Virtual tool mark generation for efficient striation analysis in forensic science

by

Laura Ekstrand

A thesis submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE

Major: Mechanical Engineering

Program of Study Committee:

Song Zhang, Major Professor

L. Scott Chumbley

W. Ross Morrow

Iowa State University

Ames, Iowa

2012

Copyright © Laura Ekstrand, 2012. All rights reserved.

TABLE OF CONTENTS

LIST OF TABLES	iv
LIST OF FIGURES	v
ACKNOWLEDGEMENTS	viii
ABSTRACT	x
CHAPTER 1. OVERVIEW	1
1.1 Background	1
1.2 Motivation	2
1.3 Proposed approach	3
1.4 Thesis overview	5
1.5 Summary	5
CHAPTER 2. REVIEW OF LITERATURE	6
2.1 Research prior to <i>Daubert</i>	6
2.2 Preliminary research in 3D automated comparisons	7
2.3 State-of-the-art in automated comparisons	9
2.3.1 Developing statistical metrics	10
2.3.2 Utilizing numerical pattern recognition	13
2.4 Summary	15
CHAPTER 3. METHODS AND PROCEDURES	16
3.1 Data acquisition and preparation	16
3.1.1 Geometry acquisition	16
3.1.2 Tip noise reduction	18

3.1.3	Plate noise reduction	22
3.1.4	Coordinate system assignment	23
3.2	Virtual tool mark generation	29
3.2.1	Depth camera	33
3.2.2	Drawing the tip	35
3.2.3	Converting from projected coordinates back to μm	38
3.2.4	Detecting the edges	39
3.3	Graphical user interface	40
3.4	Summary	44
CHAPTER 4.	RESULTS AND DISCUSSION	45
4.1	Verification with known geometric standards	45
4.2	Statistical comparison results	48
4.2.1	Known match results	50
4.2.2	Known non-match results	57
4.3	Summary	62
CHAPTER 5.	CONCLUSIONS AND FUTURE WORK	63
5.1	Conclusions	63
5.2	Future work	64
5.3	Summary	66
BIBLIOGRAPHY		67

LIST OF TABLES

Table 4.1	RMS error (%) for standard geometry virtual marks.	49
Table 4.2	Error ($^{\circ}$) between the predicted angle of a known match and the true angle.	56

LIST OF FIGURES

Figure 2.1	The origins of the T1 statistical measure. (a) and (b) Rigid-shift and maximum correlation windows. (c) Random-shift and maximum correlation windows. Reprinted from Chumbley et al. (2010) with permission.	11
Figure 3.1	The Alicona Infinite Focus Microscope (IFM) Model G3 used in this research.	17
Figure 3.2	Spike noise removal. (a) Tip digitized with no additional lighting. (b) Tip in (a) after algorithmic cleaning. (c) Tip digitized with new lighting procedure, prior to algorithmic cleaning.	19
Figure 3.3	Close-ups of the tip lighting in the IFM, showing the four fiber optic cables.	20
Figure 3.4	Tip 8 Side A, prior to algorithmic cleaning. (a) Texture overlaid on 3D geometry. (b) 2D texture. (c) Alicona-computed quality map.	21
Figure 3.5	Tip 8A, after cleaning. Texture-mapped view.	22
Figure 3.6	Plates before ((a)-(c)) and after ((d)-(f)) noise reduction. (a) and (d) Plate marked by Tip 25 Side A at 45° , scanned at 5x magnification. (b) and (e) Plate 8A 45° at 10x. (c) and (f) Plate 44B 60° at 10x.	23
Figure 3.7	The marking process. (a) The 45° marking jig with fitted screwdriver handle and tip. (b) Marking diagram.	24
Figure 3.8	Tip coordinate system. (a)-(c) View of tip side. (d) View looking at the flat side of the screwdriver.	25
Figure 3.9	Process of locating the edge points.	26

Figure 3.10	x axis determination. (a)-(b) The screwdriver tip jig. (c) Diagram of the jig during IFM measurement.	27
Figure 3.11	The coordinate system transformation applied to the data. (a)-(c) The side view with y axis rotations of (a) 0° , (b) 60° , and (c) 85° . (d) View of the flat side of the screwdriver with a rotation of 0°	29
Figure 3.12	The mark as a projection of tool tip geometry in the direction of tool travel.	30
Figure 3.13	The depth camera concept.	32
Figure 3.14	Meshing between a left and right copy of the vertex data. (a) Surfaces formed. (b) Meshing pattern that avoids redundant surfaces.	37
Figure 3.15	Mathematical model of the graphics pipeline for this research.	38
Figure 3.16	Virtual mark trimming. The sides of the purple box indicate the suggested trim points.	41
Figure 3.17	The graphical user interface.	42
Figure 3.18	Height map plot.	43
Figure 4.1	The chevron extrusion geometric standard. (a)-(c) Low resolution (200 samples). (d)-(f) High resolution (2000 samples). (a), (d) The geometry standard. (b), (e) The virtual mark compared with the ideal geometry function. (c), (f) The error between the virtual mark and the ideal function in μm	46
Figure 4.2	The half cylinder geometric standard. (a)-(c) Low resolution (200 samples). (d)-(f) High resolution (2000 samples). (a), (d) The geometry standard. (b), (e) The virtual mark compared with the ideal geometry function. (c), (f) The error between the virtual mark and the ideal function in μm	47

Figure 4.3	The trapezoid extrusion geometric standard. (a)-(c) Low resolution (200 samples). (d)-(f) High resolution (2000 samples). (a), (d) The geometry standard. (b), (e) The virtual mark compared with the ideal geometry function. (c), (f) The error between the virtual mark and the ideal function in μm	48
Figure 4.4	A comparison between a virtual and a real mark made by Tip 8 Side B at 45°	50
Figure 4.5	Averaged T values from statistical comparisons of 45° physical marks to virtual marks made with same side of the same tip.	51
Figure 4.6	Averaged T values from statistical comparisons of 60° physical marks to virtual marks made with same side of the same tip.	52
Figure 4.7	Averaged T values from statistical comparisons of 85° physical marks to virtual marks made with same side of the same tip.	52
Figure 4.8	Plate textures for comparison. (a) Plate 8A-45. (b) Plate 5A-45. (c) Plate 2A-85.	54
Figure 4.9	A successful comparison with Plate 5A-45.	55
Figure 4.10	A successful comparison with Plate 2A-85.	56
Figure 4.11	Averaged T values from statistical comparisons of 45° physical marks to known non-matching virtual marks.	58
Figure 4.12	Averaged T values from statistical comparisons of 60° physical marks to known non-matching virtual marks.	59
Figure 4.13	Averaged T values from statistical comparisons of 85° physical marks to known non-matching virtual marks.	59
Figure 4.14	A known non-match comparison with a negative T value of high magnitude.	61

ACKNOWLEDGEMENTS

I would like to say a special thanks here to all who helped me with this research. First of all, I thank my advisor, Dr. Song Zhang, for the privilege of joining his research group and working under him. I am very thankful for his kindness and patience with me during this project and that he let me pursue my passion for software development and organization. I also thank my committee members, Dr. L. Scott Chumbley and Dr. W. Ross Morrow. Dr. Chumbley has been instrumental in leading the direction of this research, and Dr. Morrow helped to convince me to go to graduate school and inspired me to solve computational problems, which I enjoy. Moreover, Dr. Max Morris was indispensable in helping me interpret and summarize my results. Thanks to his graduate student, Amy Hoeksema, for helping me understand and integrate the statistical algorithm. A special thanks also goes to Taylor Grieve, who did a fabulous job of digitizing the tips and marks. The results in this document would likely have been far less promising without her new lighting procedure, which reduced the noise drastically. Thank you to Jim Kreiser, retired Illinois State Police tool mark examiner, for sharing his professional experience with tool mark analysis. Without this, the project would certainly not have been as relevant and connected to the forensics community. I would also like to thank him for making all of those screwdriver marks in lead for this and previous studies. Finally, thanks to those other contributors who came before me to help this project take shape.

This thesis was supported by Award No. 2009-DN-R-119 from the National Institute of Justice, Office of Justice Programs, U.S. Department of Justice. The opinions, findings, and conclusions or recommendations expressed in this publication are those of myself and my collaborators and do not necessarily reflect those of the Department of Justice. Thanks also to Ames Laboratory, part of the U.S. Department of Energy; the Midwest Forensics Resource Center; and Iowa State University for supporting this work. This work was performed at the Ames Laboratory under contract number DE-AC02-07CH11358 with the U.S. Department of

Energy. The document number assigned to this thesis is IS-T 3082.

Finally, I want to thank those closest to me for their support and encouragement. Thanks especially to my husband, Jason Ekstrand, for his patience and love through the times when this really stressed me, and also for giving me valuable programming advice along the way. I would like to thank my parents for giving me the opportunity to go to Iowa State, and I would like to thank them, my in-laws, and my friends in Ames and Houston for their many prayers on my behalf and generous support during this time. Finally, the glory for this truly belongs to God the Father and the Lord Jesus Christ, who made the world and everything in it.

ABSTRACT

In 2009, a National Academy of Sciences report called for investigation into the scientific basis behind tool mark comparisons (National Academy of Sciences, 2009). Answering this call, Chumbley et al. (2010) attempted to prove or disprove the hypothesis that tool marks are unique to a single tool. They developed a statistical algorithm that could, in most cases, discern matching and non-matching tool marks made at different angles by sequentially numbered screwdriver tips. Moreover, in the cases where the algorithm misinterpreted a pair of marks, an experienced forensics examiner could discern the correct outcome. While this research served to confirm the basic assumptions behind tool mark analysis, it also suggested that statistical analysis software could help to reduce the examiner's workload.

This led to a new tool mark analysis approach, introduced in this thesis, that relies on 3D scans of screwdriver tip and marked plate surfaces at the micrometer scale from an optical microscope. These scans are carefully cleaned to remove noise from the data acquisition process and assigned a coordinate system that mathematically defines angles and twists in a natural way. The marking process is then simulated by using a 3D graphics software package to impart rotations to the tip and take the projection of the tip's geometry in the direction of tool travel. The edge of this projection, retrieved from the 3D graphics software, becomes a virtual tool mark. Using this method, virtual marks are made at increments of 5° and compared to a scan of the evidence mark. The previously developed statistical package from Chumbley et al. (2010) performs the comparison, comparing the similarity of the geometry of both marks to the similarity that would occur due to random chance. The resulting statistical measure of the likelihood of the match informs the examiner of the angle of the best matching virtual mark, allowing the examiner to focus his/her mark analysis on a smaller range of angles.

Preliminary results are quite promising. In a study with both sides of 6 screwdriver tips and 34 corresponding marks, the method distinguished known matches from known non-matches

with zero false positive matches and only two matches mistaken for non-matches. For matches, it could predict the correct marking angle within $\pm 5\text{-}10^\circ$. Moreover, on a standard desktop computer, the virtual marking software is capable of cleaning 3D tip and plate scans in minutes and producing a virtual mark and comparing it to a real mark in seconds.

These results support several of the professional conclusions of the tool mark analysis community, including the idea that marks produced by the same tool only match if they are made at similar angles. The method also displays the potential to automate part of the comparison process, freeing the examiner to focus on other tasks, which is important in busy, backlogged crime labs. Finally, the method offers the unique chance to directly link an evidence mark to the tool that produced it while reducing potential damage to the evidence.

CHAPTER 1. OVERVIEW

The goal of forensic tool mark analysis is to determine whether or not a particular evidence mark was made by a suspect tool. Forensic examiners have used conclusions from this type of analysis as expert witness testimony in the court of law for decades. Recently, legal critics have called into question the foundations of tool mark analysis along with other forensic examination practices, believing them to be subjective and unscientific. This thesis describes a proposed methodology for tool mark analysis that has the potential to both reinforce and streamline the discipline. This new methodology creates virtual marks from a 3D representation of the tool tip for comparison with evidence marks, thereby providing the possibility to link a tip directly to a mark. This chapter will discuss the background of tool mark analysis, explain its recent criticism, and outline the proposed approach and its potential advantages for forensic science.

1.1 Background

A conventional tool mark analysis involves comparing an evidence mark to known marks generated with the suspect tool. A tool mark examiner will generate the additional marks by hand on lead plates (Petraco et al., 2005). Lead is chosen because it is a soft metal and therefore easily marked. Examiners also believe that it prevents damage to the tip, which should be protected as much as possible as part of the evidence for a case. These marks are generated at multiple angles, twists, and levels of pressure. This practice results from the assumption, developed by experience over the history of the discipline, that marks made by the same tip only match at similar angles, twists, and pressures.

Once these marks are made, the examiner compares them one-by-one to the evidence mark using a comparison microscope. This is essentially a side-by-side pair of microscopes that

present their close-up views immediately adjacent to each other. The examiner carefully adjusts the position of the marks in the comparison microscope until they line up in the view (if possible). With the marks in place, the examiner uses one of two approaches to determine whether or not there is a match. The first one relies on the examiner’s experience to determine if the patterns present in the two marks are similar. The second approach is Consecutively Matching Striae (CMS). CMS involves finding groups of consecutive lines (striae) in the pattern that match. A positive match will have a combination of a certain number of lines in a group and a certain number of non-adjacent groups. Based on one of these analyses, according to the Association of Firearm and Toolmark Examiners (1998), the examiner will classify the relationship between tip and mark as Identification, Inconclusive, Elimination, or Unsuitable for Comparison.

1.2 Motivation

Recently, the admissibility of tool mark analysis results into the courtroom has come under attack. This arises from the recent change in evidence admissibility laws in most states following the results of *Daubert v. Merrell Dow Pharmaceuticals, Inc.* (1993). Prior to *Daubert*, U.S. courts used the standard established in *Frye v. United States* (1923), which allowed expert witness testimony as long as it was based on theory generally accepted in the expert’s discipline. Since tool mark analysis was generally accepted in forensic science, it was allowed. The new *Daubert* standard enforces Federal Rule 702, which requires the testimony to be “based upon sufficient facts or data” and “the product of reliable principles and methods” (The Committee on the Judiciary House of Representatives, 2009). This change has caused the legal community to question and sometimes attempt to exclude or otherwise limit testimony based on the results of impression analysis (Petraco et al., 2012b).

Critics claim that error rates are unknown and incalculable and that the experience-based approaches used are inherently subjective. This criticism is somewhat exaggerated, as there have been many prior studies that bolster tool mark analysis theory (Chumbley et al., 2010). Nevertheless, this criticism does draw reasonable attention to some weaker areas of the theory that deserve investigation and improvement. For this reason, a 2009 National Academy of

Sciences report called for additional studies into the repeatability and reliability of impression evidence findings, including attempts to quantify error rates (National Academy of Sciences, 2009). It is important to note that error rates for tool mark analysis will never be determined as accurately as those for DNA analysis (Chumbley et al., 2010). Unlike the well-characterized DNA genetic code, the population of tool marks is very difficult to quantify and always increasing with the production of new tools and marks. Moreover, the effects of many variables such as level of pressure, tool angle and motion, and the evolution of the tool surface with wear cannot be controlled or accurately measured. Therefore, it may be impossible to determine realistic error rates for tool mark analysis. However, tests with sequentially manufactured tools have the potential to provide useful approximations and/or bounds. Many recent research projects have focused on these types of experiments (Faden et al., 2007; Chumbley et al., 2010; Petraco et al., 2012a). This thesis attempts to extend these experiments in a meaningful way.

1.3 Proposed approach

The proposed methodology seeks to establish a more direct link between the tool and the evidence mark. If comparisons could be made directly between the tool and the evidence mark without the need for the intermediate marks in lead, the population under study would be greatly reduced, since it would consist only of the tool tips. This would make the establishment of meaningful error approximations or bounds simpler, since a link between a tool and mark could be verified against similar tools. Furthermore, this procedure could prevent damage to the tip caused during the marking process. Finally, the courts would likely be more amenable to the direct results.

With this goal in mind, the proposed methodology has the following procedural steps:

1. An optical profilometer is used to digitize the geometry of the tool tip and the evidence mark. The measurements are used to reconstruct the geometry inside the computer.
2. A simulation algorithm uses the tool tip geometry to make several simulated marks for comparison to the real mark. These marks are created with varying pressure, angles, and twists.

3. A statistical comparison algorithm is used to compare the virtual marks to the real mark. This algorithm will yield a meaningful numerical measure of the likelihood of the match.
4. Based on the likelihoods of the matches between the virtual marks and the real mark, the software will inform the examiner of the best matching angle, twist, and pressure. The examiner should then follow up with some manual verification of these results.

At this time, certain simplifications have been made to this procedure to result in a project of a reasonable size for a first attempt. First, the virtual marking simulation makes the assumption of complete geometry transfer between the tip and the plate. Pressure, material properties of the tool and plate, deformations, and the possibility of partial markings have been treated as insignificant for the time being. Future research may refine the simulation to handle these variables as needed.

Second, despite the capability of the simulation to make marks at any set of angles and twists, this thesis focuses on the results of virtual marking with variations only in one angle, the angle at which the tool is held relative to the horizontal plate. This assumption reduces the analysis to a reasonable size. Future work will look at the impact of other tool orientations.

Third, this thesis employs only flat-head screwdrivers and their marks. Screwdrivers were chosen for two reasons. First, screwdrivers make simple and distinct striated marks, which are relatively easy to compare and therefore good for initial testing. Second, the author's collaborators had used the screwdrivers in previous related studies and had already built up a knowledge base of working with them. In particular, they had researched and chosen an appropriate method for digitizing them. Screwdriver surfaces are at almost 90° to each other, making them difficult to image. These surfaces require a system capable of making accurate micro- and nanoscale measurements of surfaces at a high angle to the detector. A previous thesis had located such a system (Kidd, 2007). Given the relative complexity of the screwdriver head, this system should be able to accurately scan any other tool as well. Preliminary testing has partially confirmed this. Therefore, future work should be able to generalize the flat-head screwdriver results in this thesis to other tools.

Finally, this thesis focuses on Steps 1, 2, and 4 of the above proposed tool mark comparison

methodology. Step 3 uses the latest version of the statistical algorithm employed in Chumbley et al. (2010), which has not changed significantly in operation. The basic theory behind this algorithm will be discussed in Chapter 2.

1.4 Thesis overview

The following chapters will cover related work, the implementation and verification of the proposed methodology, conclusions, and future work. Chapter 2 will survey other attempts to reinforce tool mark analysis, both past and present, and include an introduction to the statistical algorithm employed in this research. Chapter 3 will discuss the methodology used for scanning and preparing tip and mark surfaces, assigning a marking coordinate system, simulating the virtual marks, and providing a user interface to the software. Chapter 4 will present the results of verifying the virtual mark simulator using standard geometry. It will also present the results of an initial study using both sides of 6 screwdriver tips and 34 corresponding plates. Finally, Chapter 5 will draw conclusions and discuss areas for future investigation.

1.5 Summary

The goal of tool mark analysis is to establish a link between an evidence mark and a suspect tool. In the traditional method, the forensic examiner makes known marks at many angles with the suspect tool and compares these to the evidence mark using a comparison microscope. The examiner uses a qualitative investigation to determine whether or not the tool made the mark. In response to recent criticism of this method, this thesis proposes a new method that attempts to link a screwdriver directly to its marks through the creation of simulated marks in the computer. Thanks to digitization of the geometry and a quantitative statistical comparison algorithm, this method shows potential for reinforcing the theory of tool mark analysis through the establishment of reasonable error approximations between tools. Moreover, the method has the unique potential to prevent damage to evidence and to save time for the examiner by focusing his or her efforts on verification of a smaller range of angles and twists of the tool.

CHAPTER 2. REVIEW OF LITERATURE

When surveying the relevant forensics literature, it is important to draw from both studies of tool marks and bullets. The discipline of firearms analysis, which seeks to identify which gun fired the bullet in question, bears a large amount of similarity to tool mark analysis. This is due to the fact that gun barrels leave similar striated marks on fired bullets due to their rifling patterns. There is often more literature available in firearms comparison because gun crime investigations have a higher profile than those involving tool marks (usually theft investigations). Therefore, this chapter includes relevant research from both disciplines.

This chapter presents a brief overview of developments in tool mark and firearms analysis. Section 2.1 discusses the state of the research approaches prior to the *Daubert* case. Section 2.2 discusses the advent of experimenting with 3D digitized geometry and numerical signatures to develop automated comparison systems. Finally, Section 2.3 briefly describes the current state of academic research with digitized geometry and numerical methods. Section 2.3 also describes the comparison algorithm used in this research, which was previously developed and tested with stylus profilometer data.

2.1 Research prior to *Daubert*

Research was indeed conducted prior to 1993 to examine error rates and determine the durability of markings. The following two papers summarize some of this research.

Biasotti and Murdock (1984) reveal that there was careful consideration of error in tool mark analysis prior to the *Daubert* case. While the authors confidently stated that it was possible for matches to be made to the exclusion of all other tools, they did not assume that their current-day practices guaranteed those types of conclusions in all cases. Instead, the

authors repeatedly cautioned examiners to do their own testing with known non-matches to see how similar they could appear. Such similarity arises due to sub-class characteristics, features commonly imparted to many different tools by the same manufacturing machinery. They cite two studies examining the frequency of occurrences of groups of consecutively matching striae in non-matching marks. They also cite a third study that used a mathematical model to correctly predict the number of matching groups of striae in a bullet data set. Therefore, it is clear from Biasotti and Murdock (1984) that concern over error rates and proper theoretical backing existed in a limited form prior to 1993 in the examining community.

Bonfanti and De Kinder (1999) summarize 10 studies performed between 1920 and 1989 on the effects of wear in a gun barrel on the similarity of bullets and cartridge casings fired from it in successive rounds. The authors conclude that the breech face impressions on the cartridge caused during firing are similar even after thousands of rounds and that the coarse striae on the sides of jacketed bullets also endure up to thousands of rounds. These impressions on the sides of bullets result from the passage of the bullet through the barrel, similar to the striated marks made by tools. Therefore, Bonfanti and De Kinder (1999) demonstrate that tool marks made by the same tool can show the same characteristics even after many subsequent tool uses.

From these two examples, we can see that tool mark analysis research is not completely new. While the *Daubert* case generated more public interest in the foundations of forensic examination, the forensic community formed theories, conducted experiments, and developed standards before then.

2.2 Preliminary research in 3D automated comparisons

In the early 2000s, the following tool mark and firearms research was conducted using 3D digitized geometry and numerical signatures to determine the feasibility of this type of approach for the reinforcement and improvement of forensic analysis. However, two of these studies tested their approaches on very limited data sets, and the third used a trade-secret commercial algorithm to compute their likelihood of match value. Therefore, this research was not particularly valuable for directly helping the tool mark analysis community. Nevertheless, it did demonstrate that 3D could prove useful for aiding that community in the future.

Geradts et al. (2001) used a commercial structured light scanner to digitize the geometry of gray casts of screwdriver tool marks in wax. A software program allowed the user to manually select comparison regions. A simple statistic formed from the standard deviation of the differences between two traces was used to indicate the likelihood of the match. Only six marks from six different screwdrivers were tested, and no error rates or matching conclusions were made. However, Geradts et al. (2001) demonstrated that a 3D optical profile of a striated mark could be retrieved and used for comparison, albeit with some artifacts due to the resolution of the commercial system. Moreover, the authors considered digitizing the screwdriver tip and comparing it directly to the evidence mark, the goal of this thesis. However, the authors did not pursue this line of investigation, likely due to the difficulty of digitizing the tip with a commercial structured light system. The high surface angles and reflectivity of the metal tip pose large problems for imaging systems.

Bachrach (2002) discussed the development of a new 3D commercial bullet characterization system, SciClopsTM. The system used a confocal laser sensor to detect a line of depth values from a rotating bullet. 3D data from two bullets fired with the same gun, once aligned, revealed very interesting features which reinforced certain assumptions made by firearms examiners. For instance, land engraved areas (LEA) matched more closely at the edges, and groove engraved areas (GEA) matched more closely in the middles. This agreed with the established firearms theory that bullets contact the edges of LEA and the middles of GEA. The two bullets' surfaces also exhibited the level of microscopic differences that forensic examiners had predicted given the variable nature of the bullet's travel during the firing of the gun. Finally, Bachrach (2002) examined six bullets, pairs of which were fired from three sequentially manufactured gun barrels. Preliminary results with a simple correlation measure showed promise for distinguishing the bullets from each other.

Roberge and Beauchamp (2006) used the commercial BulletTrax-3DTM system to successfully complete a bullet-matching test created by Evan Thompson, Washington State Police firearms examiner. The system takes 3D panoramic data of bullets and generates a database. Matches are made using a trade-secret numerical signature that has been demonstrated to not necessarily have statistical significance (Petraco et al., 2012b). Potential matches are listed in

a ranked order similar to a search-engine results page. Using known non-matches in the set, Roberge and Beauchamp (2006) computed a threshold of 650 in the signature number to distinguish between matches and non-matches without any estimated error. With this threshold, the authors matched the first six out of ten sets of four matching bullets. Excluding those matches (as instructed by the administrators of the test), they used agreement among bullets in the database rankings to find two more matches. Excluding those two, they were able to match the final two. All of the matches were made correctly. Nevertheless, the test had somewhat artificial conditions; ten sets of four bullets were guaranteed to come from the same gun each, two of them known to be from that gun and two of them unknown. Only two bullets came from a gun that was not in the set of 10 known guns. Moreover, the problem could be partially solved by elimination, which is not guaranteed in the real world. Finally, courts will not accept evidence from software with unknown comparison algorithms (Petraco et al., 2012b). Therefore, although Roberge and Beauchamp (2006) demonstrated that an automated system could correctly complete an examiner’s test by distinguishing between bullets in a limited population, the results do not significantly add to forensic theory.

2.3 State-of-the-art in automated comparisons

This section summarizes the current research in comparing digitized geometry using numerical methods. The experiments described here involve larger data sets than those in the previous section, allowing the researchers to draw more stable conclusions. Moreover, the approaches used contain more rigor and are more accepted in the scientific community. This section is divided into two different types of approaches. The studies in Subsection 2.3.1 developed statistical metrics for judging the likelihood of a match. The comparison approach used in this thesis is described in this subsection. Subsection 2.3.2 describes approaches that are instead based specifically in the discipline of numerical pattern recognition or machine learning. Both types of approaches show promise for distinguishing matches and have results that reinforce the theory of tool mark examination.

2.3.1 Developing statistical metrics

Faden et al. (2007) compared marks in lead from 44 sequentially manufactured screwdriver tips using a simple statistical algorithm. In this study, two replicated marks were made at 30° , 60° , and 85° angles of attack for both sides of each tip. A surface profilometer was used to measure a trace across each mark. The algorithm compared two mark traces by computing the correlation between a pair of small data windows drawn from each trace. The maximum correlation over the length of the traces served as the measure of the likelihood of the match. Faden et al. (2007) concluded that maximum correlation did not make a good metric for comparison since window pairs could sometimes have high correlation without the remainder of the patterns actually matching. This shortcoming was especially apparent when two partial markings were compared. Nevertheless, the algorithm could sometimes use the maximum correlation metric to distinguish between known matches and known non-matches provided that the matches were made at the same angle and using the same side of the tip. This supported the assumptions among forensic examiners that matches could only be made at similar angles and that different sides of the tip behaved as different tools.

Chumbley et al. (2010) extended the work in Faden et al. (2007) by using an improved statistical algorithm and an extended data set of 50 tools, four replicate marks at each angle, and 10 profilometer traces per mark. The statistical algorithm used in Chumbley et al. (2010) was the same one used to make the comparisons in this thesis. For this reason, a brief summary of its operation will be provided here.

The statistical algorithm starts with the Optimization step. Like the statistical algorithm in Faden et al. (2007), the algorithm finds the pair of windows in the two data traces which exhibit the largest correlation value. These windows are the connected solid black window pairs shown in the three comparisons in Figure 2.1. To solve the problems experienced in Faden et al. (2007), the algorithm then performs the Validation step, which has two distinct parts. The first part is depicted in the dashed and dotted windows in Figures 2.1(a) and 2.1(b). The algorithm randomly chooses a rigid distance from the maximum correlation windows and places a new pair of windows there. If the profiles really do match, the correlation between these rigid-shift

windows should also be quite high. Figures 2.1(a) and 2.1(b) only depict two sets of these rigid-shift windows, but in this thesis, 50 pairs of these windows were used.

Figure 2.1 The origins of the T1 statistical measure. (a) and (b) Rigid-shift and maximum correlation windows. (c) Random-shift and maximum correlation windows. Reprinted from Chumbley et al. (2010) with permission.

With only the correlations from the rigid-shift pairs, the algorithm still has no standard against which to compare the results. In other words, there is still no way to know if the correlations truly indicate a match or not. Therefore, the algorithm moves onto a second part of the Validation step. In this part, pairs of random-shift windows are formed. In each pair, the windows are located at a different random shift from the maximum correlation windows. These pairs are depicted by the dashed and dotted and dash-dotted windows in Figure 2.1(c).

The correlations between these windows represent the likelihood of a match happening by pure chance. Like the rigid-shift windows, 50 of these random-shift windows were employed in this thesis.

Once the two steps are complete, the algorithm combines the correlations from the rigid-shift windows and the random-shift windows into a nonparametric Mann-Whitney U-statistic referred to as a “T value.” The details of computing this statistic are beyond the scope of this thesis, so they will not be included here. However, a simple ratio analogy can be used to partially understand the significance of the T value. Consider the ratio $R_{\text{rigid}}/R_{\text{random}}$, where R_{rigid} is an average of the correlations from the rigid-shift windows and R_{random} is an average of the correlations for the random-shift windows. If the two profilometer traces are a true match, then R_{rigid} should be higher than R_{random} , and their ratio should be a number larger than one. If on the other hand, the traces are not a true match, R_{rigid} and R_{random} should be about the same, yielding a value of one. While the actual T value is like this ratio in that it is above one in the case of a match, when there is not a match, it can actually take a negative value. Therefore, T is generally large and positive in the case of a match and small or negative in the case of a non-match.

Using this algorithm, Chumbley et al. (2010) found results consistent with the assumptions made by tool mark examiners. The T values produced when marks from different screwdrivers were compared clustered around zero, regardless of the angles of attack used to make the marks. Similarly, T values from comparisons of marks made by different sides of the same screwdrivers also clustered around zero. Marks from the same sides of the same screwdrivers produced mostly high T values when they had the same angle of attack but produced T values clustering around zero when they had angles of attack that differed. These trends reinforced the assumptions that each screwdriver had its own unique mark, that different sides of the same screwdriver have their own unique marks, and that marks only match at a similar angle. Nevertheless, when the full spreads and outliers of the T values were considered, these trends were not as clear. Ultimately, Chumbley et al. (2010) estimated the error rates for the algorithm at 1-3 false positives, 8-9 false negatives, and 1-3 inconclusive results for every 100 comparisons. The number of false and inconclusive results varied with the angle of attack used to produce

the mark, suggesting that mark quality depends on angle.

Chumbley et al. (2010) then verified the algorithm’s performance against comparisons from forensic examiners of various training levels at the 2008 Association of Firearm and Toolmark Examiners Training Seminar. The 50 volunteer participants rated 20 comparison pairs, five of which were known matches misidentified by the software as non-matches and another five of which were known non-matches misidentified as matches. Although the study had some different constraints than the participants were used to in their respective laboratories, the examiners outperformed the software, reporting no false positives. Moreover, only 11 out of 126 of the examiners’ comparisons were false negatives. (Note: Examiners are trained to render a positive identification only in the complete absence of doubt.) From this, Chumbley et al. (2010) concluded that the algorithm’s performance could be improved with contextual information from the examiner.

Nevertheless, the algorithm of Chumbley et al. (2010) showed promise. In particular, it was a reasonable option for quantifying and automating the comparison process, since forensic examiners could use their contextual knowledge to correctly interpret the results. For this reason, it was chosen as the comparison algorithm for this thesis.

2.3.2 Utilizing numerical pattern recognition

Numerical pattern recognition, also known as machine learning, uses a slightly different approach to identify tools and/or firearms. These methods are in general “trained” on an initial data set to distinguish between different tools by some form of numerical signature derived from that tool. The identity of additional specimens can then be inferred. These methods support the validity of the examiner’s experience in being able to distinguish the identities of tools.

Petraco et al. (2012a) demonstrated that several common numerical classification algorithms could have success at distinguishing between striation patterns made with different screwdrivers. In this study, 75 marks were made by hand on modeling clay using nine screwdrivers oriented at approximately 90° to the clay surface. These marks were digitally imaged with a 2D camera. The images were sampled horizontally with 121 equally spaced samples which were thresholded to binary, yielding a black and white one-dimensional barcode-like

image of the striation. Partial Least Squares Discriminant Analysis (PLS-DA) and Principal Component Analysis (PCA) were used to reduce the dimensions of the striation barcodes to those dimensions with the most influence on the striation pattern; this is essentially the generation of a unique “signature” for the pattern that can be used to identify it relative to other patterns. PLS-DA and Support Vector Machines (SVM) were trained to distinguish between these signatures so as to classify which signature belonged to which screwdriver. Finally, Hold-One-Out Cross-Validation (HOO-CV), bootstrapping, Conformal Prediction Theory (CPT), and random data set tests were used to estimate the error rates of the classification. Petraco et al. (2012a) concluded that PLS-DA was able to distinguish tool marks with a 3% error rate using only eight dimensions of data and that PCA-SVM was able to achieve similar performance with only four dimensions. Moreover, CPT was used with PCA-SVM to yield comparisons with 95% confidence intervals 90% of the time.

Bolton-King et al. (2012) was unique in that it actually addressed characteristics of the tool instead of its marks. The study involved making plaster casts of the inside of gun barrels and cross-sectioning them in three places. Close-up images were then taken of the right and left transitions between lands and grooves and thresholded to binary to emphasize the edge of the transition. PCA was used to compute eigenimages from the 2D fast Fourier transforms of these binary land transition images. Using a Euclidean distance metric with weights on the individual components to distinguish between patterns, Bolton-King et al. (2012) demonstrated the ability to statistically differentiate between different makes of gun barrels, even those with very similar land transitions from similar manufacturing processes. The comparisons exhibited a false negative rate of 1.7% and the same false positive rate. This study is relevant to this thesis because it presents another analysis of a tool (gun barrel) instead of focusing purely on its marks (bullet land engravings). It also demonstrates success in using numerical methods of comparison on bulk (class) features in addition to the success shown for unique microscale features in other studies.

2.4 Summary

While research in tool mark analysis is nothing new, recent approaches have utilized more established and rigorous numerical methods. Moreover, using 3D instead of purely 2D data allows these techniques to more accurately represent the true nature of the surface geometry of a mark. Both of these characteristics make the results of the new methods more likely to satisfy judges and juries in the legal system. Finally, the statistical metrics and numerical pattern recognition approaches have the potential benefit of automating portions of the examiner's analysis, reducing the time required for processing. This will help reduce workloads for the often overtaxed crime labs.

CHAPTER 3. METHODS AND PROCEDURES

As demonstrated in the above chapters, virtual tool mark generation is but one step in a larger process of tool and mark evidence acquisition, analysis, and comparison. Therefore, this chapter will describe not only the virtual marking algorithm but also the steps leading up to and coming after the virtual mark itself. Section 3.1 will discuss the 3D optical profilometry process used to digitize the tips and plates, and it will also discuss the methods used to reduce the noise artifacts and assign a meaningful coordinate system for generating marks. Section 3.2 will focus on the actual algorithm used to generate virtual marks from the tip geometry, and Section 3.3 will discuss the unification of all of the algorithms in a graphical user interface for forensic examiners. Finally, Section 3.4 will summarize the overall process.

3.1 Data acquisition and preparation

This section will cover preparing both the tips and plates for the virtual marking software. Subsection 3.1.1 will discuss the rationale behind our choice of scanning system. Subsections 3.1.2 and 3.1.3 will discuss the algorithms used to pre-process the data to reduce noise on tips and plates, respectively. Finally, Subsection 3.1.4 will discuss how coordinate systems are applied to the tips and the plates.

3.1.1 Geometry acquisition

The Alicona Infinite Focus Microscope (IFM), shown in Figure 3.1, was used to digitize the 3D geometry of both the tips and the tool marks. Taylor Grieve, master's student in Materials Science and Engineering under the advisement of Dr. L. Scott Chumbley, operated this system for the research. The IFM captures the surface topology using an operating principle known as focus variation or depth from focus/defocus (Alicona Imaging GmbH, 2012; Bolton-King et al.,

2010). This technique is based on the focal plane inherent in any optical system. Objects at the focal distance from the optical detector will be in sharp focus in the resulting image, whereas objects closer or farther away from the detector will be slightly blurred in the image. In the Alicona IFM Model G3, the sample is placed on a precision stage that mechanically moves it up and down (in and out of focus) relative to the detector. Tracking the sharpness of the features and the displacement of the stage allows the IFM to measure the height of the features along with their 2D color texture.

Figure 3.1 The Alicona Infinite Focus Microscope (IFM) Model G3 used in this research.

The IFM was chosen for investigating tool tips early in 2008. For her master's thesis research, Julie Kidd (2007) tested the ability of a wide variety of 3D measurement systems to capture the geometry of screwdriver tips. Initially, she used the Alicona MeX triangulation software to compute the depth from pairs of Scanning Electron Microscope (SEM) images of the tips taken at varying angles. However, these measurements contained too much noise for forensic analysis. The stylus profilometer had difficulty measuring along the edge of the tip and required physical contact with the tip for measurements. (Contact is undesirable since it potentially alters the evidence.) The laser profilometer had difficulty staying appropriately focused on the sample, and therefore its measurements also contained a large amount of noise.

Penetrating dye, confocal florescence, and x-ray tomography all failed due to inability to resolve the tool tip microscale features responsible for producing the striations. Representatives at Alicona were able to successfully capture the surface patterns on one of Kidd’s screwdriver tips using the IFM, which played a large role in its selection.

Moreover, at that time the IFM outperformed the 3D confocal microscope in its ability to measure steep angles. Bolton-King et al. (2010) demonstrated that the confocal microscope, unlike the focus-variation technique, could not detect the transitions between the land-engraved areas and groove-engraved areas on the NIST standard bullet. The confocal technique at the time was limited to a maximum surface angle of 70° due to its numerical aperture, whereas the focus-variation technique could achieve measurements of surfaces at close to 90° to the detector. Similarly, the confocal technique would have had difficulty measuring the approximately 90° angle of the screwdriver tip. For this reason, the IFM was chosen to perform this research. In the future, as Bolton-King et al. (2010) inferred, the confocal technique may improve and become a better choice for this type of research.

3.1.2 Tip noise reduction

Although the IFM takes some initial steps to reduce noise in the output data, spike noise often remains on the screwdriver tip that would interfere with virtual marking. Figure 3.2(a) presents a screwdriver tip scanned by the IFM without any additional lighting and prior to any algorithmic cleaning. The geometry is almost entirely hidden by the spikes, which are IFM sensor artifacts arising from imaged areas that are too dark or too bright. Virtual marks made with this tip will only contain meaningless striations from the spikes. Therefore, we must remove these spikes.

Two approaches have been used to reduce this type of noise. First, an algorithm was developed to detect and remove the spikes, replacing them with interpolated regions. Each column of the screwdriver data is fitted with a seventh-order polynomial. Any point with a depth value 100 micrometers or more different from the depth value predicted by the polynomial fitting is stored as a 0 value in the mask. (A mask is a binary memory buffer the size of the screwdriver data set that is used to remember points which are “bad.”) This is then

Figure 3.2 Spike noise removal. (a) Tip digitized with no additional lighting. (b) Tip in (a) after algorithmic cleaning. (c) Tip digitized with new lighting procedure, prior to algorithmic cleaning.

repeated with the rows of the data, again using seventh-order polynomials and a 100 micrometer threshold. A fast connected components algorithm is then applied to remove islands of noise from around the main bulk of the screwdriver data. Finally, a hole-filling algorithm is used that fills gaps in the depth data. This algorithm first goes down the columns and locates spans of empty pixels inside of the main bulk. If the span contains less than 20 empty pixels, linear interpolation between the good pixels at either end of the span is used to generate substitute data, and those new points are masked in. This same treatment is applied across the data rows. Spans of 20 empty pixels or more in length are not filled in to prevent large scale tampering with the integrity of the screwdriver data.

Figure 3.2(b) shows the results of this cleaning algorithm for the tip shown in Figure 3.2(a). The algorithm successfully removed the spikes, but the remaining surface geometry seems distorted into terraces. We expect the screwdriver tip to be smoother than this at the microscale level.

Therefore, a new tip scanning procedure was developed to improve the quality of the raw data. Four fiber optic lighting cables, each pair lit by a 150 W white light source, are equally positioned around the screwdriver tip during scanning to provide more lighting. Figures 3.1 and 3.3 show these additional lights. The equal positioning discourages specular highlights which would result in spikes. The gamma (nonlinear) response of the optical sensor in the IFM is also adjusted to 0.50-1.00 to reduce the amount of under- and over-exposed pixels in the 2D IFM images. Since the IFM uses focus variation in these 2D images to compute depth, this gamma adjustment helps ensure better quality 3D data. Likewise, the Alicona automatic color correction tool is used to adjust the 2D image quality prior to scanning. Finally, an initial scan of the tip at 5x magnification is taken. If spikes exist in this scan, minor adjustments to the positioning and intensity of the fiber optics cables are made, and the tip is rescanned at 5x magnification. This is repeated until the 5x scan is devoid of severe spike noise. Then, the final scan at 10x magnification is taken.

Figure 3.3 Close-ups of the tip lighting in the IFM, showing the four fiber optic cables.

Figure 3.2(c) presents a representative tip scanned with the new lighting procedure, prior to any algorithmic cleaning. The bulk of the data has no spikes and appears smooth as expected. There is only some spike noise at the ends that requires removal. Figure 3.4 presents alternative views of the data in Figure 3.2(c). Figure 3.4(a) shows the 2D screwdriver texture (shown in Figure 3.4(b)) mapped onto the 3D data. With reference to these figures, we can see that

this end noise occurs where the screwdriver geometry falls sharply away from the detector. In particular, the spike noise in Figure 3.4(a) is very dark in comparison to the light texture of the screwdriver, and from visual inspection of the 2D texture in Figure 3.4(b), it appears that this dark area is at the edges of the screwdriver tip. Since this sharp drop is not helpful for the forensic analysis and could confuse the statistical algorithm, we will simply remove it.

Figure 3.4 Tip 8 Side A, prior to algorithmic cleaning. (a) Texture overlaid on 3D geometry. (b) 2D texture. (c) Alicona-computed quality map.

To remove this end noise, we can compute a mask based on the texture and the quality map. Figure 3.4(c) shows the Alicona IFM's quality map, a metric displaying the quality of the measurement; high quality regions are dark and low quality regions are light in this image.

Comparing this to the 2D texture, it is evident that the quality map also well defines the edges of the screwdriver. Pixels in the quality map with 8-bit grayscale values of over 200 and pixels where the maximum of the texture’s red, green, and blue 8-bit values was less than 20 were turned off in the mask (thresholded). A standard connected components algorithm was then used to find the largest single mass of on values in the mask; all other on values were turned off to prevent little islands of noise from appearing around the edge of the screwdriver. Finally, a 20 pixel border was removed from the mask to ensure that no leftover edges (and their accompanying spikes) remained.

Figure 3.5 presents the results of this noise removal process for the data set of Figures 3.2(c) and 3.4(a). These results are representative of end noise removal. As the figure demonstrates, the tip ends have become acceptably clean for forensic analysis.

Figure 3.5 Tip 8A, after cleaning. Texture-mapped view.

In summary, the current cleaning techniques prove largely successful at mitigating significant noise. The results seen in Figure 3.5 are representative of the screwdriver scans used, which were all taken with the new lighting procedure. The remaining few small spikes should have an insignificant effect on the statistical matching algorithm.

3.1.3 Plate noise reduction

Cleaning the marked plates is much simpler than cleaning the tips. The mark geometry is flat, so the resulting scans contain less noise. Figure 3.6 presents three representative plates

before and after cleaning. Prior to cleaning, there are some dark regions at the edges of the plates which are the unmarked parts. The plate cleaning algorithm uses thresholding of the 2D texture and the Alicona-computed quality map to mask off points that contain spikes due to poor imaging quality. It then examines the x and y gradients of the 2D texture and uses these to determine where the left and right edges due to the unmarked plate are located. These left and right edges are masked off, too. The connected components algorithm is used to ensure that the remaining plate is one solid object, and the depth values are median filtered to remove spikes. Finally, a plane is fitted to the plate and then subtracted from the plate data. This de-trending operation is necessary for the statistical comparison algorithm.

Figure 3.6 Plates before ((a)-(c)) and after ((d)-(f)) noise reduction. (a) and (d) Plate marked by Tip 25 Side A at 45° , scanned at 5x magnification. (b) and (e) Plate 8A 45° at 10x. (c) and (f) Plate 44B 60° at 10x.

As Figure 3.6 reveals, the resulting plate data is acceptably clean. The dark edges have been correctly removed, and no spikes are present. The algorithm functions satisfactorily.

3.1.4 Coordinate system assignment

At this point, the cleaned data of the tip measurement is defined in terms of the coordinate system of the IFM. In order to meaningfully compare virtual marks made with this tip to

marks generated in the real world, we need to give the virtual tip a new coordinate system derived from the real-world marking coordinate system. Figure 3.7 illustrates the process used to generate tool marks in lead. The screwdriver is placed in the jig shown in Figure 3.7(a), which constrains the screwdriver to move in the x direction of the coordinate system shown in Figure 3.7(b). Care is taken when tightening the black thumbscrew to align the tip edge with the direction of the y axis. The design of the jig ensures that the screwdriver shaft is held at angle α about the y axis during the marking process. The jig in Figure 3.7(a) defines an angle $\alpha = 45^\circ$; other similar jigs are also used that define the angles $\alpha = 30^\circ$, 60° , and 85° .

Figure 3.7 The marking process. (a) The 45° marking jig with fitted screwdriver handle and tip. (b) Marking diagram.

To define the position of the tip relative to the world coordinate system, we rigidly assign the coordinate system shown in Figure 3.8 to the tip geometry. Figure 3.8(a) shows the orientation of the tip when the angle α is 0° . Figures 3.8(b) and 3.8(c) show the tip's orientation at $\alpha = 60^\circ$ and 85° , respectively. It can be seen from these figures that the y axis of the tip coordinate system lies along the edge of screwdriver tip so that a rotation about the y axis is a rotation about the edge. This makes sense since the edge is what will contact the lead plate and serve as the pivot. The x axis is made to point along the shaft of the screwdriver; this

corresponds to the fact that the jig holds the shaft (and not the flat side of the screwdriver) at the angle α . Finally, since the angle between the shaft and the flat bottom of the screwdriver is approximately 90° , the z axis will point along this flat bottom. Figure 3.8(d) reveals that the coordinate system is centered along the screwdriver edge. This was chosen mainly out of convenience, since for now only variation in y -angle rotation is under study. At this time, it is suspected that the exact origin of coordinates is not as important as the orientation of the axes, since translations in the mark location can be identified in the computer and removed.

Figure 3.8 Tip coordinate system. (a)-(c) View of tip side. (d) View looking at the flat side of the screwdriver.

To make virtual marks that correspond to the real-world marks, we must affix the coordinate system of Figure 3.8 to the measured tool geometry from the IFM, redefining the geometry in terms of this new coordinate system. This is easily done by computing a linear transformation between the two coordinate systems and applying this to the geometry during the marking process. The algorithm for computing this transformation proceeds as follows:

1. *Identify the screwdriver edge.* This is accomplished by sampling approximately 250-300 rows evenly spaced throughout the height of the tip data and recording the points in those rows with the maximum depth. Figure 3.9 illustrates this process, depicting the selected rows with red arrows and the recorded points with red x 's. Only 250-300 rows are selected since this is enough samples to reasonably characterize the edge. Taking more samples would only slow the process down without a significant increase in accuracy. This process assumes that the data set is indeed that of a flat screwdriver head and that the head was fixed relative to the IFM as depicted in Figure 3.3, at an angle of 45° to the detector. To avoid having noise along the boundaries of the data set mistaken as the tip edge, any points within 10 pixels of the edge were not recorded.

Figure 3.9 Process of locating the edge points.

2. *Find the center of coordinates.* The points identified in the previous step are averaged to compute the geometric centroid of the edge, $\bar{\mathbf{v}} = [\bar{x} \ \bar{y} \ \bar{z}]^T$. The translation matrix to this point, M_T , shown in Equation 3.1, is recorded for later.

$$M_T = \begin{bmatrix} 1 & 0 & 0 & -\bar{x} \\ 0 & 1 & 0 & -\bar{y} \\ 0 & 0 & 1 & -\bar{z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

3. *Determine the direction of the y axis.* In this step, a line is fit to the previously identified edge points using linear regression. The result is a y axis vector, which is normalized to unit length and saved for later.
4. *Determine the direction of the x axis.* Figure 3.10 illustrates the derivation of the x axis direction from the measurement jig. As revealed by Figures 3.1, 3.3, 3.10(a), and 3.10(b), the tip jig consists of an adjustable knob with a hole in it for the screwdriver tip shaft. The angles of rotation are labeled on the main body of the jig, and the jig body fits snugly against a ledge on the IFM stage during measurement. For all of the tip data, the shaft angle used for measurement was 45° . Figure 3.10(c) shows the IFM coordinate system; if the jig is snug against the stage ledge, the screwdriver shaft will lie in the IFM xz plane. When this is satisfied (which it is for the tip scans used in this research), the x axis can be computed by rotating the vector $[001]^T$ through a 45° angle about the direction of the IFM y axis (that is, $[010]^T$) and then negating it.

Figure 3.10 x axis determination. (a)-(b) The screwdriver tip jig. (c) Diagram of the jig during IFM measurement.

5. *Determine the direction of the z axis.* The z axis vector is computed as the normalized cross product of the x axis and the y axis.
6. *Ensure orthonormality.* Because of the cross product, the z vector is guaranteed to be normal to the xy plane. However, the x and y vectors are not guaranteed to be normal to each other. In particular, the edge of the screwdriver may not be exactly perpendicular

to the shaft due to wear and limitations on the accuracy of the manufacturing process. In this case, it is assumed that the screwdriver marking motion will be more constrained by the edge of the screwdriver dragging along the plate than the direction of the shaft in the jig or a person's hand. Therefore, the x axis direction is corrected to be normal to the y axis direction by re-computing the x axis as the normalized cross product of the y axis and the z axis.

7. *Compute the basis change matrix.* The basis change matrix M_B is constructed as shown in Equation 3.2. Here, $[x_x \ x_y \ x_z]^T$, $[y_x \ y_y \ y_z]^T$, and $[z_x \ z_y \ z_z]^T$ are the x , y , and z axis vectors computed above, respectively.

$$M_B = \begin{bmatrix} x_x & y_x & z_x & 0 \\ x_y & y_y & z_y & 0 \\ x_z & y_z & z_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

8. *Compute the coordinate system matrix.* The coordinate system M_C is computed according to Equation 3.3.

$$M_C = (M_B)^{-1} M_T. \quad (3.3)$$

Note that M_B must be inverted so that M_C converts coordinates from the IFM coordinate system to the tip coordinate system rather than the reverse. Since the x , y , and z vectors are orthonormal, we can always invert M_B .

Therefore, M_C as computed using the algorithm above is a linear transformation from the original IFM coordinate system used in the data file to the new tip marking coordinate system. The resulting coordinate system applied to actual representative data is shown in Figure 3.11. The light-colored solid lines denote the tip coordinate system, and the darker dashed lines denote a local world coordinate system. For both of these, red denotes the x axis, green denotes the y axis, and blue denotes the z axis. Overall, the coordinate system appears correct.

Figure 3.11 The coordinate system transformation applied to the data. (a)-(c) The side view with y axis rotations of (a) 0° , (b) 60° , and (c) 85° . (d) View of the flat side of the screwdriver with a rotation of 0° .

3.2 Virtual tool mark generation

The basic principle behind the tool mark simulation is to take the projection of the tip geometry in the direction of tool travel and identify the highest points on that projection. The highest points will scrape the deepest into the plate material, so they are responsible for leaving the observed striae. Figure 3.12 gives an illustration of this idea. In this diagram, the screwdriver is held at an angle of about 45° with respect to the plate surface (a y rotation) and twisted about its shaft (x axis) by some angle β . The inset shows the measured tip geometry, resembling an extruded chevron, making the mark. The black wavy line down the middle represents the tip geometry closest to the plate. These points dig into the plate material, imparting the striae. Because the tip is at an angle β to the direction of motion, the depth

profile of the mark is the projection of the black wavy line onto a line at the angle β . From the diagram, we see that we could have found this mark profile by squishing the screwdriver tip onto a plane perpendicular to the direction of tool travel. The lowest points on this squished tip would be the ones that generate the mark and the same ones participating in the black wavy line. This type of collapsing process will be used to make virtual mark profiles.

Figure 3.12 The mark as a projection of tool tip geometry in the direction of tool travel.

Clearly, this approach is quite simple and ignores several complexities of mark making including material properties of the tool and plate, forces, deformations, and the possibility of partial markings. These phenomena can be very difficult to control and account for in a simulation. Since this is a first approximation, the assumption of complete geometry transfer from the tip to the plate is made. Hence, any effects of the specific forces and deformations are treated as insignificant since the geometry seems likely to be the strongest signal in the mark.

Although this method is quite simple in principle, the scale of the geometry simulation required makes it complex to carry out on the computer. Each tip data set at the 10x magnification level used in this research contains more than $9000 \times 1000 = 9$ million points. Projecting and determining the edge of such a large amount of data on the CPU would take a long time. Therefore, the Open Graphics Library (OpenGL) and its OpenGL Shading Language (GLSL) are used to build the simulation in order to take advantage of the parallel capabilities

of the computer’s Graphics Processing Unit (GPU). Both OpenGL and GLSL are described in Shreiner et al. (2007). Dividing the geometric computations among the shaders in the GPU drastically reduces the computation time required and makes rapid virtual mark generation feasible.

The following basic process was used to implement this simulation in OpenGL:

1. A rotation matrix M_R is formed to describe the desired angular position of the tip relative to the local world coordinate system. The user specifies a set of three angles (α , β , γ) in degrees which are the desired rotations about the x , y , and z axes, respectively. Equation 3.4 is used to compute M_R . Note that since the rightmost matrix is the x rotation, it will get applied first to each vertex. The y rotation will be applied next, and the z rotation will be applied last.

$$\begin{aligned}
 M_R &= R_z R_y R_x \\
 &= \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 & 0 \\ \sin \gamma & \cos \gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \beta & 0 & \sin \beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \beta & 0 & \cos \beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.4)
 \end{aligned}$$

2. The tip’s bounding box is rotated into the desired position using M_R . The “squish” matrix, M_S in Equation 3.5 below, is then applied to the bounding box to project it onto the plane normal to the local world x axis (the direction of tool travel). This matrix sets the x coordinates of each vertex to zero. The eight points of the bounding box are then iterated over to find the maximum and minimum y values taken up by the squished bounding box. This is a quick way to estimate the height of the tip projection in the scene, which will tell us how tall to make the scene window and the depth camera.

$$M_S = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

3. A one-dimensional depth camera is created, which we will use to find maximum points along the flattened edge of the screwdriver tip as illustrated in Figure 3.13. The number of pixels in this camera is carefully chosen to achieve the desired sampling resolution for the edge.

Figure 3.13 The depth camera concept.

4. OpenGL is directed to apply M_C , M_R , and M_S , in that order, to the tip data when it is drawn. The camera is also positioned at $5000\ \mu\text{m}$ from the center of coordinates so the whole tip can be imaged, and it is set up to use orthographic projection (meaning that it becomes a perfect non-pinhole camera that never views things with the distortion of perspective).
5. The tip geometry is drawn in the scene.
6. The sampled tip geometry is pulled out of the camera. It is converted from OpenGL camera coordinates back to μm and rotated 180° to become an impression. The rotation is performed by multiplying the mark by -1 and then flipping it from left to right.
7. An edge detection algorithm is used to clip off the parts of the mark due to the sides of the screwdriver tip. This is performed since the sides of the screwdriver can sometimes

confuse the statistical comparison algorithm. The sides are not in general part of the striae used for matching.

Section 3.2.1 will elaborate on the creation of the depth camera, and Section 3.2.2 will discuss special considerations for drawing the large number of tip vertices. Section 3.2.3 will explain the mathematics behind the conversion to μm , and Section 3.2.4 will discuss the edge detection algorithm.

3.2.1 Depth camera

OpenGL can be thought of as a simulation of taking a 2D picture of a 3D scene with perfectly known vertices. In OpenGL, we can create a special virtual camera that looks straight down upon the flattened edge of the screwdriver tip data as illustrated in Figure 3.13. This camera records the distance to the points that it sees rather than their color; it is a depth camera formed from an OpenGL object known as a depth buffer. In depth test mode, OpenGL uses an array called the depth buffer to keep track of whether or not a certain point in a graphics scene is obscured by another one. In order to get drawn by OpenGL, each pixel-sized element sampled from the scene (referred to as a fragment) must survive the “depth test.” OpenGL starts with the depth buffer initialized to the back limit of the camera’s view (the far clip plane). As it receives the fragments, it records their distance from the camera in the depth buffer. If a new fragment is received that belongs in the same pixel as an older one, it checks the depth of the newcomer. If the depth value is less than that of the old one (meaning that the new fragment is closer to the camera), that new fragment replaces the old one. Its depth value is written into the depth buffer for the next round of the depth test. There are several ways to direct OpenGL to return this depth buffer instead of a color image. Therefore, it can become a depth camera, returning the depth values of those points closest to it.

Since the data samples needed are one-dimensional, the depth camera is created as single column. A two-dimensional depth camera would simply record extra scene background points that would have to be filtered out. The number of pixels h needed to sample the tip geometry at the desired resolution δ is computed according to Equation 3.6, where y_{Delta} is the height

computed from the bounding box in Step 2 of the overall procedure above.

$$h = \text{floor}\left(\frac{y_{\text{Delta}}}{\delta}\right) + 1 \quad (3.6)$$

Equation 3.7 readjusts y_{Delta} so that δ is assured to be the desired value.

$$y_{\text{Delta}} = h\delta \quad (3.7)$$

To create this special one-dimensional depth buffer of the correct size without interfering with the on-screen window, an OpenGL framebuffer object (FBO) is used. A framebuffer object is like a bundle to collect various images for OpenGL to draw into instead of the standard window. The FBO in this research contained only one image: the $1 \times h$ depth buffer for the depth camera. Originally, the depth buffer was created as a renderbuffer object; however, ATI graphics cards would not accept an FBO with only a renderbuffer in it. Therefore, the depth camera had to be a 2D texture object.

The maximum allowable height of the depth camera 2D texture object in the FBO depends upon the capability of the computer's GPU. For a 10x resolution data set, $\delta = 0.803568 \mu\text{m}$, which causes h to be about 8000 when the tip is at the angles $(0, 0, 0)$. Many typical graphics cards cannot handle a texture this tall. Therefore, the flattened screwdriver scene is automatically divided into equal vertical partitions as needed to satisfy the demands of the graphics card. The depth camera then takes an image for each one of these partitions, and these images are later stitched together into a complete virtual mark. Equation 3.8 is used to compute the height of the depth camera for the case of multiple partitions, where n is the number of partitions and y_{max} and y_{min} are the extreme values of the collapsed bounding box in Step 2, depicted in Figure 3.13. The y_{Delta} parameter is still corrected with Equation 3.7, but instead of being equivalent to $y_{\text{max}} - y_{\text{min}}$ as shown in Figure 3.13, it is now only equal to the height of the depth camera (which only covers a fraction of the mark) in μm .

$$h = \text{floor}\left(\frac{y_{\text{max}} - y_{\text{min}}}{\delta n}\right) + 1 \quad (3.8)$$

Finally, although the FBO with the depth texture is the major component of the depth camera, the OpenGL viewport and projection matrix need to be correctly set. The viewport

(the size of scene on the computer screen) is set to $1 \times h$. The orthographic projection matrix is set such that the visualized part of the scene is y_{Delta} μm tall. In the first scene partition, the lower clip plane is set to y_{min} and the upper clip plane to $y_{\text{min}} + y_{\text{Delta}}$. For subsequent partitions, y_{Delta} is added to these clip planes to advance the part of the scene that the depth camera captures.

3.2.2 Drawing the tip

As mentioned above, the average digitized tip in this research contains slightly more than $9000 \times 1000 = 9$ million points. For OpenGL to draw these points, it must transfer at least three floats (floating point numbers) representing x , y , and z to the graphics card for each vertex. This amount of information takes up $(4 \frac{\text{bytes}}{\text{float}})(3 \frac{\text{floats}}{\text{vertex}})(9 \times 10^6 \text{ vertices})(\frac{1 \text{ kB}}{1024 \text{ bytes}})(\frac{1 \text{ MB}}{1024 \text{ kB}}) \simeq 100 \text{ MB}$ of memory. There is a limit on the rate of transfer from the CPU memory to the GPU, so in the ideal case, we would store this vertex data in a vertex buffer object (VBO) on the GPU to avoid transferring it each time the mark is drawn. However, the GPU only possesses a certain amount of RAM. While higher-end GPUs have gigabytes of RAM, many existing consumer-level GPUs do not. Therefore, we decided to conservatively limit the GPU RAM usage to 128 MB, a minimum system recommendation for modern video games. With approximately 60 MB of the GPU RAM already devoted to showing down-sampled 3D representations of the tip and mark in the user interface, the mark vertex data needed to be streamed to the graphics card rather than stored.

At first, the OpenGL `glVertex` command was used for streaming. However, most implementations of OpenGL could not handle this much streaming data and reserved several gigabytes of CPU RAM as a buffer for the data during streaming. This inhibited operations of other programs during mark generation. The mark-making process also ran slowly. Therefore, we created a special object called a `StreamBuffer` to stream the vertices to the GPU instead of OpenGL. The `StreamBuffer` allocates a pointer to 30 MB of CPU RAM and creates a 30 MB VBO on the GPU. The mark generation program hands its vertices to the `StreamBuffer` object, which stores them in the CPU pointer. When the CPU pointer becomes full, the `StreamBuffer` copies them to the VBO and instructs OpenGL to draw them. The data is initially stored

in the CPU memory as a backup for the GPU RAM; the process of editing GPU RAM can sometimes fail and need to be repeated. A dynamically allocated CPU array was used to avoid the function overhead associated with vector objects. For normal operations, this function overhead is not noticeable, but for the intense streaming operation, it slowed mark generation by several seconds overall on a desktop PC.

Moreover, in order to generate the virtual mark at the desired resolution δ , we will need to interpolate between the data points. OpenGL will automatically perform this interpolation if the data is appropriately formatted in a triangle mesh. Two problems arise from this requirement for a triangle mesh. First, we can make a simple meshing algorithm to determine the proper relationship between the vertices. If each vertex is assigned a unique number, then we can store the results of meshing as a vector of integers and re-use the solution each time the tip is drawn. However, each vertex number needs to be a full-sized integer. A short integer can only hold at most the value 65535, not enough to count all 9 million vertices. Therefore, assuming that each vertex participates in only three triangles, the required index array would take up roughly 100 MB. Again, this is too much to store on the GPU. Since meshing is a relatively quick operation while transferring data is relatively slow, the mark generator simply re-meshes the data each time.

Another bigger problem arises from the way in which OpenGL samples the geometry for the depth camera. This sampling process is known as rasterization. In the interpolated mode, OpenGL draws only those fragments which lie inside of the established triangles in the mesh (Segal and Akeley, 2006). Since the meshed geometry is collapsed into an infinitely thin line using the squish matrix, none of the points reside inside of a triangle. Therefore, the depth camera cannot “see” these points and record them.

Therefore, the projected edge needs to be extruded to a finite thickness in the camera’s field of view. To do this, we adopt a meshing algorithm that meshes between two instances of the tip geometry as shown in Figure 3.14. For each rectangle of four points in the tip data (labeled 0, 1, 2, and 3 in the diagram), the algorithm forms triangles on six surfaces between a left and a right copy of the points: bottom, top, left, right, top-left to bottom-right diagonal, and top-right to bottom-left diagonal. Triangles are not formed out of any masked-off points. To

save processing and drawing time, the right and bottom surfaces are only formed at the right and bottom borders of the tip data, respectively, to avoid drawing redundant surfaces. As a valid triangle is formed, its vertices are sent to the GPU with w , the fourth vertex coordinate which is typically equal to 1, set to one of two pre-determined values indicating whether the vertex belongs to the left or right data instance.

Figure 3.14 Meshing between a left and right copy of the vertex data. (a) Surfaces formed. (b) Meshing pattern that avoids redundant surfaces.

At the GPU, we process each vertex with a GLSL vertex shader. This shader retrieves the left/right indicator in the w coordinate and temporarily stores it. It sets the w coordinate back to 1 and applies the modelview matrix stack from OpenGL (the coordinate system, the desired rotations, and the squish operation). Then, it pulls the two squished instances of data apart by setting the vertex's x coordinate to the left/right indicator value. For the current software version, these indicators are -2 and 2. Finally, the shader applies the projection matrix to the data.

The net effect of this shader program is something analogous to a game of cat's cradle. The

shader rotates and then squishes two identical copies of the tip that exist in the exact same space. Then it moves each copy to an opposite side of the screen. When this happens, the triangles formed between the two data sets “stretch” to form an extrusion of the projected tip. The depth camera can then image the edge of the tip projection, taking samples in a vertical line from the center of these triangles.

3.2.3 Converting from projected coordinates back to μm

Figure 3.15 presents a mathematical model of the graphics pipeline. At the far right, a generic vertex enters the pipeline; at the far left, it has been transformed into x and y window coordinates w_x and w_y and a value of depth for the depth buffer z_{db} . Throughout the bracketed section, the data remains in μm . The projection matrix normalizes the data such that x , y , and z all range between -1 and 1. The `glViewport` command, provided that the window positioning arguments are both 0 as they are in this research, behaves as a pair of matrices that normalizes x to range between 0 and the window width, y to range between 0 and the window height, and z to range between 0 and 1.

Figure 3.15 Mathematical model of the graphics pipeline for this research.

Therefore, to convert the depth buffer values back into μm values for the virtual mark, we must undo the action of the `glViewport` command and the projection matrix. If the z

coordinate after the application of the projection matrix is designated z_{ndc} , then Equation 3.9 converts the depth buffer value back into projected (or normalized device) coordinates.

$$z_{ndc} = 2z_{db} - 1 \quad (3.9)$$

The virtual mark software uses an orthographic projection matrix, which has the form

$$M_P = \begin{bmatrix} \frac{2}{r-l} & 0 & 0 & -\left(\frac{r+l}{r-l}\right) \\ 0 & \frac{2}{t-b} & 0 & -\left(\frac{t+b}{t-b}\right) \\ 0 & 0 & \frac{-2}{f-n} & -\left(\frac{f+n}{f-n}\right) \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

where l , r , b , t , n , and f are the locations of the left, right, bottom, top, near, and far clip planes, respectively. Therefore, the virtual mark data can be unprojected using Equation 3.10 to yield a mark in μm .

$$z_{\text{mark}} = \frac{-1}{2} \left[(f - n)z_{ndc} + f + n \right] \quad (3.10)$$

Combining Equations 3.9 and 3.10 yields a simple conversion (Equation 3.11) from depth buffer values to mark values in μm .

$$z_{\text{mark}} = z_{db}(n - f) - n \quad (3.11)$$

3.2.4 Detecting the edges

The edge detection algorithm was developed through trial and error by analyzing mark data with MATLAB. The finalized C++ version follows these steps:

1. The derivative is approximated using partial differences. The resulting vector has elements $z_i - z_{i-1}$ and is one member shorter than the original mark data because it starts with $z_1 - z_0$.
2. The derivative vector is filtered with a size 15 Gaussian 1D filter to remove the effects of noise.

3. The mean and standard deviation of the filtered derivative are computed.
4. The algorithm begins looking in from both ends of the mark for a new start and end point with these criteria: the new points must be at least 20 points in from either edge of the mark and must be the first points beyond this border to satisfy $|z'(i) - \mu_{z'}| \leq \sigma_{z'}$. In the above condition, $z'(i)$ is the derivative at the i th point and $\mu_{z'}$ and $\sigma_{z'}$ are the mean and standard deviation of the derivative, respectively. If the search fails, the original start and end points are used.
5. If the search is successful, the mean and standard deviation are recomputed for the portion of the derivative within the new start and end points. Step 4 is then repeated once using this updated mean and standard deviation along with the original, full-sized derivative vector.
6. The final values of the start and end points are returned as a suggestion to the user for later trimming.

Figure 3.16 shows a representative result from this trimming algorithm. The red line is the complete virtual mark retrieved from Tip 8A at a y rotation of 60° . The left and right sides of the purple box indicate the trimming suggestion from the edge detection algorithm, and the trim controls boxes show the corresponding values of the new start and end indices. This trimming algorithm seems successful at removing the steep slopes from the sides of the virtual mark.

3.3 Graphical user interface

Figure 3.17 presents the main window of the graphical user interface (GUI) for the virtual mark software. This GUI was designed using Qt. The window is divided into three widgets: tip (top), plate (middle), and statistical comparison (bottom). The tip and plate widgets feature 3D representations of the file geometry on the left side. For these views, the geometry is down-sampled by a factor of 6 to improve graphics speed and performance. Users can left click and drag on the geometry to translate it, and a Qt-provided trackball model allows users

Figure 3.16 Virtual mark trimming. The sides of the purple box indicate the suggested trim points.

to intuitively rotate the geometry with right click and drag. The scroll wheel allows users to zoom in and out. Users can double-click on the plate view to interactively select a column of plate data for comparison. This selected column appears in the plate view as a red plane as shown in Figure 3.17. Users can view the geometry in one of four modes by clicking the buttons immediately to the left of the geometry views: shaded, wireframe, textured, and height-mapped. Textured mode overlays the 2D texture from the Alicona onto the 3D geometry. A fifth viewing mode is provided for the tip widget which shows the tip geometry projected in the direction of tool travel; this mode helps users understand the mark generation process.

The right sides of the tip and plate widgets provide plots for profile data. The plate widget provides the name of plate file and a box for changing the selected column. The tip widget provides the name of the tip file and boxes for editing the desired tip rotation for mark generation. Users can click the adjacent button to create a virtual mark; when the mark is complete, users can click on the virtual mark tab to see the view in Figure 3.16. This trim view

Figure 3.17 The graphical user interface.

presents the recommended end points from the edge detection algorithm as the left and right sides of a purple box and allows users to interactively change these end points. Moreover, the trim view features a flip button that flips the virtual mark to compensate for a plate scanned backwards.

For the tip widget, the statistics plot tab provides a view of the trimmed and de-trended mark as shown in Figure 3.17. De-trending is the process of fitting a first-order line to the data with linear least squares and then subtracting this line from the data; it is an essential preparatory step for the statistical comparison. For the plate widget, the statistics plot tab displays the selected column (the red plane in the 3D view). Once the user clicks the calculate button in the statistics widget, purple windows pop up on both statistics plots denoting the locations of the search windows (maximum correlation window pair).

Figure 3.18 presents the third type of plot view, the height map. This is an alternative view of the profile data that represents height as a color between black and white. The color bar on the right shows the scale of the data. This plot is designed to resemble the physical mark views on the 2D comparison microscopes used by practicing forensics examiners.

Figure 3.18 Height map plot.

The statistical comparison widget provides an interface to the algorithm validated in Chum-

bley et al. (2010) and described in Subsection 2.3.1. Users can use the boxes to edit the size of the search window and the validation windows (the rigid- and random-shift windows). The calculate button is enabled when there is both a virtual mark and a selected plate column; this button allows users to perform the comparison. The R and T values from the comparison appear on the right side of the window after the comparison. The R value represents the maximum correlation statistic, which corresponds to the goodness of match between the contents of the purple windows. The T value given here is actually the average of 200 T values, each T value resulting from the comparison of 50 rigid-shift window pairs to 50 random-shift window pairs.

Finally, the file menu allows users to open tip and mark files. Users can import the data from an Alicona .al3d file and save it after cleaning as a Qt-based .mt file for later use. The tools menu allows users to bring up the “Automator,” a dialog that sets up the GUI to automatically perform comparisons between several plate files and a tip at multiple angles. This automated process saves the T values to a comma-separated list file for later viewing and takes a screenshot of each comparison.

3.4 Summary

This chapter discussed the implementation of the new methodology for tool mark analysis. Tool tips and plates are scanned with the Alicona Infinite Focus Microscope, and the digitized geometry is algorithmically cleaned to reduce the effect of scanning artifacts. Virtual marks are then generated at desired angles by projecting the tip geometry in the direction of tool travel. The noise cleaning and virtual marking processes are fast. On a standard desktop computer with a 3.20 GHz processor and an NVIDIA Quadro FX 580 graphics card (a midrange card targeted for CAD applications), noise cleaning takes a few minutes, and virtual marking can be done in under ten seconds. The statistical algorithm of Chumbley et al. (2010) takes around one second to compute the averaged result of 200 T values. Cleaning, virtual marking, and comparisons can all be performed within one easy-to-use graphical user interface. Therefore, the software portion shows promise for streamlining tool mark comparisons.

CHAPTER 4. RESULTS AND DISCUSSION

This chapter presents the results of using the implemented virtual tool mark methodology. In particular, Section 4.1 discusses the validation of the tool mark simulation software against standard geometric models. This test gives an indication of how well the OpenGL algorithm performs at producing a simulated mark from the tool geometry. In other words, it indicates the quality of the re-sampling performed to capture the mark geometry from the projected tip. Section 4.2 presents the preliminary results of using the statistical algorithm to compare virtual marks to real marks. In the results of this study, distinct separation in the T values is observed when virtual marks are made at similar angles to physical marks made by the same tool, which is quite encouraging for forensic science.

4.1 Verification with known geometric standards

For this study, six simulated tool files were created containing data points sampled from ideal geometry functions. Three geometry functions were used: a chevron extrusion, shown in Figure 4.1; a half circular pipe, shown in Figure 4.2; and a trapezoid extrusion, shown in Figure 4.3. For each shape, two files were generated. The low-resolution file was constructed by taking 200 evenly spaced samples of the geometric profile (triangle, semicircle, trapezoid) and duplicating the profile thus obtained to create the extruded shape. The high-resolution file was generated in the same manner with 2000 samples of the geometric profile. Each file was opened in the Tool Mark Simulator software (with the cleaning algorithms bypassed, of course) and used to make a virtual mark. The virtual mark was then loaded into MATLAB and compared to the ideal geometry function.

In MATLAB, the inverse of the ideal function was used to estimate the offset in the sampling

Figure 4.1 The chevron extrusion geometric standard. (a)-(c) Low resolution (200 samples). (d)-(f) High resolution (2000 samples). (a), (d) The geometry standard. (b), (e) The virtual mark compared with the ideal geometry function. (c), (f) The error between the virtual mark and the ideal function in μm .

that resulted from the virtual marking process. When the OpenGL algorithm is used to re-sample the tip projection with the “depth camera,” there is no guarantee that the camera is perfectly aligned with the original data vertices. Therefore, the camera will most likely image the interpolated surfaces between the vertices. In other words, even though the camera samples at the same resolution as the underlying geometry, there is some unknown offset in where the sampling starts. In order to compute meaningful error between the ideal function and the OpenGL results, the ideal function has to be sampled at the same locations. To determine these locations, the inverse function was applied to the virtual mark to estimate the x axis locations where the samples were taken. The median of the error between these computed x axis values and the x values in the original file was used as the offset estimate.

Figure 4.2 The half cylinder geometric standard. (a)-(c) Low resolution (200 samples). (d)-(f) High resolution (2000 samples). (a), (d) The geometry standard. (b), (e) The virtual mark compared with the ideal geometry function. (c), (f) The error between the virtual mark and the ideal function in μm .

The offset thus obtained, the ideal function was sampled at the appropriate locations, and the virtual mark and this ideal profile were aligned vertically. This vertical alignment is necessary since the unprojection step used to convert the virtual mark back to μm did not undo the action of any of the scene rotation and translation matrices (see Section 3.2.3). Finally, the root-mean-square (RMS) error was computed between the two profiles.

Table 4.1 presents the resulting RMS errors. All of the errors are below 0.1%, which is a relatively small amount of error. Moreover, the error is reduced by an entire order of magnitude when the number of samples used to represent the ideal geometry in the tool file is increased by a factor of ten. These results are similar to those found by Zhang et al. (2012) for 2D re-sampling with OpenGL. Zhang et al. (2012) found RMS values of a similar magnitude and observed the same trend of decreasing error with increasing resolution.

Given that each tip file contains about 9000×1000 samples and that the RMS errors are

Figure 4.3 The trapezoid extrusion geometric standard. (a)-(c) Low resolution (200 samples). (d)-(f) High resolution (2000 samples). (a), (d) The geometry standard. (b), (e) The virtual mark compared with the ideal geometry function. (c), (f) The error between the virtual mark and the ideal function in μm .

low even for 200 samples, the level of error is acceptable for the application. Therefore, the virtual marking algorithm is judged to be effective at reproducing the geometry.

4.2 Statistical comparison results

A preliminary study involving both sides of 6 screwdriver tips and 34 plates marked by those tips was carried out with the Tool Mark Simulator software to determine the suitability of the virtual marking methodology as a method for streamlining tool mark analysis and to assess whether or not it could confirm basic assumptions of impression analysis theory. The sequentially manufactured screwdriver tips and some of the marked plates used were drawn from the same set used in both Faden et al. (2007) and Chumbley et al. (2010). Retired Illinois

Table 4.1 RMS error (%) for standard geometry virtual marks.

Resolution	Chevron Extrusion	Half Cylinder	Trapezoid Extrusion
200 samples	0.0358%	0.0613%	0.0127%
2000 samples	0.0011%	0.0055%	0.0004%

State Police tool mark examiner Jim Kreiser utilized the mark-making jig shown in Figure 3.7(a) to generate the physical marks on lead plates. For this study, sides A and B of tips 2-5, 8, and 44 were digitized and compared to plates marked by both sides of those tips at angles of 45° , 60° , and 85° . These angles were defined according to the coordinate system shown in Figure 3.7(b); thus, at 85° , the screwdriver handle was almost perpendicular to the plate. For Tip 5 Side B and Tip 44 Side A, physical marks were not available at 45° , so only marks at 60° and 85° from those tip sides were used. Finally, different sides of the same tip were treated as unique tool working surfaces that produced unique marks; prior studies such as Chumbley et al. (2010) confirm that this is a valid assumption.

Virtual marks were made with each tip side at y axis rotations in increments of 5° beginning at 30° and ending at 90° . Each virtual mark was then statistically compared with profiles taken from the middle of each digitized physical mark. Statistical comparisons were made with the algorithm of Chumbley et al. (2010), described in detail in Section 2.3.1, using a maximum correlation window size of 300 pixels and a validation window size of 50 pixels. Fifty rigid-shift and 50 random-shift windows were employed to compute each T value, and the average of 200 such T values was recorded for each comparison. During this study, it was discovered that certain plates were scanned in the reverse direction in the interests of improved scan quality. Therefore, each comparison was performed twice, once with a forward virtual mark and once with the virtual mark flipped.

Figure 4.4 presents a comparison between a virtual mark and a physical mark both made at 45° using Tip 8 Side B. Comparing the plots of both marks, the visual resemblance is uncanny. While there is some spike noise on the left-hand side of the virtual mark, it does not prevent the statistical algorithm from finding a T value of 4.41, well above one. This and other known-match comparisons look very promising. The following subsections will present and discuss the

results for known matches and the results for known non-matches.

Figure 4.4 A comparison between a virtual and a real mark made by Tip 8 Side B at 45°.

4.2.1 Known match results

Figures 4.5, 4.6, and 4.7 present the resulting averaged T values for known match comparisons at angles of 45°, 60°, and 85°, respectively. Here, “known match” indicates that the same tip side was used to make both the virtual marks and the real physical mark that were compared. These figures include only the forward comparisons in cases where plates were determined to be scanned in the forward direction and only the flipped comparisons in cases where plates were determined to be scanned in the opposite direction. The correct direction of the plate scan was determined both by observing whether the forward or reverse comparisons

had higher T values and by visual comparison of the plots of the virtual mark and the plate profile. Features present in the plots made it simple in most cases to determine the correct direction visually. In every case, the comparison direction with the higher T values agreed with the visual inspection.

In these figures, the T values are displayed in box-and-whisker plots (boxplots) made with the statistical program R. Each boxplot represents the spread of the T values. The T values were sorted in ascending order, and the box indicates the range of the middle 50% of the data, which is also known as the Interquartile Range (IQR). The thick black line denotes the position of the median value of the data. The bottom whisker indicates the smallest data point within 1.5IQR of the bottom edge of the box, and the top whisker indicates the largest data point within 1.5IQR of the top edge of the box (Vardeman and Jobe, 2001). The remaining data points, known as outliers, are depicted with dots.

Figure 4.5 Averaged T values from statistical comparisons of 45° physical marks to virtual marks made with same side of the same tip.

Figures 4.5, 4.6, and 4.7 reveal clear separation in the T values near the angle of the physical mark. For the 45° angle data in Figure 4.5, at virtual mark angles above 65° , the T values

Figure 4.6 Averaged T values from statistical comparisons of 60° physical marks to virtual marks made with same side of the same tip.

Figure 4.7 Averaged T values from statistical comparisons of 85° physical marks to virtual marks made with same side of the same tip.

are quite low. Between 30° and 60° , the T value distributions rise to a peak near 40° and 45° . At this peak, the spread of the T values is actually very small; they are consistently between 3 and 4. There are two outliers near zero which belong to Plate 5A-45, the striation left by using Tip 5 Side A at 45° . The T values were consistently below one for 5A-45. One possible explanation for this is the presence of oxidation on the surface of the plate. Figure 4.8(b) displays the texture of this plate. In comparison with Plate 8A-45 in Figure 4.8(a), we can see that Plate 5A-45 appears darker and has a large red spot near the end. At this point, the source of the oxidation is unclear. Using the Tool Mark Simulator GUI interactively to select a different column from the plate for comparison did yield a T value of 3.4 at a virtual mark angle of 45° as shown in Figure 4.9. Nevertheless, the T value retrieved from the comparison seemed to vary greatly with the location of the column selected from the plate, much more so than it varied for other plates. Moreover, as seen in Figure 4.9, when compared to the virtual mark made at 45° with Tip 5 Side A, the mark appears to be only about 60% of the full width of the screwdriver blade. Between the oxidization and the partial marking, it does not seem surprising that the algorithm had some trouble. Excluding Plate 5A-45, it appears that the software can identify that the marks were made at an angle of 45° .

For the 60° angle data in Figure 4.6, the T values cluster around 0.5 between 70° and 90° and around 1.0 for 30° and 35° . Between 40° and 65° , the T values rise to a somewhat broad peak. Like the peak in the T values for the 45° data, the spread in T values is actually quite small at this peak, especially at the virtual mark angles of 50° and 55° . There are outliers in the peak area, but these are not caused by a false negative match such as that seen for Plate 5A-45. Instead, they are caused by variability in the location of the peak for some individual pairs of tips and plates. A few pairs peak near 50° , and some peak near 60° or 65° . Despite the broad peak in the T values, taking into account both Figure 4.5 and Figure 4.6, it appears that the software can generally distinguish a 45° angle from a 60° angle.

For the 85° angle data in Figure 4.7, the presence of a peak is again clear. Between 30° and 55° and at 90° , the T values generally cluster below 1.0. Between 60° and 85° , the values rise to a peak, although for the most part, the spread of T values is still quite large except at 75° . Even at 75° , there are some outliers present. Part of the reason for this large variability

(a)

(b)

(c)

Figure 4.8 Plate textures for comparison. (a) Plate 8A-45. (b) Plate 5A-45. (c) Plate 2A-85.

is another false negative involving Plate 2A-85. All of the T values for this plate are below 1.1. By comparing the textures of Plates 8A-45 and 2A-85 in Figures 4.8(a) and 4.8(c), we can see that Plate 2A-85 is also very dark due to some oxidation. Moreover, as seen in Figure 4.10, the mark on Plate 2A-85 is also about 60% of the full screwdriver blade. In general, the same variability in average T value with column selection was observed as was seen for Plate 5A-45. Therefore, it appears that the two false negatives are caused by approximately the same problem which may be oxidation and/or the lack of a full mark.

In addition, some of the variability seen in the peak of the 85° angle data is due to the effect of different comparisons peaking at different locations. Four out of the 12 comparisons peak near 75° , six at 80° , and one at 85° . Nevertheless, again looking at Figures 4.5, 4.6, and 4.7 together, the ability of the software to distinguish the angle of a known match within a range of $\pm 5\text{-}10^\circ$ seems clear. This reinforces the tool mark analysis assumption that marks made by

Figure 4.9 A successful comparison with Plate 5A-45.

the same tip will only match if they are made at similar angles.

Table 4.2 summarizes the error between the angle of the known match predicted by the Tool Mark Simulator and the known angle. In computing this error, the predicted angle was interpreted as the virtual mark angle with the highest T value for a given comparison between a tip side and its physical mark. Moreover, the false negatives (Plates 5A-45 and 2A-85) were excluded from these statistics. From this table, we see that the magnitude of the average angle error increases as the angle the screwdriver handle makes with the horizontal increases. Despite this trend, the median reveals that the most common angle error is always -5° .

What is the source of this angle error? Given that Figure 3.11 is representative of the results for the application of the marking coordinate system to the screwdriver data, it does not seem likely that the angle error arises from the coordinate system or the digitization process. Moreover, the OpenGL simulation should only exhibit floating point error in the angles of

Figure 4.10 A successful comparison with Plate 2A-85.

rotation. Therefore, the error must arise from either the physical mark-making process or the statistical comparison.

Table 4.2 Error ($^{\circ}$) between the predicted angle of a known match and the true angle.

Statistic	45° plates	60° plates	85° plates	Total
Mean	-1.667°	-4.583°	-6.364°	-4.375°
Median	-5°	-5°	-5°	-5°

Figure 3.7(a) shows the mark-making jig. Experimentation with this jig revealed that there was slop of about 1° in the four-screw pattern around the y axis shown in Figure 3.7(b). This slop disappeared once the screws were tightened, but it is unclear whether or not re-tightening the screws often was part of the procedure during the physical marking process. The slop could have appeared during the marking process, especially since the marking process requires the examiner to exert a strong force on the screwdriver handle. A more troublesome source of

error came from the screwdriver itself. As shown in Figure 3.7(a), the screwdriver was not one piece but instead a handle with a magnetic bit holder that the various tips were fitted into for making the marks. It was found that the bit exhibited a slop of between $2\text{-}5^\circ$ in the magnetic holder. This, combined with the considerable force required to make the mark, probably led the marks to be made several degrees shallow, which agrees with the observed negative angle error.

The trend of increasing magnitude in the average angle error with increasing angle can also be explained by the force required to make the marks. As the angle increases, the amount of force required also increases, perhaps because more of the screwdriver bottom contacts the plate. This would tend to increase the effect of the slop on the mark produced. With less force, the tip might not rotate to the full extent in its seat.

If this is truly the explanation for the error, then why is the average error for the 85° plates less than -5° ? One explanation is that since the virtual marks were made at increments of 5° , if the actual error in the screwdriver orientation happened to be between -5° and -10° , the maximum T value might show up in either the -5° or the -10° virtual mark. Moreover, there is some variability in the average T value computed each time the Calculate button in the Tool Mark Simulator is clicked to perform the comparison. This is because the statistical algorithm randomly chooses its rigid and random shifts. The averaging of 200 T values acts to reduce this variability, but these average T values can still vary by about ± 0.15 for a particular pair of marks. This is enough in many cases to shift the maximum T value by $\pm 5^\circ$, but it is not enough to cause false matches or false non-matches. Therefore, the error in the seating of the screwdriver tip during marking combined with the variation in the averaged T value from the statistical software accounts for the observed angle error.

4.2.2 Known non-match results

Figures 4.11, 4.12, and 4.13 present the resulting boxplots for known non-match comparisons at the angles of 45° , 60° , and 85° , respectively. For these plots, the “known non-match” category included known matches where the virtual mark was flipped opposite to the correct direction of the plate scan such that the compared marks were flipped relative to each other.

The correct direction of the plate scan was determined according to the same procedure given in Subsection 4.2.1. Moreover, both forward and reverse comparisons between virtual and physical marks known to be made by different screwdriver tips and different sides of the same screwdriver tips were also included in the known non-match plots.

Figure 4.11 Averaged T values from statistical comparisons of 45° physical marks to known non-matching virtual marks.

As expected, no clear trend can be seen in Figures 4.11, 4.12, and 4.13. All of the T values cluster around zero, and excluding outliers, they all fall between ± 1.5 . Nevertheless, there are a large number of outliers, and some of them take negative values of large magnitudes. One reason for the large number of outliers is that there are ten times as many data points for each non-match boxplot as there are for each match boxplot, since each of the 12 tip sides is compared to 9-11 non-matching physical marks and only one matching mark. However, this cannot be the only the reason, since the spread is so large.

According to Professor Max Morris of the Iowa State University Statistics Department, the development lead for the statistical algorithm in Chumbley et al. (2010), the outliers lying within the range of ± 2.5 are reasonable for the operation of this algorithm. However, there

Figure 4.12 Averaged T values from statistical comparisons of 60° physical marks to known non-matching virtual marks.

Figure 4.13 Averaged T values from statistical comparisons of 85° physical marks to known non-matching virtual marks.

are still many negative outliers outside of this range. The source of these outliers can be understood by viewing the locations of the maximum correlation windows found for these comparisons. Figure 4.14 presents a typical comparison with a very low negative T value. The purple boxes show the locations of the maximum correlation windows, which are at opposite ends of the compared marks. For the virtual mark plot, there are very few points to the right of the purple box, and for the physical mark plot, there are very few points to the left. This creates a problem when the rigid shift windows are created since they must be at a common distance from the maximum correlation windows. If the rigid shift is too far to the right of the purple box, it runs into the end of the virtual mark and gets rejected. If the rigid shift is too far to the left of the purple box, it runs into the end of the physical mark. Therefore, the possible population of the rigid shift windows is very limited.

Why does this restriction on the location of the rigid shift windows cause a T value with a large magnitude? To the right of the purple box, the virtual mark data has a large positive slope and the physical mark data has a negative slope. If the rigid shift window lands there (which it is more likely to do since there is not as much room to the left of the purple box), the data will have a strong negative correlation. This negative correlation will be larger in magnitude than the correlations due to random chance, since the random shift windows can still be located anywhere in the two marks. In this way, a strongly negative T value was obtained.

Visual inspection of the matches that produced large-magnitude negative T values revealed that this “opposite end problem” was responsible for all of them. In every one of these cases, the limited locations of the rigid-shift windows had opposite slopes, producing a strong negative correlation. Professor Morris indicates that a future update will be made to the statistical algorithm to prevent this from happening. There is a possibility that this bug could cause a false positive, but visual inspection of the known matches indicated that their high T values were caused by true matching. There was one case of a T value slightly above 2 occurring due to this opposite end problem, but the T value did not make it above the threshold of 2.5 that would have placed it in competition with the matches.

Therefore, the Tool Mark Simulator was successful at distinguishing between known matches and known non-matches in this preliminary study. All of the matches identified (defined for

Figure 4.14 A known non-match comparison with a negative T value of high magnitude.

the purposes of this study as having a maximum T value above 2.5) were true matches near the correct marking angle. Only two false negatives occurred, and those could be plausibly explained by partial marking and/or oxidation. Once the opposite end problem is fixed in the statistical package, the strongly negative T values seen for non-matches should also be eliminated. The results seem to strongly support the assumptions made by tool mark examiners.

4.3 Summary

The developed virtual marking application is successful at simulating the marking process. It is able to accurately reproduce standard geometry, and its marks closely resemble cross sections of physical marks. Moreover, the software appears to successfully determine which of the 12 tip sides made which physical mark, with zero false positive matches and only two false negative matches. Therefore, it both reinforces the conclusions of the tool mark analysis community and offers a promising new method for linking a physical mark directly to a tool tip.

CHAPTER 5. CONCLUSIONS AND FUTURE WORK

In this thesis, a new methodology for forensic tool mark analysis was implemented and tested. This methodology involves digitizing the tool tip and marked plate geometry, reducing the effect of any noise artifacts in those scans, projecting the tip geometry to simulate the marking process, and using the statistical algorithm developed in Chumbley et al. (2010) to compare the virtual mark to a physical mark. This chapter will make conclusions about this research in Section 5.1 and discuss the future direction of this methodology in Section 5.2.

5.1 Conclusions

Overall, the methodology seems very promising. In the preliminary study of both sides of 6 screwdriver tips and 34 corresponding marks, the software was able to distinguish matches from non-matches with zero false positive matches and only two matches mistaken for non-matches. Moreover, the software could predict the true angle of the mark within $\pm 5\text{-}10^\circ$. A few comparisons suffered from the opposite end problem discussed in Subsection 4.2.2, but this did not affect the results. This particular problem was not attributable to the methodology; rather, it was an inherent limitation of the statistical algorithm. Therefore, it seems that the model of the mark as the projection of the tool tip geometry in the direction of tool travel captured a significant amount of the physical marking process, although the false negatives suggest that adjustments for partial marking phenomena may need to be added.

The software components of the methodology were fast and able to be packaged in a simple graphical user interface (GUI). The software was used on a commercial desktop computer with a 3.20 GHz processor and an NVIDIA Quadro FX 580 graphics card (a midrange card targeted for CAD applications). On this system, cleaning the tip or plate geometry took a few minutes

and making virtual marks about ten seconds. Moreover, computing 200 T values and returning their average value took on the order of one second. Therefore, the software components have the potential to be deployed on existing computers in crime laboratories for usability testing in the near future. Furthermore, the GUI contains nicely formatted displays of the data and comparisons and a preview of the projected geometry. These features mean that the GUI may be good for demonstration purposes in the court of law.

The results of this methodology reinforce the current conclusions of tool mark analysis. The statistical comparisons in the preliminary study agree with the premise that marks are unique to the tool that produced them since it was possible to distinguish known matches from known non-matches. Furthermore, different tip sides did indeed behave as unique tool working surfaces, making their own unique marks. Finally, the ability of the software to distinguish the correct angle reinforces the conclusion that tool marks from the same tool must be made at the same angle in order to match.

Finally, the methodology demonstrates potential for streamlining and improving the practice of tool mark analysis. With this procedure, the examiner can use the touch-free optical profilometer to digitize the geometry of the tip and the mark, reducing the potential damage to the evidence. The software then provides an automated analysis that can run unsupervised while the examiner investigates another issue. The final results can be viewed and used to predict the angle of the match. Then, the examiner only needs to make marks around the suggested angle and compare those to the evidence mark. In this way, the methodology displays promise for saving the examiner's time, which is important in busy crime labs with large case loads. Most importantly, the methodology may provide a means for directly linking an evidence mark to the tool that made it. This more direct link will avoid some of the uncertainty in comparing two marks and is more likely to satisfy the courts.

5.2 Future work

There are many dimensions of the project that need to be addressed in future work. First and foremost, the statistical study needs to be enlarged to fully confirm the capability of this method. A study with the full set of 50 screwdrivers and their corresponding marks utilized

in Chumbley et al. (2010) would greatly further these results. Moreover, another preliminary test should be performed with one-piece sequentially manufactured screwdrivers to see if this eliminates the error currently attributed to the slop between the screwdriver bit and the handle. Additionally, examining some shallower angles (such as 30°) and/or twists would be good for completeness. It would also be insightful to try identifying the unknown marks that Jim Kreiser made during the Chumbley et al. (2010) study.

A usability study should also be performed with the examiners to incorporate their input on the design and features of the software. This work will be presented at the 2013 American Academy of Forensic Sciences Annual Scientific Meeting, where the community will have an opportunity to make suggestions for the software and the research approach. Later, copies of the software and sample data could be distributed under an open source agreement to allow volunteers to try it out on machines in their laboratories. This would be insightful for finding bugs and ensuring that the software performs well on examiners' existing computer hardware.

Development of the code base will also need to continue. It should be generalized and tested for use with other types of tool data, such as Phillips-head screwdrivers. Moreover, improved automation facilities should be added to speed up batch cleaning and comparisons. The current automation interface does not perform comparisons with multiple columns of the plate, which might be helpful to reduce false negatives and angle error. It might also be advantageous to make and store multiple virtual marks from a tool tip for later rapid comparison to a large number of plates.

Software development will also hopefully include some code that adjusts the performance to suit the user's system. It should be robust enough to work around some missing OpenGL functionalities and to take advantage of more advanced tools if the user's graphics card can support them. Moreover, it should work equally well on Windows, Mac, and Linux. Currently, it runs fully featured on Ubuntu Linux, and it runs well on Windows without the Automator feature.

Ultimately, it is hoped that the Tool Mark Simulator will become part of a larger Mark and Tool Inspection Suite (Mantis) which will also feature a mark-to-mark comparison interface. Mantis will hopefully provide the option of using a different statistical comparison algorithm

and possibility even a machine learning algorithm. Furthermore, Mantis may have a companion application for viewing comparison results on a smartphone and/or tablet. Finally, Mantis may be able to directly interface with the optical profilometer, allowing it to retrieve the geometry and clean it immediately.

Finally, some related issues will need to be addressed. The statistical algorithm should be improved to eliminate the opposite end problem. The lighting and scanning procedure for digitizing the tip should be further standardized. The problem of mark corrosion should also be addressed, perhaps with the selection of a different marking material.

5.3 Summary

The virtual marking methodology shows great promise for reinforcing and streamlining the discipline of forensic tool mark analysis. In a preliminary study, it was able to distinguish known matches from known non-matches without any false positive matches, and it could predict the angle of the match within $\pm 5\text{-}10^\circ$. Future work will expand the study to include more tools and will allow forensic examiners to provide input for the direction of future related research and software development. Finally, this software will hopefully become part of a larger suite for aiding tool mark examiners.

BIBLIOGRAPHY

- Alicona Imaging GmbH (2012). *Focus Variation*. English edition. Retrieved from is-suu.com/aliconadocs/aliconafocusvariation3_e_high_resolution on 29 Oct. 2012.
- Association of Firearm and Toolmark Examiners (1998). Theory of identification as it relates to toolmarks. *AFTE Journal*, 30(1):86–88.
- Bachrach, B. (2002). Development of a 3D-based automated firearms evidence comparison system. *Journal of Forensic Sciences*, 47(6):1–12.
- Biasotti, A. A. and Murdock, J. (1984). “Criteria for identification” or “state of the art” of firearm and toolmark identification. *AFTE Journal*, 16(4):16–24.
- Bolton-King, R. S., Bencsik, M., Evans, J. P. O., Smith, C. L., Allsop, D. F., Painter, J. D., and Cranton, W. M. (2012). Numerical classification of curvilinear structures for the identification of pistol barrels. *Forensic Science International*, 220:197–209.
- Bolton-King, R. S., Evans, J. P. O., Smith, C. L., Painter, J. D., Allsop, D. F., and Cranton, W. M. (2010). What are the prospects of 3D profiling systems applied to firearms and toolmark identification? *AFTE Journal*, 42(1):23–33.
- Bonfanti, M. and De Kinder, J. (1999). The influence of the use of firearms on their characteristic marks. *AFTE Journal*, 31(3):318–323.
- Chumbley, L. S., Morris, M. D., Kreiser, M. J., Fisher, C., Craft, J., Genalo, L. J., Davis, S., Faden, D., and Kidd, J. (2010). Validation of tool mark comparisons obtained using a quantitative, comparative, statistical algorithm. *Journal of Forensic Sciences*, 55(4):953–961.
- Daubert v. Merrell Dow Pharmaceuticals, Inc.* (1993). 509 U.S. 579, 113 S.Ct. 2786.

- Faden, D., Kidd, J., Craft, J., Chumbley, L. S., Morris, M., Genalo, L., Kreiser, J., and Davis, S. (2007). Statistical confirmation of empirical observations concerning tool mark striae. *AFTE Journal*, 39(3):205–214.
- Frye v. United States* (1923). 293 F. 1013 D.C. Cir.
- Geradts, Z., Zaal, D., Hardy, H., Lelieveld, J., Keereweer, I., and Bijhold, J. (2001). Pilot investigation of automatic comparison of striation marks with structured light. *Proceedings of SPIE*, 4232:49–56.
- Kidd, J. A. (2007). Comparison of screwdriver tips to the resultant toolmarks. Master’s thesis, Iowa State University.
- National Academy of Sciences (2009). *Strengthening forensic science in the United States: A path forward*. National Academies Press, Washington, D.C.
- Petraco, N., Petraco, N. D., and Pizzola, P. A. (2005). An ideal material for the preparation of known toolmark test impressions. *Journal of Forensic Sciences*, 50(6):1–4.
- Petraco, N., Shenkin, P., Speir, J., Diaczuk, P., Pizzola, P., Gambino, C., and Petraco, N. (2012a). Addressing the National Academy of Sciences’ challenge: A method for statistical pattern comparison of striated tool marks. *Journal of Forensic Sciences*, 57(4):900–911.
- Petraco, N. D. K., Chan, H., Forest, P. R. D., Diaczuk, P., Gambino, C., Hamby, J., Kammerman, F. L., Kammrath, B. W., Kubic, T. A., Kuo, L., McLaughlin, P., Petillo, G., Petraco, N., Phelps, E. W., Pizzola, P. A., Purcell, D. K., and Shenkin, P. (2012b). Application of machine learning to toolmarks: Statistically based methods for impression pattern comparisons. U.S. Department of Justice Grantee Report No. 239048. Retrieved from <https://www.ncjrs.gov/pdffiles1/nij/grants/239048.pdf> on 29 Oct. 2012.
- Roberge, D. and Beauchamp, A. (2006). The use of BulletTrax-3D in a study of consecutively manufactured barrels. *AFTE Journal*, 30(2):166–172.
- Segal, M. and Akeley, K. (2006). *The OpenGL Graphics System: A Specification*. Retrieved from <http://www.opengl.org/registry/doc/glspec21.20061201.pdf>.

Shreiner, D., Woo, M., Neider, J., and Davis, T. (2007). *OpenGL Programming Guide*. Addison-Wesley Professional, Boston, MA, sixth edition.

The Committee on the Judiciary House of Representatives (2009). *Federal rules of evidence*. Retrieved from <http://www.uscourts.gov/uscourts/RulesAndPolicies/rules/EV2009.pdf> on 29 Oct. 2012.

Vardeman, S. B. and Jobe, J. M. (2001). *Basic engineering data collection and analysis*. Duxbury Thomson Learning, Pacific Grove, CA.

Zhang, S., Ekstrand, L., Grieve, T., Eisenmann, D. J., and Chumbley, L. S. (2012). 3-D data processing with advanced computer graphics tools. *Proceedings of SPIE*, 8493:849310–849310–7.