for steering the project and thesis in the right direction, for the equipment that made me develop the fingerprint extraction algorithm, and for the long and helpful discussions. Lastly, I want to thank my parents for the support and love that helped me finishing this project.
Contents

List of Figures ix
List of Tables xi

1 Introduction 1
  1.1 Fingerprint Identification Pipeline Overview 4
  1.2 Aims of the Project 5
  1.3 Research Questions 5
  1.4 Research Contributions 5

2 Previous Work 7
  2.1 3D Fingerprint Unwrapping 7

3 Background 11
  3.1 Image Processing 11
    3.1.1 Otsu’s Method 11
    3.1.2 Convolution 12
    3.1.3 Gaussian Filter 12
    3.1.4 Median Filter 13
  3.2 Additional Image Processing Methods 14
    3.2.1 Gabor Filter 14
    3.2.2 Coherence-Enhancing Diffusion 15
    3.2.3 Fingerprint filter in Fourier Domain 15
  3.3 Diffusion Tensor Imaging 16
  3.4 Curvature 17

4 3D Fingerprint Unwrapping 19
  4.1 Definitions 19
  4.2 Filters 20
  4.3 Surface Extraction 21
  4.4 Curvature Calculation 21
  4.5 Sphere Unwrapping 24
  4.6 Measure of Distortions 26

5 AFIS 29
  5.1 MINDTCT 29
**CONTENTS**

6 Results ................................................................. 33
   6.1 Samples .......................................................... 33
   6.2 Fingerprint Representation Analysis .......................... 34
   6.3 CT Scans & Algorithm Output .................................. 35
   6.4 Qualitative Identification Analysis ............................ 36
   6.5 Performance & Implementation ................................. 38
   6.6 Automatic Minutiae Extraction with MINDTCT ............... 38

7 Discussion .............................................................. 41

8 Conclusion ............................................................. 43

Bibliography ............................................................. 45

Appendix ................................................................. 49

A Dents detection ....................................................... 49
   A.1 Literature Review .................................................. 49
   A.2 Proposed Method .................................................. 49
## List of Figures

1.1 A subset of fingerprint classes from Henry’s classification system[19] .......................... 2
1.2 Bifurcation and ridge ending minutiae indicated on a fingerprint image. ......................... 3
1.3 Sweat pores visible as holes on top of the ridge lines [49]. ........................................... 3
1.4 An overview of the pipeline to perform fingerprint identification having a CT-scan as input. This scalar volumetric CT-scan is depicted as a blue cube. Each rectangle indicates a stage of the pipeline. The first stage selects subvolumes which are likely to contain fingerprints. The second stage extracts these fingerprints and maps them to 2D images. The third stage consists of an AFIS which will show if two identical fingerprints are found. Only the first two boxes are researched in this thesis. An existing AFIS is used to perform the fingerprint matching stage. .............................. 5

3.1 Application of the Gaussian filter with a different standard deviation. .......................... 13
3.2 Application of the median filter on a gray scale image. Salt and pepper noise is added to the gray scale image and removed with a median filter. ................................. 14
3.3 3 different anisotropic filters applied to a fingerprint from J. Weickert [44]. The top left image contains the original image. The top right image is computed using mean curvature motion. The bottom left image is anisotropic filtering similar to the Cottet-Germain model. The bottom right image is the result of using coherence-enhancing anisotropic diffusion. ................................................. 16
3.4 The osculating circle, i.e. a circle which is touching the line at point \( p \). The circle has a radius of \( R \) and the curvature of \( \psi \). ....................................................... 16

4.1 The 6 steps of the pipeline regarding the creation of a 2D image. From 1 to 6: noise reduction with median and Gaussian filter, surface extraction, computing mean curvature for voxels belonging to the surface, fit sphere on surface voxels, cartesian coordinate system to spherical coordinate system, mapping of volumes to 2D images. ................................................. 19
4.2 Produced by the marching cubes algorithm[29] applied to a scalar field representing a part off a fingerprint region. The threshold value computed with Otsu’s method is used for the iso value of the marching cubes algorithm. ................................. 20
4.3 Produced by the marching cubes algorithm[29] applied to a scalar field representing a part off a fingerprint region. The threshold value computed with Otsu’s method is used for the iso value of the marching cubes algorithm. ................................. 21
4.4 Skeletonization of the negative mean curvature (yellow) displayed on top of a marching cubes obtained isosurface where the iso value is determined with Otsu’s method. ....................................................... 23
4.5 A visual representation of a spherical coordinate system defined in convention ISO 13-11 [10]. ................................................................. 25
4.6 Increasing resolution of the parameter space with domain \( \phi \) and range \( \theta \) by rotating the surface voxels, indicated with small crosses, to one side of the XZ plane. In the left graph, the domain of \( \phi \) is \([0, 2\pi]\) while the domain of \( \phi \) in the left graph is \([0, \pi]\). 25
4.7 A slice of the fingerprint image along \( \theta \) where the distances between two neighboring pixels is measured. Images taken from Wang’s paper [42]. ................................. 27
LIST OF FIGURES

5.1 The different stages used in MINDTCT. Image from the user guide of NBIS [43]. . 30
5.2 Output of MINDTCT where endpoints are indicated by circles and bifurcations
   with squares. Furthermore, the angle the ridge line at the minutiae is given as well.
   Image from the user guide of NBIS [43]. ................................. 31
6.1 An example of a sample that is to be scanned with CT. ......................... 33
6.2 All different depictions of a fingerprint using $\rho$ or specific curvature field. .... 35
6.3 Four different fingerprint samples. Surfaces extracted with marching cubes with an
   isovalue obtained with Otsu’s method. Two pairs of fingerprints belonging to the
   same finger. ................................................................. 37
6.4 Four different fingerprint samples obtained with sphere unwrapping. Two pairs of
   fingerprints belonging to the same finger. ............................. 37
6.5 Zoomed in area of FP1 and FP2. Minutiae found in both fingerprints are marked
   with yellow circles. Furthermore, a red outline is formed by connecting minutiae. . 38
6.6 Zoomed in area of FP3 and FP4. Minutiae found in both fingerprints are marked
   with yellow circles. Furthermore, a red outline is formed by connecting minutiae. . 38
6.7 Output of FpMV (Fingerprint Minutiae Viewer) which utilizes MINDTCT. The red
   circles indicate bifurcations while the green squares indicate ridge endpoints. . . 39
7.1 Examples for the cause of some artifacts which occur when multiple surfaces are
   extracted. The images depict a cross section of the surfaces which are currently
   extracted. The bumpy green surface is the surface containing the fingerprint ridge
   lines. The red surfaces should not be extracted. The black dot indicates the center
   of the sphere. ................................................................. 42
List of Tables

6.1 The performance of the developed algorithm for the four labeled fingerprint samples. The total time includes the median and Gaussian filtering step. .......................... 38
Chapter 1

Introduction

Determining the authenticity of a piece of art can be a daunting but important task. Daunting in the sense that such a task includes many different fields, important in the sense that the historical and cultural values should be truthful and may not be tampered with by economic incentives. Not only wealthy collectors, but also the general population can be harmed by spurious artifacts. For instance, the Dutch government bought the *Washing of Christ’s feet* which was thought to be painted by the famous Dutch artist Vermeer [20]. Though, after spending 1.3 guilders (several million of today’s euros) it was determined that the original painter was Van Meegeren who forged many works of Vermeer. Thus, from both a cultural and economic perspective, it is necessary to unravel the secrets that are seemingly hiding themselves.

Daktyloskopy is a prevailing method of identification enforced by the police in which fingerprints are recorded [24]. The word is a combination of the Greek words daktylos, which means finger, and skopein which means to see or to explore. While daktyloskopy has been proven useful by the police, the human fingerprint is rarely used in archaeological research even though there are a plethora of mediums that do contain them. The usefulness of fingerprints is undervalued due to the fact that they seem to occur randomly and have no value in ‘the world of statistical testing’. The real reason for the few recorded fingerprints on old artifacts is mainly caused by a lack of awareness.

Fingerprints have been found on various mediums [24]. The most prominent medium for finger imprints on old artifacts are ceramics. Fingerprints are naturally added to the clay during the formation of the ceramic artifact. The plasticity at the molding stage combined with the rigid state that is chemically stable after drying the clay results into imprints that are long lasting. While chemically stable, ceramics can still be considered a fragile medium at their final stage. However, fingerprints have been found on artifacts which are as old as 25 000 years (Králíc et al. 2002). The abundance of fingerprints on ceramics in combination with the minimal usage of them in archaeology indicates room for exploration.

Methods to extract fingerprints belonging to the outside of a structure do exist [25]. Fingerprints on the inside of hollow sculptures cannot be gathered in the same manner since these areas are hard to reach and the sculpture may not be broken. Thus, a scanning method is needed in which the whole volume is recorded. These methods can be found in the medical world. Tomography is an imaging modality in which projections of a volume are recorded by penetrating an object with a wave. In particular, computed tomography (CT) uses radioactive waves that penetrate the object at different angles. A whole volume can then be computed by combining multiple projections. Thus, CT-scans can be used to capture both the outside and inside of an object allowing for extraction of fingerprints inside an object.

This thesis will focus on a pipeline which performs fingerprint feature line extraction on volumetric CT-scans of ceramic sculptures, in particular art objects, with the goal to identify its creator. In order to get an understanding on the exact features of the fingerprint that are to be extracted, a brief definition is given, and different classification schemes that are applied on different scales are discussed. Fingers, palms, soles and toes are creased with parallel running
CHAPTER 1. INTRODUCTION

ridges [24]. Deeper in the skin, the epidermal lines of the fingerprint are underset by two rows of papillae of corium. Therefore, these lines are often called papillary lines which together form the papillary terrain. In this thesis, the papillary lines will be referred to as ridge lines. These ridge lines are thought to have functions like the amplification of tactile information [37], or as a means to improve friction between the hand and other objects [47]. There are three types of classification methods which act on different scales and can be used to identify or classify a fingerprint [49]:

1. **Henry’s classification system**[19]: The most common classification system relies on the global structure of the fingerprint and was devised by Henry in 1900. Examples of these structures are loops, whorls and arcs as is shown in Figure 1.1. The different categories were specifically useful for indexing large fingerprint libraries such that the fingerprint identification process would be less time consuming.

2. **Minutiae**: Small structures that pertain to a single ridge are the second type of structures. These small characteristics, also called minutiae, can be used to identify a single person. The two basic examples are bifurcations, where one ridge splits up in two, and ridge endings. These are depicted in Figure 1.2. All other minutiae are based on a combination of these two [49].

3. **Sweat pores**: The third type of structure consists of small features such as sweat pores, as shown in Figure 1.3. This type of structure is hardly used for automated systems since they are sensitive to noise and require high quality scans.

![Figure 1.1: A subset of fingerprint classes from Henry’s classification system[19]](image)

Henry’s classification system is an excellent tool to classify fingerprints into a few categories. However, it is not suitable for identification of individuals. On the other hand, identification of a person can be performed using ridge lines and sweat pores. With regards to matching based on minutiae, it was mathematically determined that the probability that a fingerprint with 36 minutiae will share 12 minutiae with another arbitrary fingerprint having 36 minutiae is $6.10 \times 10^{-8}$ [33]. For intra-ridge sweat pores, the probability of occurrence of a particular combination of 20 ridge-independent pores is $5.186 \times 10^{-8}$ [35]. While sweat pores can be used to perform fingerprint identification, they are not suitable for fingerprints on ceramics. They are hard to find and the visibility is dependent on the grain size of the used material. Furthermore, many automated fingerprint identification systems (AFIS) are developed with minutiae extraction in mind [14, 48, 4, 21]. This facilitates testing of the pipeline with these already established systems. Since this thesis is concerned with the extraction of the fingerprint, and not on minutiae detection for fingerprint identification, the focus is on extracting ridge lines while a generally available AFIS can be used for testing. Furthermore, even the relatively large ridge lines still require high resolution CT-scans, a requirement which will become clear in the next paragraph.

To compute the size needed to capture the fingerprint and its minutiae, the size of each ridge line needs to be known. The ridge lines were found to have a height of 59.0 $\pm$ 19.2 $\mu m$, a width of 435.5 $\pm$ 57.4 $\mu m$ and are spaced 484.9 $\pm$ 70.6 $\mu m$ [39]. In order to capture each ridge line, it
is needed to use a CT-scan with a spatial resolution which captures more detail than the smallest property of the fingerprint. Thus, a scan should on average have a spatial resolution around 50 µm in each axis in order to capture the height of the ridge lines and therefore the fingerprint itself. With the following back-of-the-envelope calculation, it is possible to estimate the amount of space required to store the detailed information of a fingerprint. Assume that each voxel is stored as an 8-bit value and that the fingerprint is located in a 27 cm³ volume where each side is 3 cm long. Then each axis contains 600 voxels which translates to a CT-scan of 216 MB. However, the pipeline is to be used to capture fingerprints in large sculptures which can be 1000 times as large (or more) as the small example. It is not feasible to extract the fingerprints of such large datasets since operations on the data will be both time consuming and require either an astronomically amount of RAM or I/O instructions. Thus, for the developed pipeline, there is opted for a hierarchical method in which a low resolution scan of the complete sculpture is used to indicate regions of interests. These regions of interests may then be scanned in a higher resolution such that the fingerprint is captured by the scan.

When acquiring a fingerprint, it is shown that a detailed CT-scan is needed. This introduces the problem that the effect of noise will be more noticeable. Not only random noise, but also other by-products or artifacts are bound to happen\[11\]. These artifacts are caused by the non-perfect nature of CT-scanners and the discretization of the scanned object. Namely, it is not possible to have an infinite detector resolution, perfect detectors, no scatter of the photons etc. Techniques do exist to reduce these artifacts to a certain extent. For instance, iterative reconstruction is a scanning method in which data is combined from multiple scans. This results in a reduction of noise, radiation dosis, and the method allows for high resolution scans. For the developed pipeline, only standard imaging filters (e.g. Gaussian filter) are used to combat the presence noise.

Artifacts in art objects caused by CT-scans are not the only obstacles when acquiring high quality fingerprint images. The fingerprints on the art object itself might be in an abysmal state.
CHAPTER 1. INTRODUCTION

There are a multitude of factors which can degrade the quality of the fingerprint. Distortions to the fingerprints will occur during the highly plastic phase of clay. All finger imprints are already deformed during its formation due to pressure. In contrast, this is not the case when creating a cast of the finger. Additional forces will distort the fingerprint while the art object is being shaped. Thus, the plasticity of the ceramic clay has both a positive and a negative side. Not only the plastic phase, but also the drying and burning process has effect on the imprint. These processes will shrink the clay, and thus the fingerprint. Furthermore, incorrect burning of ceramics can lead to abrasion of the surface layers.

Another problem in acquiring detailed fingerprint lies in the fact that fingerprints are normally not left behind intentionally. The intent is not to capture the fingerprint but to shape the object. This has multiple consequences to the quality of the fingerprint. The applied force might be too small to leave a detailed fingerprint behind. Multiple fingerprints which are found in the same area can be intertwined with each other, resulting in imprints with lost minutiae and fingerprints which are hard to distinguish from each other. A third problem lies in partial fingerprints. Like the first two problems, this decreases the probability of shared minutiae between multiple fingerprints. However, methods do exist to tackle the partial fingerprint problem. AFIS have been created which takes these partial fingerprints in mind [22]. Furthermore, methods do exist which combine these partial fingerprints in a complete one [36].

The developed pipeline will completely focus on the extraction of fingerprints without being concerned about CT-scanner artifacts and partial fingerprints. The pipeline will incorporate an open source minutiae extraction algorithm to test if automatic identification is possible. Since the fingerprint will be distorted by the many effects that have been discussed, it is expected that an AFIS will not be able to accurately match two fingerprints. Therefore, a detailed qualitative analysis will be performed as well to assess the quality of the extraction.

1.1 Fingerprint Identification Pipeline Overview

In this section, the proposed pipeline for fingerprint identification is introduced. The pipeline consists of three stages which are depicted in Figure 1.4 as black bordered rectangles. The input consists of a low spatial resolution (100µm) volumetric CT-scan of the complete art object and is indicated as a blue cube. The subvolume selection stage selects dents in the surface of the art object in which a possible fingerprint does exist. The locations which are found to have a fingerprint are scanned at a higher spatial resolution (50µm). This is the output of the first stage. The Subvolume to 2D fingerprint image stage applies sphere unwrapping to obtain 2D images of the fingerprints found in the subvolumes. Finally, an AFIS is used to find a match between any of the found fingerprints.

Each stage will will be described in a separate chapter. The focus of this thesis is on the second stage of the pipeline. A detailed approach to tackle this stage will be discussed in Chapter 4. Tools are explored in Chapter 5 which can be used to extract minutiae from 2D images and perform automatic fingerprint identification. A devised method which performs automatic detection of dents on a surface, in order to capture regions with possible fingerprints, is briefly discussed in Appendix A.
Figure 1.4: An overview of the pipeline to perform fingerprint identification having a CT-scan as input. This scalar volumetric CT-scan is depicted as a blue cube. Each rectangle indicates a stage of the pipeline. The first stage selects subvolumes which are likely to contain fingerprints. The second stage extracts these fingerprints and maps them to 2D images. The third stage consists of an AFIS which will show if two identical fingerprints are found. Only the first two boxes are researched in this thesis. An existing AFIS is used to perform the fingerprint matching stage.

1.2 Aims of the Project

This project has the following aims:

1. To provide a pipeline which automatically extracts and identifies fingerprints found on ceramics.

2. To implement and test a method to extract fingerprints from scalar volumetric data (in particular CT-scans).

1.3 Research Questions

The research questions answered in this thesis are as follows:

1. What are the different approaches through which fingerprints can be extracted from volumetric data?

2. How do the different fingerprint extraction algorithms compare to each other?

3. How well can the fingerprints, which are extracted by the developed method in this thesis, be distinguished from each other?

4. What is the performance in time of the developed method?

The measurement on how well different fingerprints can be compared with each other will be performed qualitatively, through manually finding corresponding minutiae, and quantitatively through automated qualitative tools and an AFIS.

1.4 Research Contributions

The following research contributions are made:

1. Fingerprint extraction from scalar volumetric data (CT-scans).

2. Improved definitions and small changes to Wang’s sphere unwrapping algorithm.

3. Computation of curvature properties on the surface in voxel space in contrast with related work that obtains these values from a point cloud.
4. An investigation of different curvature values for capturing ridges of fingerprints starting from an implicit fingerprint surface representation.

Where sphere unwrapping is a method to map 3D scalar data to a 2D image. This will be explained in Chapter 2.
Chapter 2

Previous Work

In this chapter, relevant articles regarding fingerprint extraction from 3D data which are mapped to 2D images are explored. While none of the upcoming articles use CT-scans as their input, many methods can be applied to our specific application. Furthermore, only articles related to 3D fingerprint unwrapping algorithms are considered while methods do exist which obtain the 3D fingerprint itself. There is chosen to focus on an unwrapping algorithm since the 2D fingerprint images are both easier to analyze and are compatible with currently available AFIS.

2.1 3D Fingerprint Unwrapping

A nonparametric unwrapping method which resembles the effect of virtually rolling a 3D finger on a 2D plane was devised in 2006 by Chen et al.\[12\]. The paper discusses two types of unwrapping: parametric and non-parametric unwrapping. Parametric unwrapping refers to the projection of a 3D object onto some mathematical shape (cylinder/sphere). Non-parametric unwrapping involves no projection step onto a parametric model and is instead directly applied on the object itself by preserving a specific spatial property (angular/distance relationships). It is noted that parametric unwrapping is efficient and straightforward, but will fall short in preserving relative distances. Since many AFIS use the distances between minutiae to perform fingerprint matching, the proposed non-parametric unwrapping method gives a higher matching accuracy than parametric unwrapping methods. The proposed method divides a fingerprint into thin slices and locally unfolds these slices such that the inter-point surface distances and scale are preserved. Slices are created with interpolation such that all slices are equally spread out having a distance of $h$ between each other. Furthermore, all points in the slice are equally spaced with the same $h$. The equal spacing preserves the distance in each direction and thus the scale likewise.

In 2006, Fatehpuria et al.\[15\] developed a novel method to iteratively flatten the 3D fingerprint into a 2D plane. In order to capture the surface of a given point cloud, the algorithm first obtains a rough shape of the fingerprint without including detailed valley-ridge lines. This is performed with a weighted non-linear least square algorithm where the weight is calculated with a Gaussian function. The resulting image is obtained by computing the difference between the smoothed fingerprint surface with the scanned points. Flattening of the fingerprint is performed with the "Springs method" proposed by Atkins et al. which has originally been used to process digital halftoning [9]. A rectangular mesh is created using a subset of the point cloud where each edge corresponds to a spring. The energy of each spring can be calculated by an energy function which takes the preferred distance in a 2D plane and the current relaxed distance of the spring between two endpoints. Iteratively, a small amount of the stored energy is used to displace the mesh into a 2D plane. This resembles a rolled equivalent of the 3D scan. The smoothed fingerprint surface is then warped onto the resulting points such to obtain a 2D image by taking the difference between this surface and the displaced scanned points.

Shafaei et al.\[38\] devised in 2009 an additional step to the method of Fatehpuria et al.\[15\].
CHAPTER 2. PREVIOUS WORK

Instead of using the difference between the smooth surface of the fingerprint with the point cloud, curvature analysis was performed to obtain the fingerprint texture. Both the Gaussian and mean curvature are suggested to be used to assess if a point belongs to a ridge. The constructed algorithm only utilizes the mean curvature such that every point is colored black in the final image when the mean curvature is positive.

Wang et al. created in 2010 a parametric sphere unwrapping algorithm which incorporated a method to reduce mapping distortions [42]. A sphere is fitted through all points with a least squares fitting algorithm. Then, a linear mapping is created such that an angle corresponds to a pixel value in the non distortion corrected image. A distance metric is used which incorporates the angles between two consecutive points and the distance from the center of the sphere to these points. This metric is utilized to create a non-linear mapping in which distortions in distance are reduced. Quality has been assessed with NIST Fingerprint Image Software (NFIS). It is noted that 3D unwrapped fingerprints have a higher quality compared to 2D ink rolled fingerprints in local quality and minutiae detection. Furthermore, it was shown that quality is improved compared to 3D unwrapped fingerprint using the Springs method.

Wang published that same year another article in which an improvement was made to the parametric approach of fingerprint unwrapping with a cylinder [41]. This new method combined the sliced based approach found in the paper by Chen et al. with the parametric approach of using a cylinder to perform unwrapping. Instead of using a cylinder with a fixed radius, the cylinder was broken down in many circles having different radii. This more accurately approximates the shape of a finger.

In 2013, Pang et al. [32] used a point cloud generated by a photometric stereo 3D reconstruction system to fit a local paraboloid surface. At each sampled point and with the use of the fitted surface, the principal curvatures (both the maximum and minimum curvature through that point) are computed. The principle curvatures correspond to the eigenvalues of the hessian of the local fitted paraboloid surface. The biggest absolute principal curvatures are used to detect if a point either lies on a valley or a ridge. After removing the outliers using both covariance analysis and cross-correlation coefficient estimation, polylines are generated along the valleys and ridges which form together a fingerprint. While an exact time complexity analysis is not performed by the authors, the performance with regard to running time and memory usage is given. Each 3D fingerprint of roughly 50K points is computed in a couple of seconds while the memory usage is around 50 Mb. The authors argue that optimization was not the primary focus of their current work and feel like improvements may result in the algorithm to be used in real-time embedded systems. This technique is based on a point cloud of the surface of the finger. Therefore it is applicable in combination with a method to extract the isosurface of the finger.

Conclusion

The papers can be divided in two categories: the parametric and non-parametric approach. Non-parametric approaches are said to be difficult to implement but relative distances or angles between minutiae can be preserved according to Chen et al.. Since distances between minutiae can be used to compare different fingerprint with each other, non-parametric approaches seems to be a preferred method of choice. Due to lack of information in the paper by Chen et al., it is not explained how a pixel value in the final image is related to a sampled point. In combination with other papers, two possible solutions could be used. The paper by Fatehpuria et al. indicates that a pixel value could be computed by obtaining a rough shape of the fingerprint which does not contain any ridges or valleys. A pixel value will then be the euclidean difference between a sampled point and the rough shape. The second approach involves the computation of curvature values at each sampled point as has been performed by Shafaei et al..

The conventional parametric approach which is described by Chen et al. does not address the preservation of distances and angles. However, Wang et al. did show that distance preservation can be obtained with the parametric approach. Furthermore, it was shown that this parametric approach performs better than the non-parametric springs method. The relatively bad perform-
formance of the springs method seems to be caused by the uncontrollable nature of the flattening process. Furthermore, this method iteratively displaces points on a mesh which can result in an unstable system when these steps become too large.

In this thesis, it is chosen to utilize sphere unwrapping for obtaining a 2D fingerprint image. This method has the advantage that the implementation is relatively straightforward while additional measures against distortions can be implemented as an additional step in the process. Rather than recording the distance between the center of the sphere and a point of the fingerprint, as has been conducted by Wang et al., curvature values will be computed in the volume itself from implicit surfaces to obtain the final 2D image. This is different from the papers of Shafaei and Pang where different curvature values are computed on a point cloud by obtaining an explicit mesh representation of the surface. Implicit methods for determining curvatures are more accurate and less sensitive to surface discretization errors than explicit methods and are thus preferable to use when scalar volumetric data is available [3]. This is the case since explicit methods fail in measuring accurate curvatures on irregular meshes. Surface extraction with marching cubes, which is the method of choice for surface extraction, will usually result in irregular meshes.
Chapter 3

Background

This chapter will introduce some general techniques which are used in the various parts of this thesis. Some methods are generally useful for enhancing images while others are more tailored for enhancing fingerprint images.

3.1 Image Processing

Image processing techniques can enhance images by e.g. removing noise and enhancing edges. With respect to 3D fingerprint extraction, these methods can drastically improve the the quality of the final extracted 2D image. This results in minutiae which are more perceptible to both the human eye and automated systems. All image processing methods in this section are used in this thesis.

3.1.1 Otsu’s Method

Nobuyuki Otsu devised in 1977 a nonparametric and unsupervised method to acquire a threshold value for gray scale image segmentation \[31\]. In our project, Otsu’s method is crucial for performing surface extraction. This method is generally used for segmenting two classes in an image but can be extended for multiple classes. For this project, it is assumed that the CT-scan only contains two homogeneous classes. Each voxel belongs either to air or the scanned object. These classified voxels can then be used to mark the to be extracted surface. The environment will now be created to explain the main concepts of Otsu’s method. Let a picture have \(L\) gray levels be thresholded by some \(k\). Each gray level has a probability \(p_i = \frac{n_i}{N}\) to occur at a randomly selected pixel, which is the number of pixels at level \(i\) normalized by the number of pixels \(N\) in the whole image. Then the probability of class occurrence is as follows,

\[
\omega_0 = Pr(C_0) = \sum_{i=1}^{k} p_i \quad (3.1)
\]

\[
\omega_1 = Pr(C_1) = \sum_{i=k+1}^{L} p_i \quad (3.2)
\]

The following three equations represent some important characteristics of a two segmented image

\[
\sigma_W^2 = \omega_0\sigma_0^2 + \omega_1\sigma_1^2, \quad (3.3)
\]

\[
\sigma_B^2 = \omega_0(\mu_0 - \mu_T)^2 + \omega_1(\mu_1 - \mu_T)^2, \quad (3.4)
\]
\[ \sigma^2_T = \sum_{i=1}^{L} (i - \mu_T)^2 p_i, \]  
\[ (3.5) \]

where \( \sigma^2_W \) is the within-class variance, \( \sigma^2_B \) is the between-class variance, \( \sigma^2_T \) is the total variance of levels, \( \sigma^2_0 \) and \( \sigma^2_1 \) are the variance of class 0 and 1 respectively, \( \mu_0 \) and \( \mu_1 \) are the mean of class 0 and 1 respectively, and \( \mu_T \) is the total mean of all levels in the image. Note that both \( \sigma^2_W \) and \( \sigma^2_B \) are dependent on thresholding level \( k \) while \( \sigma^2_T \) is not. The following equation does always hold,

\[ \sigma^2_W + \sigma^2_B = \sigma^2_T. \]  
\[ (3.6) \]

A well segmented image should have a low within-class variance since this translates to a coherent class. Furthermore, the between-class variance should be high for a well segmented images since this is an indication on how different the two classes are. Since \( \sigma^2_T \) is constant and Equation 3.6 holds, a well segmented image can be obtained by either minimizing \( \sigma^2_W \) or maximizing \( \sigma^2_B \). Thus an algorithm can select threshold value \( k \), by looping through all thresholding levels \( L \), where \( \sigma^2_W \) is the lowest.

### 3.1.2 Convolution

One important concept which is at the heart of many imaging methods is convolution [30]. Convolution between signal functions \( p_1(t) \) and \( p_2(t) \) is denoted by \( \ast \) and is defined as follows,

\[ (p_1 \ast p_2)(t) = \int_{-\infty}^{\infty} p_1(\tau)p_2(t - \tau)d\tau \]  
\[ (3.7) \]

This is the basis of system theory and the formula corresponds to a system with a stimulus \( p_1(t) \) and a system’s response \( p_2(t) \). Function \( p_2 \) is shifted by \( t \) and the time axis is inverted. This can be viewed as a memory function. Convolution sums the effect of a stimulus multiplied by the memory function where the output of the system is the cumulative response to a stimulus. In our context, we are working in a discretized world of 3D and 2D images. Convolution is performed by group operations to compute new pixel values from a pixels’ neighborhood. Group operations are usually expressed by template convolution, where the template (or kernel) is a set of weighting coefficients. For instance, a \( 3 \times 3 \) kernel has a height of 3 and a width of 3 pixels. New pixel values are computed by placing the kernel on a pixel of interest in the input image. Pixel values in the image are multiplied by their corresponding weighting coefficient and added to a total sum. Usually, the sum evaluates a new value for the center pixel (of the template) and this becomes the pixel in a new output image. A new image is obtained by applying the kernel on all pixels of the input image. Since the kernel cannot be applied to the pixels at the border of the image, a choice must be made:

1. Set the border to black
2. Assume that the image is replicated to infinity in all dimensions.
3. Decrease the size of the output image

The current explanation was specifically on 2D images but the same concepts can be applied in a similar fashion in 3D.

### 3.1.3 Gaussian Filter

The most notable image filter is based on the Gaussian distribution. A 2D Gaussian function \( G \) is defined as follows,

\[ G(x, y) = \frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}}, \]  
\[ (3.8) \]
CHAPTER 3. BACKGROUND

where \( \sigma^2 \) is the variance. The Gaussian function is essentially removing the influence of points that are greater than \( 3\sigma \) in radial distance from the center of the kernel. A larger variance will result in the removal of noise at the expense of losing features. A example of a 5x5 discrete approximation for a Gaussian kernel with \( \sigma = 1 \) is as follows

\[
\begin{bmatrix}
1 & 4 & 7 & 4 & 1 \\
4 & 16 & 26 & 16 & 4 \\
7 & 26 & 41 & 26 & 7 \\
4 & 16 & 26 & 16 & 4 \\
1 & 4 & 7 & 4 & 1
\end{bmatrix}
\]

(3.9)

An example of the Gaussian filter is shown in Figure 3.1. Figure 3.1 shows the application of the Gaussian filter with a different standard deviation. It is clear that a larger \( \sigma \) leads to a reduction in detail.

![Figure 3.1: Application of the Gaussian filter with a different standard deviation.](image)

3.1.4 Median Filter

A filter that is useful for this project is an edge preserving filter such that the information of the fingerprint is not lost. The Median filter is edge preserving and is an excellent tool for removing speckle and salt-and-pepper noise [7]. Instead of computing a weighted average using a kernel, this filter computes the median value in a certain window. Thus, it can be considered as a tool to remove local extremes that have a smaller size than such a window. Salt-and-pepper noise are local extremes which are visible as white and black spots in a gray scale image. These spots are thus logically removed by the median filter. An application of this filter is shown on a gray scaled version of our example image in Figure 3.2.
3.2 Additional Image Processing Methods

The discussed image processing methods in this section have not been used in this thesis but were considered to be useful in the context of this project.

3.2.1 Gabor Filter

This type of filters are based on the work of D. Gabor on communication theory[17]. Gabor recognized the fact that in our everyday experiences, both time and frequency play a role in communication. So called information diagrams were defined to be a 2 dimensional plot of both time and frequency in a limited time and frequency window. The smallest possible areas in this diagram are the so called elementary signals and are harmonic oscillations modulated by a probability pulse like the Gaussian function. A suggested usage of these notions lie in data reduction by compressing before transmission and decompressing when receiving audio data. These definitions apply on a 1D signal and were expanded to 2D by J.G. Daugman [13]. This paper focused on modeling the visual cortical neurons which plays a role in vision of mammals. Not only the frequency but also the rotation of the wave had to be captured in this new definition. The same notions of an elementary signal that can be captured in a 2D information diagram can be applied in this new definition. Now, the filter is located in a 4D information hyperspace where the orthogonal axes are x, y, u and v (2D space and 2D frequency coordinates). A 2D image can be captured efficiently by decomposing it in the elementary signals. This property is not only useful for efficiently storing images but also for image analysis. The family of Gabor filter functions in 2D, where \( f(x,y) \) centered on \((x_0, y_0)\) is transformed with \( F(u,v)\) centered at \((u_0, v_0)\) is defined as follows

\[
f(x,y) = \exp\left\{-\pi[(x - x_0)^2 a^2 + (y - y_0)^2 b^2]\right\}
\times \exp\left\{-2\pi i[u_0(x - x_0) + v_0(y - y_0)]\right\}
\]

(3.10)

which is the product of an elliptical Gaussian with an aspect ratio of \(b/a\) and a center located at \((x_0, y_0)\) times a complex exponential representing harmonic modulation. The corresponding Fourier transform is as follows
\[ f(x, y) = \exp\left\{-\pi\left[\frac{(u - u_0)^2}{a^2} + \frac{(v - v_0)^2}{b^2}\right]\right\} \times \exp\left\{-2\pi i[x_0(u - u_0) + y_0(v - v_0)]\right\}. \] (3.11)

Jain et al. used a bank of gabor filters to divide the content of patterns in the image based on the idea that Gabor filters mimick vision cells in mammals [16]. The properties which distinguish these patterns are the spacial frequency, density of the elements, orientation, phase and energy. The constructed algorithm could distinguish these properties but failed to predict human performance with patterns having differences in densities. Nevertheless, these findings lead to multiple methods to both enhance fingerprint images and to perform feature extraction.

### 3.2.2 Coherence-Enhancing Diffusion

Coherence-enhancing diffusion (CED) filtering [44] is a type of filtering which is specifically made to enhance flow lines. It is based on the class of nonlinear diffusion filters which have been defined by Perona and Malik [34]. The diffusion equation for anisotropic diffusion is as follows

\[ \frac{\partial u}{\partial t} = \nabla \cdot (D \nabla u), \] (3.12)

where \( u \) is the concentration at a specific location, \( t \) is the time and \( D \) is a positive definite symmetric diffusion tensor which indicates the degree of diffusion at a certain direction. The standard isotropic version of this formula contains a scalar diffusion coefficient instead of a diffusion tensor. This tensor can be computed by the eigenvalues of the so-called structure tensor \( J_\rho(\nabla u_\sigma) \) which is also known as the interest operator or second-moment matrix.

\[ J_\rho(\nabla u_\sigma) = G_\rho * (\nabla u_\sigma \nabla u_\sigma^T). \] (3.13)

where \( G_\rho \) is a Gaussian kernel with standard deviation \( \rho \) and \( u_\sigma \) is defined as concentration at a particular location, and \( * \) denotes convolution. The eigenvectors of \( J_\rho(\nabla u_\sigma) \) indicate the direction of flow while the eigenvalues correspond to the strength of the flow. The value of \( \rho \) should reflect the characteristic size of the texture and is normally considerably larger than the noise scale \( \sigma \). This filtering method does not suffer from washing out parallel structures in an image. Using direct gray-scale image enhancement in comparison with approaches that require binarization and thinning as intermediate steps, is shown to be more effective [18].

The diffusion equation can be solved numerically with a finite difference method like the different Euler’s methods or Runge-Kutta. J. Weickert et al. [45] developed a scheme which is both faster to compute, due to its stability, and results in crisper and more accurate images. Images are considered to be more accurate since the scheme is designed to be strongly rotational invariant. Conventional finite difference methods like the Forward Euler can be performed by two convolutions with a 3x3 mask (in 2D). This corresponds to convolution over a 5x5 mask and is being used as a basis for this new scheme.

### 3.2.3 Fingerprint filter in Fourier Domain

The frequency in how the ridge lines of a fingerprint are separated suggests that the Fourier domain can be considered a helpful tool. It was found that this domain was useful for the design of filters which remove pseudo-minutiae [23]. Both a directional and frequency filter were used in combination with the energy minimization principle such to smooth local areas. The filters were able to decrease the error rate by about two-thirds compared to Asai’s method when using the standard NIST fingerprint data set.
CHAPTER 3. BACKGROUND

Figure 3.3: 3 different anisotropic filters applied to a fingerprint from J. Weickert [14]. The top left image contains the original image. The top right image is computed using mean curvature motion. The bottom left image is anisotropic filtering similar to the Cottet-Germain model. The bottom right image is the result of using coherence-enhancing anisotropic diffusion.

3.3 Diffusion Tensor Imaging

A this method is directly related to the usage of applying diffusion to enhance the flow lines in images. Diffusion tensor imaging (DTI) is a technique which is highly popular in brain research [8]. It is used to map the white matter fibers which have been captured with Magnetic Resonance Imaging (MRI). Individual strands are created based on diffusion tensors. By reapplying the same technique to fingerprint extraction, it might be possible to construct the ridges of the finger from the 3D image itself. To our knowledge, the application of DTI to fingerprint analysis has not been researched before.

Figure 3.4: The osculating circle, i.e. a circle which is touching the line at point \( p \). The circle has a radius of \( R \) and the curvature of \( \psi \).
CHAPTER 3. BACKGROUND

3.4 Curvature

The curvature of a surface can be used to capture the ridge lines or valleys of a fingerprint. It is therefore useful to obtain some knowledge on this topic. The book by John M. Lee which is on *Riemannian Manifolds: An Introduction to Curvature* is used to summarize some key definitions and principles used in this thesis[26]. First the curvature is defined in 2D. In Figure 3.4, an osculating circle is displayed which is a circle tangent to point $p$ which has the same solution for the second derivative. Then, the curvature at point $p$ is defined as $\frac{1}{R} = \psi$, where $R$ is the radius of the osculating circle. Thus, a small radius corresponds to a large curvature. It is useful to extend the definition such that the curvature can take both positive and negative values. This is done by choosing a normal vector field $N$ along the curve. A positive curvature then corresponds to circle being on one side and a negative on the other side.

Different curvatures can be computed on a surface using the definitions of the 2D case. For a point $p$ on surface $S$, choose a plane $\Pi$ which is contained in the normal vector of $p$. The intersection of $S$ with $\Pi$ is a plane curve. The maximum and minimum signed curvatures $\kappa_{\text{max}}$ and $\kappa_{\text{min}}$, or principal curvatures, are the maximum and minimum signed curvatures for all planes which are contained in the normal vector of $p$. The principal curvatures will be used in Chapter 4 and the computation of these values in a scalar volume will be explained in detail.
Chapter 4

3D Fingerprint Unwrapping

The second stage of the proposed pipeline involves unwrapping of a 3D fingerprint to a 2D image, as is shown in Figure 1.4. Figure 4.1 is a diagram explaining the 6 key steps which belongs to the stage to perform 3D fingerprint unwrapping. As the input for the algorithm, volumes containing exactly one fingerprint are used. The first step is on the application of imaging filters to reduce noise and other artifacts to improve the final quality of the unwrapped fingerprint. Both the Gaussian and median filter and their meaning in this pipeline will be discussed in section 4.2. Step 2 depicts an application of Otsu’s method, as explained in section 3.1.1, to perform surface extraction. In step 3, the curvature is computed for each voxel in the volume that belongs to the surface. The last three steps are part of the spherical unwrapping approach. Each step in the fingerprint unwrapping stage has its corresponding section while the last three steps are explained in one coherent section.

![Figure 4.1: The 6 steps of the pipeline regarding the creation of a 2D image. From 1 to 6: noise reduction with median and Gaussian filter, surface extraction, computing mean curvature for voxels belonging to the surface, fit sphere on surface voxels, cartesian coordinate system to spherical coordinate system, mapping of volumes to 2D images.](image)

4.1 Definitions

This section will give some short definitions which are used throughout this chapter. Let \( V = \{v_0, v_1, ..., v_N\} \) be the set of all \( N \) voxels contained in one CT-scan. Each voxel corresponds to a vector which indicates its location in the scan.
with $0 \leq i < N$ and $v_i^T = x$ etc. Each voxel does hold a number of properties which is denoted as a mapping of the voxel vector to a certain property,

\[
\begin{align*}
\gamma(v_i) &:= \text{function defining if } v_i \text{ belongs to either "air" or an "object} \\
\kappa_m(v_i) &:= \text{mean curvature at voxel } v_i \\
\rho(v_i) &:= \text{distance from the center of the sphere to voxel } v_i \\
\omega(v_i) &:= \text{measure of importance of voxel } v_i
\end{align*}
\]

where the mean curvature will be computed using the methods in section 4.4, a sphere will be fit on all the surface voxels in section 4.5 and $\rho$ is used for sphere unwrapping.

### 4.2 Filters

After obtaining a volume containing a single fingerprint, two filters are applied. First, a median filter is used to reduce artifacts of the scan. The effect of this filter is shown in Figure 4.2. It is clear that this filter is an essential tool for obtaining a surface which is less noisy. The window which is used for this filter needs to be small enough such that the ridges of the finger are not lost in the process. Furthermore, a large window leads to a blocky surface which is not favorable for computing the mean curvature at a later stage of the pipeline. In this particular example which has a size of $1944 \times 1944 \times 1536$, a window with a size of 3 in each dimension has been applied.

![Without median filter](image1)

![Median filtered](image2)

(a) Without median filter  
(b) Median filtered

Figure 4.2: Produced by the marching cubes algorithm applied to a scalar field representing a part of a fingerprint region. The threshold value computed with Otsu’s method is used for the iso value of the marching cubes algorithm.

Further irregularities can be reduced by applying a Gaussian with a small standard deviation as can be seen in Figure 4.3. The result of this step is a relatively smooth surface which mainly contains the information of the ridge lines.
CHAPTER 4. 3D FINGERPRINT UNWRAPPING

(a) Median filtered  (b) Median filter and Gaussian filtered

Figure 4.3: Produced by the marching cubes algorithm\cite{29} applied to a scalar field representing a part off a fingerprint region. The threshold value computed with Otsu’s method is used for the iso value of the marching cubes algorithm.

4.3 Surface Extraction

After the filters are applied, surface extraction can be performed. In this stage, all voxels that lie on the surface will be selected. It is needed to determine if a voxel belongs to the scanned object or the air before the surface voxels can be selected. Otsu’s method can be used to make this distinction by minimizing intra-class intensity variance while maximizing inter-class variance as is explained in Chapter 3.1.1. Using this classification method, it is possible to determine if a voxel $v_i$ lies on the surface with the following predicate:

$$
\gamma(v_i) = \text{'object'} \land \exists [v_j \in \lambda(v_i)] \gamma(v_j) = \text{'air'},
$$

where $\lambda(v_i)$ is the set of all 26 neighbors (or less on the edges of the volume) of the voxel. The predicate holds when a voxel lies on a surface. Intuitively speaking, a voxel that belongs to the object which is touching at least one air voxel should belong to the surface. We will refer to these voxels as surface voxels. These surface voxels are contained in the set $V_S$ which contains $M$ voxels.

4.4 Curvature Calculation

We define $\psi(x,y,z) \mapsto \mathbb{R}$ to be a function which maps a location in a 3D euclidean space to a real number. In the context of this project, this corresponds to a voxel $v_i$ in the CT-scan and the value belonging to $v_i$. A 3D surface can be defined as the isocontour $\psi(x,y,z) = c_0$ where $c_0$ is a constant. The formulas required to compute the normal and mean curvatures at an implicit surface are well established and are as follows \cite{5}

$$
\mathbf{n} = \frac{\nabla \psi}{||\nabla \psi||} = \frac{\nabla \psi}{\left( (\frac{\partial \psi}{\partial x})^2 + (\frac{\partial \psi}{\partial y})^2 + (\frac{\partial \psi}{\partial z})^2 \right)^{1/2}},
$$

where $\mathbf{n}$ is the normal vector of the implicit surface at some voxel, and

$$
\kappa_H = \frac{\nabla \cdot \mathbf{n}}{2} = \frac{\psi_x^2(\psi_{yy} + \psi_{zz}) + \psi_y^2(\psi_{xx} + \psi_{zz}) + \psi_z^2(\psi_{xx} + \psi_{yy})}{2 \cdot (\psi_x^2 + \psi_y^2 + \psi_z^2)^{3/2}} - \frac{\psi_x \psi_y \psi_{xy} + \psi_x \psi_z \psi_{xz} + \psi_y \psi_z \psi_{yz}}{(\psi_x^2 + \psi_y^2 + \psi_z^2)^{3/2}},
$$

where $\psi_{ij}$ denotes the second partial derivative of $\psi$ with respect to $i$ and $j$. These equations are used to compute the normal and mean curvatures at each surface voxel.
where $\kappa_H$ is the mean curvature, and $\psi_x$ is the partial derivative of $\psi$ with respect to $x$. The formula for the Gaussian curvature $\kappa_K$ is as follows

$$\kappa_K = 2 \psi_x \psi_y (\psi_{xx} \psi_{yy} - \psi_{xy}^2) + \psi_x \psi_z (\psi_{xy} \psi_{xz} - \psi_{xx} \psi_{yz}) + \psi_y \psi_z (\psi_{yx} \psi_{yz} - \psi_{yy} \psi_{xz})$$

$$+ (\psi_x^2 + \psi_y^2 + \psi_z^2)$$

(4.6)

Since $\kappa_K = \kappa_{\text{min}} \cdot \kappa_{\text{max}}$ and $\kappa_H = \frac{\kappa_{\text{max}} + \kappa_{\text{min}}}{2}$, the minimal and maximal curvatures are deduced from $\kappa_K$ and $\kappa_H$ as follows

$$\kappa_{\text{min}} = \kappa_H - \sqrt{|\kappa_H^2 - \kappa_K|}$$

(4.7)

$$\kappa_{\text{max}} = \kappa_H + \sqrt{|\kappa_H^2 - \kappa_K|}$$

(4.8)

To compute the curvature at a certain voxel, all partial derivatives in Equations 4.4, 4.5 and 4.6 are approximated with a finite difference method. Some well known finite differencing methods are the forward difference, backward difference and central difference [6]. These equations follow from the application of Taylor expansion. For the forward difference, Taylor expansion is applied to compute $\psi$ at position $(x+1, y, z)$ using $\psi$ at $(x, y, z)$

$$\psi(x+1, y, z) = \psi(x, y, z) + \psi_x \Delta x + \psi_{xx} \frac{(\Delta x)^2}{2} + \psi_{xxx} \frac{(\Delta x)^3}{6} + ...$$

(4.9)

This formula can be rearranged into the following,

$$\psi_x = \frac{\psi(x+1, y, z) - \psi(x, y, z)}{\Delta x} = \frac{\psi(x, y, z) - \psi(x-1, y, z)}{2 \Delta x} + O(\Delta x),$$

(4.10)

which is the forward difference approximation of $\psi_x$ where the remaining terms on the right constitutes the truncation error. Both the backward difference and central difference can be obtained in a similar way. Below, three different approximation methods are given.

$$\psi_x = \begin{cases} \psi(x+1, y, z) - \psi(x, y, z) + O(\Delta x) & \text{Forward difference} \\ \frac{\psi(x, y, z) - \psi(x-1, y, z)}{\Delta x} + O(\Delta x) & \text{Backward difference} \\ \frac{\psi(x+1, y, z) - \psi(x-1, y, z)}{2 \Delta x} + O(\Delta x)^2 & \text{Central difference} \end{cases}$$

(4.11)

The truncation error for central difference is $O(\Delta x)^2$ in comparison with $O(\Delta x)$ for the other methods. This indicates a smaller error when $\Delta x$ is significantly small. Thus, higher resolution CT-scans will result in an improved approximation, which is even more noticeable when using central difference. Thus, the central difference method is used in this thesis. We assume that the scanner resolution is the same in each dimension. Therefore, we set $\Delta x = \Delta y = \Delta z = 1$. Using central difference, it is possible to compute all first order derivatives which are used in the Gaussian and Mean curvature formulas

$$\psi_x \approx \frac{\psi(x+1, y, z) - \psi(x-1, y, z)}{2}$$

(4.12)

$$\psi_y \approx \frac{\psi(x, y+1, z) - \psi(x, y-1, z)}{2}$$

(4.13)

$$\psi_z \approx \frac{\psi(x, y, z+1) - \psi(x, y, z-1)}{2}$$

(4.14)
Second order derivatives can be found in a similar manner.

\[
\psi_{xx} \approx \psi(x-1, y, z) - 2\psi(x, y, z) + \psi(x+1, y, z)
\] (4.15)

\[
\psi_{yy} \approx \psi(x, y-1, z) - 2\psi(x, y, z) + \psi(x, y+1, z)
\] (4.16)

\[
\psi_{zz} \approx \psi(x, y, z-1) - 2\psi(x, y, z) + \psi(x, y, z+1)
\] (4.17)

\[
\psi_{xy} \approx \frac{1}{4}(\psi(x-1, y-1, z) + \psi(x+1, y+1, z) - \psi(x-1, y+1, z) - \psi(x+1, y-1, z))
\] (4.18)

\[
\psi_{xz} \approx \frac{1}{4}(\psi(x-1, y, z-1) + \psi(x+1, y, z+1) - \psi(x-1, y, z+1) - \psi(x+1, y, z-1))
\] (4.19)

\[
\psi_{yz} \approx \frac{1}{4}(\psi(x, y-1, z-1) + \psi(x, y+1, z+1) - \psi(x, y, z+1) - \psi(x, y, z-1))
\] (4.20)

The derivation of the approximation of both the first and second derivatives can be found in *Computational fluid dynamics: the basics with applications* by John D. Anderson [6].

Figure 4.4 shows an application of computing the mean curvature at voxels belonging to the surface. Voxels having a negative mean curvature value are selected and skeletonized with a 3D skeletonization algorithm. The skeletonized voxels are laid on top of a fingerprint surface which is obtained with marching cubes[29] where the isovalue is determined with Otsu’s method. It is clear that the mean curvature captures the imprinted ridges well (or valleys of the original fingerprint), however imperfections of the clay will change the obtained fingerprint. The figure shows disconnections whenever a hole is occurring. Furthermore, lines perpendicular to the fingerprint are generally not found on fingerprints.
4.5 Sphere Unwrapping

The sphere unwrapping algorithm is based on the work by Wang et al. [42] which is discussed briefly in Chapter 2. In this section, a slightly modified version of Wang’s sphere unwrapping algorithm is explained in detail. Their 3D fingerprint extraction acquisition was exercised with a multipattern, phase-measuring profilometry (PMP). This technique projects sine waves on the to be scanned object while a camera observes the distorted sine patterns. The distortions on multiple different patterns helps in obtaining 3D information of the scanned object. Their method used a resolution of 1392 x 1040 pixels to acquire points that vary from 20 to 25 µm which is below the required 50 µm discussed in the introduction of this thesis. The volumetric data acquired by a CT-scan in this thesis will have additional information since a full volume is scanned instead of a surface of an object. This will allow scanning of fingerprints that are hidden inside the object. Wang’s method is used on fingerprints which are all scanned in a similar manner whereas the hidden fingerprints in the full CT-scan all have different orientations and will be distorted in some way. The distortions in the clay will depend on many factors and will not be dealt with in this paper. These factors includes the type of clay, how the fingerprints were applied by the creator of the object, the drying process etc.. The orientation can be corrected by rotating the surface with the center of mass to the desired orientation. Before applying a rotation to the fingerprint, a sphere is fit onto all voxels in $V_S$ with a least squares fitting algorithm which solves a system of linear equations. Solving this system results in obtaining the center point and the radius of the sphere. We construct this system as follows. First, an equation is given which relates the radius of the sphere with the center point of the sphere and a voxel touching the sphere as follows,

\[(v_i^x - x_c)^2 + (v_i^y - y_c)^2 + (v_i^z - z_c)^2 = r^2,\]  

(4.21)

where $x_c$, $y_c$ and $z_c$ are the coordinates of the center of the sphere having a radius of $r$. Furthermore, the $x$, $y$ and $z$ coordinates of a voxel are indicated with superscript. This equation can be rearranged to the following,

\[2v_i^x x_c + 2v_i^y y_c + 2v_i^z z_c + r^2 - x_c^2 - y_c^2 - z_c^2 = (v_i^x)^2 + (v_i^y)^2 + (v_i^z)^2.\]  

(4.22)

This can be rewritten in a system of linear equations for all $M$ voxels on the surface with vectors $\vec{x}$ and $\vec{b}$ and matrix $A$ as follows,

\[A\vec{x} = \vec{b},\]  

(4.23)

with,

\[\vec{b} = \begin{bmatrix} (v_1^x)^2 & (v_1^y)^2 & (v_1^z)^2 \\ (v_{i+1}^x)^2 & (v_{i+1}^y)^2 & (v_{i+1}^z)^2 \\ \vdots & \vdots & \vdots \\ (v_M^x)^2 & (v_M^y)^2 & (v_M^z)^2 \end{bmatrix} \]  

(4.24)

\[A = \begin{bmatrix} 2v_1^x & 2v_1^y & 2v_1^z & 1 \\ 2v_{i+1}^x & 2v_{i+1}^y & 2v_{i+1}^z & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 2v_M^x & 2v_M^y & 2v_M^z & 1 \end{bmatrix} \]  

(4.25)

\[\vec{x} = \begin{bmatrix} x_c \\ y_c \\ z_c \\ r^2 - x_c^2 - y_c^2 - z_c^2 \end{bmatrix} \]  

(4.26)

This linear system is efficiently solved for unknown vector $\vec{x}$ using singular value decomposition (SVD) with the Eigen3 library in c++ [3].
CHAPTER 4. 3D FINGERPRINT UNWRAPPING

Figure 4.5: A visual representation of a spherical coordinate system defined in convention ISO 13-11 [10].

Figure 4.6: Increasing resolution of the parameter space with domain $\phi$ and range $\theta$ by rotating the surface voxels, indicated with small crosses, to one side of the XZ plane. In the left graph, the domain of $\phi$ is $[0, 2\pi]$ while the domain of $\phi$ in the left graph is $[0, \pi]$.

It is important to choose a specific spherical coordinate system, which will become clear shortly. In this thesis, the ISO 13-11 convention is used which was a convention proposed by Misner et al. [46]. With this convention, visualized in Figure 4.5, an arbitrary point in euclidean space is described with $(r, \theta, \phi)$. The radial variable $r$ is the distance from origin to a particular point $P$. The azimuthal angle, which is the angle of rotation along z, is expressed as $\phi$. The third coordinate is designated to the polar angle $\theta$, which is the angle between $OP$ an the z-axis.

In sphere unwrapping, every $\theta$ and $\phi$ corresponds to a specific property of a surface voxel, which is a selected voxel belonging to the surface of an object. The distance between a voxel and the center of the sphere, the mean curvature, Gaussian curvature and the principal curvature values are a few of those properties which are examined in this thesis. Since a fingerprint only needs to be captured by one half of the sphere, or dome, the parameter space with domain $\phi$ and range $\theta$ can be decreased. The orientation of the surface voxels in the volume will influence the domain and range. Figure 4.6 illustrates how the placement of the surface voxels, depicted by the small crosses, will affect the parameter space. In both graphs, the range of $\theta$ is roughly $[\pi/4, 3\pi/4]$. On the other hand, the domain of $\phi$ differs in each graph. In the left graph, the domain of $\phi$ is $[0, 2\pi]$ while it is roughly $[\pi/4, 3\pi/4]$ in the right graph. The decrease in domain $\phi$ is met when the surface voxels are not crossing the XZ or YZ plane. Since the extracted fingerprints can be oriented in any way, it is necessary to rotate the surface voxels to an optimal orientation. Rotation of the surface voxels can be performed by obtaining the center of mass and rotating this vector to the correct axis. In this thesis, the center of mass vector will be rotated the positive side of the Y-axis. It is assumed that each voxel has the same weight. The center of mass is then obtained.
as follows,

\[ v_{\text{center}} = \frac{1}{M} \sum_{i=1}^{M} v_i. \]  

(4.27)

Note that \( v_{\text{center}} \) does not point to an exact location in the fingerprint but to the weighted average over all voxel locations. Thus it is expected that \( v_{\text{center}} \) is a certain distance away from the center of the fingerprint surface.

\[ \theta_{l_1}^{\text{linear}} = (l_1 - 1)t_\theta + \theta_{\text{min}} \]  

(4.28)

\[ \phi_{l_2}^{\text{linear}} = (l_2 - 1)t_\phi + \phi_{\text{min}} \]  

(4.29)

where \( l_1 \in \{1, 2, \ldots, L_1\} \) and \( l_2 \in \{1, 2, \ldots, L_2\} \). \( L_1 \) and \( L_2 \) are the width and height of the resulting image, and \( \theta_{\text{min}} \) and \( \phi_{\text{min}} \) are the minimum value in the set of \( \theta \) and \( \phi \) respectively. The stepsize \( t_\theta \) and \( t_\phi \) must be sufficiently small to capture the full detail of the scanned fingerprint. Both \( L_1 \) and \( L_2 \) can be obtained as follows,

\[ L_1 = \frac{\theta_{\text{max}} - \theta_{\text{min}}}{t_\theta} \]  

(4.30)

\[ L_2 = \frac{\phi_{\text{max}} - \phi_{\text{min}}}{t_\phi} \]  

(4.31)

where \( \theta_{\text{max}} \) and \( \phi_{\text{max}} \) are the maximum values in the set of \( \theta \) and \( \phi \) respectively. Which voxel is used at location \((\phi, \theta)\) to compute a certain property (\( \rho \) or a curvature) is determined with a nearest neighbor algorithm.

### 4.6 Measure of Distortions

The currently implemented sphere unwrapping algorithm does apply a linear mapping from voxel space to 2D. Since the sphere is not an exact approximation of the surface of the fingerprint, and the distortions that occur due to mapping a dome to a flat surface, distance and angle relations between points in the final 2D image will not fully capture the original data. Relative angles or distances between minutiae are used in AFIS to compare different fingerprint samples with each other. Wang’s method for distortion correction has not been implemented in this thesis. However, their quantitative distortion analysis can be used to get a sense of the amount of distortion occurring with non-linear unwrapping. In Figure 4.7, the distance between neighboring pixels are shown for both linear and non-linear unwrapping. The distance between different voxels, or points in the point cloud, should be constant such that the distance between two minutiae is preserved. The figure shows that with linear unwrapping, points tend to be more spread out in the center of the image in comparison with points on the edges of the image. A non-linear unwrapping approach eliminates almost all differences in distance between the sampled points.
Figure 4.7: A slice of the fingerprint image along $\theta$ where the distances between two neighboring pixels is measured. Images taken from Wang’s paper [42].
Chapter 5

AFIS

The last stage of the pipeline consists of finding a match between multiple 2D fingerprint images. The type of AFIS that is used takes two or more images as input, and gives a score based on the similarity in their minutiae. There are many AFIS systems available. SourceAFIS is an open source AFIS which delivers decent accuracy with a high matching speed [40]. The system is both rotation- and translation-invariant but not scale-invariant. It is suggested to use an algorithm for determining the DPI of the input images or to know the DPI beforehand.

Another AFIS, or rather a group of packages containing biometric imaging software, which is used among many research papers regarding fingerprint extraction is developed by the National Institute of Standards and Technology (NIST). The institute provide a total of five mayor packages under the name NIST Biometric Image Software (NBIS)

1. PCASYS: a neural network based fingerprint pattern classification system. Fingerprints are categorized in a subset of Henry’s classification system: arch, left loop, right loop, scar, tented arch, and whorl. Combinations of these different classes are not classified.

2. MINDTCT: automatic identification of minutiae where both the orientation and location of minutiae are recorded.

3. NFIQ: a fingerprint image quality tool. The quality of a fingerprint image can be used to perform a specific strategy for fingerprint matching or to indicate if the fingerprint should be reextracted to reliably perform fingerprint matching.


5. IMGTOOLS: a collection of image utilities. In particular, this package contains BOZORTH3 which is a minutiae based fingerprint matching system. This system can be used to analyze two fingerprint samples or to compare one fingerprint to a database of fingerprints. BOZORTH3 uses the the minutiae found with MINDTCT as its input.

It is clear that a combination of MINDTCT and BOZORTH3 can be used as an AFIS. A brief explanation of MINDTCT is given in this chapter. For a detailed explanation of MINDTCT, consult the detailed user guide of NBIS [43]. MINDTCT is free to use and open source while BOZORTH3 is closed source and not explained in the user guide of NBIS. The minutiae identification software will be applied on our own data in Chapter 6.

5.1 MINDTCT

Each step of the algorithm is analyzed and remarks are given where needed, including the input constraints and output. The algorithm has been designed to work optimally for scanned at 19.69
pixels per millimeter (ppmm), or 500 pixels per inch (ppi), and quantized to 256 levels of gray. Thus, this minutiae extraction algorithm is not scale invariant.

The minutiae extraction pipeline consists of eight stages. They are displayed in Figure 5.1. All different image maps which are obtained by the algorithm are used to construct the image quality map and to perform image binarization. The image quality maps stage does produce an image which indicates what the quality of the fingerprint is. This quality map is based on many characteristics of good and bad regions such as the local contrast, the lack of flow, regions of extreme curvature. This step is mainly performed to cope with latent, i.e., bad quality, fingerprints.

Image binarization is performed with the help of the ridge flow map which was obtained in the previous stage. It is noted that binarization is a critical phase in this minutiae extraction algorithm. While significant effort was designated to obtain reliable and robust binarized images, the authors emphasize that this step will produce a considerable number of false minutiae.

The next step involves obtaining minutiae. This is performed by using pixel patterns: 6 binary pixels in a 2x3 image. The binarized fingerprint image is scanned both vertically and horizontally with these pixel patterns. Whenever the fingerprint image contains one of the fingerprint patterns, that spot is marked as a minutia. This is a greedy approach in obtaining locations of minutiae and many false minutiae are obtained in this step. The subsequent stages of the algorithm is therefore dedicated to deletion of false minutiae.

A visualization of the output of this algorithm is given in Figure 5.2. The endpoints are indicated by circles and bifurcations with squares. Furthermore, the angle the ridge line at the minutiae is given as well.
Figure 5.2: Output of MINDTCT where endpoints are indicated by circles and bifurcations with squares. Furthermore, the angle the ridge line at the minutiae is given as well. Image from the user guide of NBIS [43].
Chapter 6

Results

In this chapter, the output of the sphere unwrapping algorithm will be explored. An analysis on the different curvatures with respect to $\rho$ will be performed. Furthermore, a qualitative analysis will show whether the developed algorithm can be used to identify fingerprints belonging to the same finger. The performance of the algorithm in terms of the runtime will be shown for four different samples. Finally, MINDTCT is applied on one sample to show that our data can be used for automatic minutiae detection.

6.1 Samples

A total of five fingerprint samples, which belong to three distinct fingers, are used in this chapter. Each scanned object is a small ball of modeling clay containing fingerprints. The subjects either put one or two fingerprints on each ball. An example of a scanned object is shown in Figure 6.1. Multiple factors can influence the quality of the sample. The wetness of the clay, the wetness of the finger, and the room climate are some of these factors. These factors have not been recorded. However, we can say that the samples are created at room temperature and the modeling clay was acclimated to room temperature.

![Sample](image.jpg)

Figure 6.1: An example of a sample that is to be scanned with CT.
6.2 Fingerprint Representation Analysis

In this analysis, it is explored which type of representation captures the ridge lines the best. Five different representations of the fingerprint are shown in Figure 6.2. A black value corresponds to a low value of the used property that is displayed. The initial sphere unwrapping approach by Wang et al. used the distance between the center of the sphere to a voxel (\( \rho \)) to obtain a 2D image. While the fingerprint is visible with this method, large dark areas will appear in the image. This is caused by the fact that the surface of the fingerprint cannot be modeled by a sphere exactly. Thus, a \( \rho \) corresponding to a ridge line in one part of the image can correspond to a valley in another part of the image. A curvature based approach is more robust since the curvature property does not change for a different placement of the sphere. The maximum curvature representation shows interesting white spots. They correspond to parts of the fingerprint that are highly curved into the clay which are normally not found in a valley. These highly curved areas give an indication that the quality of the fingerprint is worse in these regions.

A ridge line can be described with the principal curvatures. Since inward dents are considered to be positive, ridge lines will have low \( k_{\text{max}} \) and a high \( k_{\text{min}} \). This is the opposite for valleys. For the mean curvature, which is defined to be \( \frac{k_{\text{max}} + k_{\text{min}}}{2} \), this equates to low values for ridges and high values for valleys. The Gaussian curvature is defined to be the product of the principal curvatures. Valleys will be closely related to the ridge lines due to the values of \( k_{\text{max}} \) and \( k_{\text{min}} \) on valleys and ridges as has been previously discussed. This is why the valleys are relatively dark in contrast with the mean curvature image. Since a ridge can both be captured by either one of the principal curvatures, but we are not interested in capturing regions of extreme curvatures, it makes sense to use the mean curvature (or an addition of \( k_{\text{min}} \) and \( k_{\text{max}} \)) to capture the ridge lines.
6.3 CT Scans & Algorithm Output

Regions of interest (ROI), are selected from the scanned samples. For every ROI, a fingerprint is unwrapped and a median filter with a radius of 3 and gaussian filter with a variance of 3 is applied. Isosurfaces are extracted with marching cubes where the iso value is determined with Otsu’s method. They are shown in Figure 6.3. It is clear that fingerprint identification with these surfaces is hard to perform by eye since the ridge lines are displayed subtly. The output of the developed fingerprint extraction method is shown in Figure 6.4. The positive and negative curvature discriminate the valleys from the ridge lines. This results in images with high contrast between the valley and ridge lines in comparison with the marching cubes extracted isosurface. Both the global structure and individual minutiae are now clearly visible. The unwrapping process uses a nearest neighbor algorithm to obtain the nearest voxel to a certain $\theta$ and $\phi$. Since there are
no voxels on the sides of the unwrapped fingerprint images, triangular shaped areas are visible. Furthermore, some black spots spanning a few minutiae can be seen. Those are additional surfaces which are extracted during the surface extraction stage. Additional surfaces can be extracted when small holes are present in the used medium. Furthermore, thin objects can lead to multiple surfaces being extracted at once. Solutions to the extraction of multiple surfaces are discussed in Chapter 7.

6.4 Qualitative Identification Analysis

In this section, four different fingerprint samples are qualitatively analyzed in order to identify which pair belongs to the same fingerprint. There are two pairs of fingerprints that are scanned at a different location of the clay. The four fingerprint samples are shown in Figure 6.4 and are labeled as FP1, FP2, FP3, and FP4. It must be taken into account that all fingerprint can be oriented in any way. It can be helpful to classify a fingerprint based on a level 1 feature (Henry’s classification) to get a first impression. A loop, visible in the top right of FP1 and FP2, can be seen in only two of all the samples. Furthermore, there is no obvious level 1 characteristic property visible in FP3 and FP4. This is a first indication that FP2 and FP3 forms a pair, and FP1 and FP4 forms a pair. Level 3 features (sweat pores) are not possible to obtain from these images as they require higher resolution scans as has been explained in the introduction. Thus, level 2 features remain to be explored (minutiae). For each fingerprint, a small area is shown where bifurcations and ridge endings are marked. The pairs that were found to be similar on level 1 features are displayed besides each other in Figure 6.5 and 6.6. The minutiae are marked with yellow circles and a red outline is shown. It is clear that the shape of the red outlines is similar in both pairs shown in the subfigures. Thus, qualitatively, it is shown that two different fingerprint samples can be compared and distinguished from each other.
Figure 6.3: Four different fingerprint samples. Surfaces extracted with marching cubes with an isovalue obtained with Otsu’s method. Two pairs of fingerprints belonging to the same finger.

Figure 6.4: Four different fingerprint samples obtained with sphere unwrapping. Two pairs of fingerprints belonging to the same finger.
CHAPTER 6. RESULTS

Figure 6.5: Zoomed in area of FP1 and FP2. Minutiae found in both fingerprints are marked with yellow circles. Furthermore, a red outline is formed by connecting minutiae.

Figure 6.6: Zoomed in area of FP3 and FP4. Minutiae found in both fingerprints are marked with yellow circles. Furthermore, a red outline is formed by connecting minutiae.

6.5 Performance & Implementation

The developed algorithm for fingerprint extraction was performed on an Intel® Xeon® CPU E5-1650 v3 at 3.50GHz. The time of both the whole fingerprint extraction process and the time excluding the median and Gaussian filter steps are shown in Table 6.1. The program is implemented in C++. The Insight Toolkit (ITK)[3] library has been used to perform the filtering tasks and to obtain a threshold value with Otsu’s method. A nearest neighbor algorithm was used, Approximate Nearest Neighbor (ANN)[1], to perform the unwrapping stage.

Table 6.1: The performance of the developed algorithm for the four labeled fingerprint samples. The total time includes the median and Gaussian filtering step.

<table>
<thead>
<tr>
<th></th>
<th>total (s)</th>
<th>unwrapping (s)</th>
<th>size</th>
</tr>
</thead>
<tbody>
<tr>
<td>FP1</td>
<td>87</td>
<td>20</td>
<td>750x363x737</td>
</tr>
<tr>
<td>FP2</td>
<td>159</td>
<td>37</td>
<td>899x484x859</td>
</tr>
<tr>
<td>FP3</td>
<td>87</td>
<td>22</td>
<td>740x363x723</td>
</tr>
<tr>
<td>FP4</td>
<td>150</td>
<td>32</td>
<td>938x484x781</td>
</tr>
</tbody>
</table>

6.6 Automatic Minutiae Extraction with MINDTCT

Figure 6.7 shows the application of the MINDTCT algorithm applied on our own data. Red circles indicate bifurcations while the green squares indicate ridge endpoints. It is clear that the output of the sphere unwrapping algorithm can be used for minutiae matching. However, certain parts of the fingerprint, especially the right part of the image, contain many false positive minutiae due to how the fingerprint was left behind in the clay.
Figure 6.7: Output of FpMV (Fingerprint Minutiae Viewer) which utilizes MINDTCT. The red circles indicate bifurcations while the green squares indicate ridge endpoints.
Chapter 7

Discussion

As has been shown by the qualitative analysis in Chapter 6, the developed fingerprint extraction method is suitable for fingerprint identification with CT data. Currently, there are no papers which describe fingerprint extraction from CT data. Additionally, contrary to all other papers which are based on 3D point clouds, curvature computation is performed in voxel space. This increases the accuracy and decreases discretization errors in comparison with explicitly extracted surfaces.

Some parts which were not clearly explained in Wang’s sphere unwrapping paper have been well defined in this thesis. In this thesis, a specific spherical coordinate system has been chosen. This was necessary in order to obtain an unwrapped image with less distortions. In contrast to Wang’s sphere unwrapping algorithm, our algorithm can not assume that each fingerprint is oriented equally. Thus an additional step where the surface voxels are rotated to a correct position was necessary.

It has been shown that the samples used in this thesis can be discerned from each other. Henry’s classification method allowed for an initial impression of the fingerprints. Focusing on high quality regions in the 2D image, where minutiae are easily identified by a user, reveals groups of minutiae which are shared among samples belonging to the same fingerprint. The relative distances and angles within minutiae in those groups can be used to compare different samples with each other and to obtain a match. Automatic marking of minutiae on fingerprints was performed on a high quality scan. This indicates that automatic matching based on minutiae can be performed but it does not indicate if the distortions introduced by sphere unwrapping are low enough to obtain accurate results.

The developed fingerprint extraction algorithm is a deterministic approach and only uses three parameters in total. The first two parameters are used for the filtering stage. These are the radius of the median filter and the variance of the Gaussian filter. The third parameter is the stepsize of the angles in the linear map. This is a measure for the amount of angles which are sampled for the final image and therefore correlates with the resolution of the final image.

Additional methods to reduce distortions for sphere unwrapping, as has been proposed by Wang et al., have not been implemented. Therefore, distortions will be more prominent than they could be. Distortion correction methods can either preserve angles or distances between minutiae but not both. These angles and distances can be used by an AFIS to perform fingerprint identification. It is therefore expected that implementing an anti-distortion will increase the chance of having a match between similar fingerprints.

The developed fingerprint extraction algorithm assumes that the extracted surface contains the fingerprint. Furthermore, the current surface extraction algorithm may extract multiple surfaces. Small holes in the volume will degrade the quality of the final 2D image. They are visible in the 2D image as black shapes covering a few ridge lines. Furthermore, fingerprint extraction of thin objects can be difficult since the surface on the other side of the object might be included in the final image. Additional measures must be taken to solve these issues. Both problems can be categorized to be the problem of extracting more than one surface. If the surfaces are grouped,
CHAPTER 7. DISCUSSION

(a) Small holes.  
(b) Thin object.

Figure 7.1: Examples for the cause of some artifacts which occur when multiple surfaces are extracted. The images depict a cross section of the surfaces which are currently extracted. The bumpy green surface is the surface containing the fingerprint ridge lines. The red surfaces should not be extracted. The black dot indicates the center of the sphere.

the removal of small groups of voxels is a trivial solution for small holes in the volume. For thin surfaces, the amount of selected voxels at the side of the fingerprint are close to the amount of voxels on the opposite side. Since we expect the fingerprint to be located in the concave part of the object, the amount of voxels that contain the fingerprint are expected to be less than the amount of voxels on the opposite side. Thus for similar sized groups of voxels, the smallest group should be selected. While it is expected that the voxel group containing the fingerprint will be smaller than the surface on the opposite, this is not necessarily true for all cases. Thus one could opt for a solution where sphere unwrapping will be performed on both voxel groups individually, a method at which a user can specify which side should be unwrapped, or an automated method which analyses the two groups on certain frequencies.

Additional steps are required to increase the usability and effectiveness of the developed fingerprint extraction algorithm. For instance, the algorithm uses a region of interest (ROI) as its input. This is currently created manually. For a fully automated system, the ROI should be found automatically. This is currently an open question. A small literature review and a possible implementation is discussed in Appendix A.

An area which should be explored to increase the effectiveness of the given algorithm is on partial fingerprints. The intention by the artist is not to leave his fingerprint behind but rather to create a piece of art. Therefore, fingerprints that are left by the creator of the art object are generally not imprints of a full fingerprint. Thus, additional software is needed that can stitch different fingerprints based on shared minutiae.
Chapter 8

Conclusion

Fingerprint extraction is normally performed with scanning methods which capture fingerprints in plain sight. We have shown that fingerprint detection from CT data is feasible. This has the advantage over all current fingerprint identification methods that fingerprints can be extracted which are hiding within an object. Additionally, the pipeline is completely in voxel space up until the point at which a 2D fingerprint image is formed. All fingerprint extraction algorithms which use curvatures normally compute the curvatures on a point cloud. A point cloud corresponds to an explicit surface. The advantage of computing the mean curvature in voxel space, on an implicit surface, is the increased accuracy and a reduction in discretization errors. Lastly, the thesis adds some additional definitions to the sphere unwrapping process which are not explained in the literature. Namely, it is necessary to choose a specific spherical coordinate system and rotate the fingerprint in a predefined position to extract the fingerprint such that the resolution of the parametric space is as high as possible.

While this thesis shows that fingerprint extraction from CT data is feasible, many problems still need to be addressed to fully automate the whole process. The key area of improvement is the automated extraction of ROIs which contain a fingerprint. This would automate the fingerprint extraction process completely and would be a complete breakthrough in acquiring fingerprints from ancient museum sculptures.
Bibliography


Appendix A

Dents detection

In this appendix, both a small literature review and a possible solution to dents detection is given. The algorithm has not been implemented but contributes to the currently lacking view on this problem.

A.1 Literature Review

With respect to automatic detection of dents on a surface, not much research has been conducted. Though, there exists a field within computer vision on feature extraction in both 2D and 3D data. Methods for dents detection are normally focussed on finding faults in produced products. This simplifies the detection process since the algorithm can use a model of reference. In this thesis, we are interested in an approach which finds characteristic dents that are likely to contain a fingerprint. It is thus of importance to identify these specific characteristics in order to select them.

Lippincott et al. created an optical approach on the detection of dents and scratches on specular metal surfaces in 1982[28]. Detection was performed by projecting a grid onto a reflecting flat surface of metal. Dents were revealed by distortions in the projected grid. The method was able to detect dents as small as 1% of the viewed field and 0.0053 cm deep. While the proposed method was used to find dents on a flat surface, it was noted that the method purely relied on distortions in the projected grid which could likewise be applied on a non-flat surface.

A method for dents detection was performed by Leroy et al. and was on crater detection for autonomous landing on asteroids [27]. Craters were extracted from a 2D image of the asteroid. Curvatures in the 2D images were estimated and used to extract ellipses by a non iterative least square algorithm. A 3D model of the asteroid, where each crater was already marked, was used to assess the orientation of the spacecraft.

A.2 Proposed Method

Based on the surface extraction and curvature calculation methods used in the developed fingerprint extraction algorithm, a possible method for dents detection is given. The proposed method assumes a low resolution scan of the art object. Thus, the ridge lines will not be captured by the scan.

The proposed method resembles partly the crater detection method by Leroy et al.. Curvature estimation in this paper was limited in the sense that only a part of the rim of a crater could be used for curvature estimation. Thus, the authors applied ellipse fitting to obtain a full description of a crater. This problem does not exist in our case since we can accurately compute the curvature at each voxel.

Once the surface voxels have been extracted and the mean curvature is computed at each voxel, all voxels belonging to a dent need to be found. The rim of a dent can be obtained by selecting
APPENDIX A. DENTS DETECTION

voxels having substantial negative mean curvature. The selected rims are then to be grouped with a clustering algorithm. A clustering algorithm which does not assume a certain amount of groups must be used since a priori it is not known how many dents an object has. Thus, k-means clustering should not be used. However, clustering based on density could be performed. A specialized clustering algorithm can also be performed in which each voxel in a cluster is a neighbor of one other voxel in the cluster. It is expected that such a method is not as robust as a general density based clustering algorithm.

After obtaining all clusters in the volume, specific clusters can be selected based on the size of their bounding box. Only clusters that are about the size of a finger, with a range specified by the user or analytical data, should be selected. The coordinates of the bounding boxes, and the bounding box itself, can be used to scan specific regions of the object at a higher resolution. These subvolumes are then to be used by the fingerprint extraction stage of the proposed pipeline in this thesis.