

A Cartridge Case Comparison Pipeline

by

Joseph Zemmels

A thesis submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY

Major: Statistics

Program of Study Committee:
Heike Hofmann, Major Professor
Alicia Carriquiry
Kori Khan
Danica Ommen
Richard Stone
Susan VanderPlas

Iowa State University

Ames, Iowa

2023

DEDICATION

dedication text (optional)

TABLE OF CONTENTS

LIST OF TABLES	vii
LIST OF FIGURES	ix
ACKNOWLEDGEMENTS	xxii
ABSTRACT	xxiii
CHAPTER 1. Literature Review	1
1.1 Preliminaries: Forensic Examinations	1
1.1.1 Firearm and Toolmark Identification	2
1.1.2 Why Should Firearm and Toolmark Identification Change?	8
1.2 Forensic Comparison Pipelines	9
1.2.1 Digital Representations of Evidence	12
1.2.2 Pre-processing Procedures for Forensic Data	14
1.2.3 Forensic Data Feature Extraction	15
1.2.4 Similarity Scores & Classification Rules for Forensic Data	16
1.2.5 Reproducibility of Comparison Pipelines	18
1.3 Diagnostic Tools	19
1.3.1 Visual Diagnostics	20

1.3.2	Interactive Diagnostics	26
1.4	Automating and Improving the Cartridge Case Comparison Pipeline	27
1.4.1	Image Processing Techniques	27
1.4.2	Density-Based Spatial Clustering of Applications with Noise	36
1.4.3	Implementation Considerations	40
CHAPTER 2.	A Study in Reproducibility: The Congruent Matching Cells Algorithm and cmcR package	46
Abstract	46
2.1	Introduction	47
2.1.1	Repeatability and reproducibility	47
2.1.2	The Congruent Matching Cells algorithm	50
2.2	The CMC pipeline	53
2.2.1	Initial data	54
2.2.2	Pre-processing procedures	57
2.2.3	“Correlation cell” comparison procedure	63
2.2.4	Decision rule	68
2.3	Discussion	76
2.3.1	Ambiguity in algorithmic descriptions	76
2.3.2	CMC pattern matching pipeline	78
2.3.3	Processing condition sensitivity	79
2.4	Conclusion	83
2.5	Acknowledgement	85

2.6	Computational details	85
CHAPTER 3. Diagnostic Tools for Cartridge Case Comparison Algorithms		88
Abstract		88
3.1	Introduction	88
CHAPTER 4. Automatic Matching of Cartridge Case Impressions		89
Abstract		89
4.1	Introduction	90
4.1.1	Previous Work	93
4.2	Cartridge Case Data	94
4.3	Methods	95
4.3.1	Pre-processing	96
4.3.2	Comparing	97
4.3.3	Scoring	112
4.4	Results	114
4.5	Discussion	120
4.6	Conclusion	125
CHAPTER Computational Details		127
CHAPTER Acknowledgments		128
CHAPTER Appendix		129
A	Registration Procedure Details	129
A.1	Cell-Based Registration Details	131

A.2	Registration-Based Feature Distributions	132
B	DBSCAN Algorithm Details	132
B.1	Density-Based Feature Distributions	133
C	Visual Diagnostic Details	135
C.1	Visual Diagnostic Feature Distributions	136
D	Model-Specific Results	137
	REFERENCES	140

LIST OF TABLES

1.1	Moments of the two variables in Anscombe's quartet.	22
1.2	Two examples of data analysis workflows that utilize the pipe operator. The left side shows a data frame manipulation while the right side shows a comparison of two cartridge cases.	42
2.1	Description of pre-processing procedures from Song et al. 2014 vs. considerations that need to be made when implementing these procedures. Each of these considerations requires the implementer to decide between potentially many choices.	59
2.2	Example of output from correlation cell comparison procedure between Fadul 1-1 and Fadul 1-2 rotated by -24 degrees. Due to the large proportion of missing values that are replaced to compute the FFT-based correlation, the pairwise-complete correlation is most often greater than the FFT-based correlation.	69
2.3	Different thresholds for translation, rotation, and CCF_{max} are used across different papers. The range in CCF_{max} is particularly notable.	70
4.1	Six similarity features based on registering full scans and cells.	105
4.2	Four similarity features based on the density-based clustering procedure.	108

4.3	Nine similarity features calculated based on visual diagnostics.	113
4.4	Comparison of the ACES algorithm to previous work. Although ACES shares similarities to previously-proposed algorithms, it includes additional nuance by computing features across both comparison directions and using these features in a classifier model.	122
4.5	Testing classification error, false positive, and false negative rates for four types of classifier models. The CMC method results are derived from the implementation available in Zemmels et al. (2022a). The "Only C_0 feature" classifier is analogous to the classification rule used in Zhang et al. (2020). The last two rows show results from the Logistic Regression classifier trained on the all 19 ACES features under two optimization criteria - balancing the true negative and true positive rates or maximizing the overall classification accuracy on the training data set.	123
4.6	Accuracy, True Positive, and True Negative rates based on the training data for the 13 binary classifier models. This table shows a numeric summary of the results shown in [Figure]. We bold the largest values in each column for emphasis.	138
4.7	Accuracy, True Positive, and True Negative rates based on the test data for the 13 binary classifier models. This table shows a numeric summary of the results shown in [Figure]. We bold the largest values in each column for emphasis.	139

LIST OF FIGURES

1.1	A cartridge containing primer, powder, and a bullet. The firing process is initiated by loading a cartridge into the barrel of a firearm.	2
1.2	Cross-section of a pistol with a chambered cartridge and drawn-back hammer. Pulling the trigger releases the firing pin which strikes the cartridge case primer.	3
1.3	A cartridge after a firing pin has struck the primer. The explosion of the primer ignites the powder within the cartridge, causing gas to rapidly expand and force the bullet down the barrel.	3
1.4	Examples of common breech face impression patterns. These are considered analogous to a breech face fingerprint left on the cartridge surface.	4
1.5	A fired 9mm Luger cartridge case with visible firing pin and breech face impressions.	4
1.6	A comparison microscope consists of two stages upon which evidence is placed. These stages are placed under two compound microscopes that are joined together via an optical bridge and allow for viewing of both stages simultaneously under a single eyepiece. The image on the right shows an example of a bullet viewed under a comparison microscope.	5

1.7	Variations upon the cartridge case comparison pipeline. The first two columns detail the pipeline with different sub-procedures. The third column shows the parameters that require manual specification at each step. The fourth column shows alternative processing steps that could replace steps in the existing pipeline.	11
1.8	A cartridge case captured using 2D confocal reflectance microscopy (left) and 3D disc scanning confocal microscopy (right).	12
1.9	The TopMatch-3D High-Capacity Scanner from Cadre Forensics TM . The scanner captures topographic scans of a gel pad into which a cartridge case surface is impressed.	13
1.10	The hierarchy of information stored in the x3p file format for both bullet and cartridge case evidence.	14
1.11	A visualization of Anscombe's quartet. Despite there being obvious differences between these four data sets, their summary statistics are nearly identical	21
1.12	An example of a statistical graphic that uses the Gestalt Laws of Perceptual Organization to communicate data findings.	24
1.13	An example of using the ggplot2 package to construct a residual plot from a simple linear regression. The features of the statistical graphic are combined layer-by-layer using the + operator as we see in the accompanying code chunk.	25
1.14	A screenshot of the TopMatch-3D TM Virtual Comparison Microscopy software. In this example, similar and different markings on the cartridge case scans are manually annotated by the user using shades of blue and yellow/red, respectively.	27

1.15	(Left) A reference image A and template image B both featuring a white box of dimension 10×10 . (Right) The cross-correlation function (CCF) between A and B . The index at which the CCF is maximized represents the translation at which A and B are most similar.	30
1.16	An image A of a box with Gaussian noise undergoing a lowpass, highpass, and bandpass filter operation.	32
1.17	A 7×7 image A featuring a 3×3 box undergoing dilation and erosion by a 3×3 structuring element B	35
1.18	An ε -neighborhood around a observation located at $(3, 2)$ for $\varepsilon = 3$. Points are colored blue if they are neighbors to this observation and red otherwise.	37
1.19	An example of three points that are density-reachable with respect to $\varepsilon = 3$ and $Minpts = 2$	38
1.20	An example of two points that are density-connected, but not density-reachable, with respect to $\varepsilon = 3$ and $Minpts = 2$	38
1.21	Cluster labeling for 10 data points using the DBSCAN algorithm with parameters $\varepsilon = 3$ and $Minpts = 2$. Seven points are assigned to a single cluster and the remaining three are classified as noise.	39
1.22	A pre-processing procedure applied to a 2D image of a cartridge case to identify the firing pin impression. The procedure results in a 2D image of a cartridge case without the firing pin impression region.	43

1.23	A pre-processing procedure for extracting 2D bullet signatures” from a 3D topographic bullet scan. The procedure results in an ordered sequence of values representing the local variations in the surface of the bullet.	43
1.24	A pre-processing procedure applied to an image of the handwritten word ”csafe.” The procedure results in a skeletonized version of the word that has been separated into graphemes as represented by orange nodes.	44
1.25	A cartridge case undergoing various pre-processing steps. The procedure results in a cartridge case scan in which the breech face impressions have been segmented and highlighted.	44
2.1	A cartridge case pair with visible breech face impressions under a microscope. A thin line can be seen separating the two views. The degree to which the markings coincide is used to conclude whether the pair comes from the same source.	51
2.2	The stages of CMC pipelines. In the pre-processing stage, each scan is prepared for analysis, removing extraneous information and noise. Then, each scan is broken up into cells, which are numerically compared to cells in the other scan to determine an optimal alignment. Finally, each of the scores arising from the cells in the second stage are compared to a reference distribution to determine whether the scans originate from the same source or from different sources. . . .	54

2.3	Unprocessed surface matrices of the known-match Fadul 1-1 and Fadul 1-2 Fadul et al. 2011. The observations in the corners of these surface matrices are artifacts of the staging area in which these scans were taken. The holes on the interior of the primer surfaces are caused by the firing pin striking the primer during the firing process. The region of the primer around this hole does not come into uniform contact with the breech face of the firearm.	56
2.4	Overview of the set of pre-processing steps used in the CMC algorithms. Where a procedure step is not discussed or explicitly not applied in the paper, the path traverses empty space.	58
2.5	Illustration of the sequential application of pre-processing steps implemented in cmcR. We map the cartridge case surface height values to a divergent purple-white-orange color scale to emphasize deviations from the median height value (represented here as 0 micrometers). At each stage, the variability in height across the scan decreases as we emphasize the regions containing breech face impressions.	60
2.6	Fadul 1-1 and Fadul 1-2 after pre-processing. Similar striated markings are now easier to visually identify on both surfaces. It is now clearer that one of the scans needs to be rotated to align better with the other.	62

2.7	Illustration of comparing a cell in the reference cartridge case scan (left) to a larger region in a questioned cartridge case scan (right). Every one of the cells in the reference cartridge case is similarly paired with a region in the questioned cartridge case. To determine the rotation at which the two cartridge cases align, the cell-region pairs are compared for various rotations of the questioned cartridge case.	63
2.8	Each CMC implementation uses a slightly different procedure to obtain a similarity score between two cartridge cases. Steps which are implemented with additional user-specified parameters are shaded purple; steps which are described but without sufficient detail are shaded grey.	64
2.9	CMC results for the comparison between Fadul 1-1 and Fadul 1-2 using the original decision rule. The two plots in the top row show the 18 CMCs when Fadul 1-1 is treated as the "reference" cartridge case to which Fadul 1-2 (the "target") is compared. The second row shows the 17 CMCs when the roles are reversed. Red cells indicate where cells not identified as congruent achieve the maximum pairwise-complete correlation across all rotations of the target scan.	74

2.10	Applying the High CMC decision rule to the comparison of Fadul 1-1 and Fadul 1-2 results in 20 CMCs when Fadul 1-1 is treated as the reference (top) and 18 CMCs when Fadul 1-2 is treated as the reference (bottom). Although the individual comparisons do not yield considerably more CMCs than under the original CMC pipeline, Tong et al. (2015) indicate that the High CMCs from both comparisons are combined as the final High CMC count (each cell is counted at most once). Combining the results means that the High CMC decision rule tends to produce higher CMC counts than the original CMC pipeline. In this example, the combined High CMC count is 24 CMCs.	75
2.11	Applying both decision rules to the comparison between the non-match pair Fadul 1-1 and Fadul 2-1 results in 2 CMCs under the original decision rule (shown above) and 0 CMCs under the High CMC decision rule (not shown). The seemingly random behavior of the red cells exemplifies the assumption that cells in a non-match comparison do not exhibit an observable pattern. Random chance should be the prevailing factor in classifying non-match cells as CMCs.	77
2.12	CMC count relative frequencies under the original decision rule and the High CMC decision rule for $T_{\Delta x} = 20 = T_{\Delta y}$ pixels, $T_{CCF} = 0.5$, and $T_\theta = 6$ degrees. An $AUC = 1$ corresponds to perfect separation of the match and non-match CMC count distributions. We can see that, for this set of processing parameters, the High CMC decision rule yields higher CMC counts for known matches than the original decision rule while known non-matches have the same distribution under both methods.	80

2.13	Variance ratio values are plotted for different parameter settings. High variance ratios are indicative of a good separation between CMC counts for known matching pairs and known-non matching pairs. The High CMC decision rule generally performs better than the original decision rule. Removing the trend during pre-processing has a major impact on the effectiveness of the CMC pipeline. In this setting, translation thresholds $T_x, T_y \in [15, 20]$, a rotation threshold $T_\theta = 6$, and a CCF threshold $T_{CCF} \in [0.4, 0.5]$ lead to a separation of results.	81
2.14	Variance ratios based on results reported in various CMC papers. The High CMC decision rule tends to outperform the original decision rule. However, it should be emphasized that each paper uses very different processing and parameter settings meaning the results are difficult to compare. The values labeled "cmcR" show the largest variance ratio values for the original and High CMC decision rules based on a limited grid search. These results indicate that the CMC pipeline implementation provided in cmcR yields comparable results to previous CMC papers.	82

4.1	Comparison of the traditional examination vs. the currently proposed method for comparing cartridge cases. Both start with two fired cartridge cases. In traditional examination, an examiner uses a microscope to assess the "agreement" of markings on the two cartridge case surfaces. They decide whether or not the cartridge cases were fired from the same firearm, or if there is inconclusive evidence to decide. In the ACES algorithm, we take a topographical scan of the cartridge case surfaces and manually identify the regions containing distinguishable markings (shown in red). We pass these scans to the ACES algorithm, which processes and compares the two scans. The final result is a numerical measure of similarity of the two cartridge cases.	91
4.2	We apply a sequence of pre-processing functions to each scan. Each pre-processing step further emphasizes the breech face impressions in the scan.	97
4.3	A matching pair of processed cartridge case scans. We measure the similarity between these cartridge cases using the distinguishable breech face impressions on their surfaces.	99
4.4	Estimated registrations of cells from a non-match pair of cartridge cases. A source scan (left) is separated into an 8×8 grid of cells. We exclude cells containing only missing values (visualized here as gray pixels). Each source cell is compared to a target scan (right) to estimate where it aligns best. We show a handful of cells at their estimated alignment in the target scan and magnify the surfaces captured by cell pairs 5, 1 and 7, 7. Although the cartridge case pair is non-matching, we note that there are similarities in the surface markings for these cell pairs.	103

4.5	Cluster assignments based on the Density Based Spatial Clustering with Applications to Noise (DBSCAN) algorithm for estimated translations in two comparison directions. Using scan A as source results in a cluster of size 14 (left) compared to 13 when scan B^* is used as source (right). Noting the reversed axes in the right plot, we see that the clusters are located approximately opposite of each other. Points are jittered for visibility.	107
4.6	(Left) After aligning two scans, we filter regions that are "different" from each other, meaning the absolute difference between surface values is larger than some threshold. (Middle) We binarize the scan into "filtered" or "non-filtered" regions - shown in white and black, respectively. (Right) Using a connected components labeling algorithm, we identify connected "neighborhoods" of filtered elements. We assume that these neighborhoods will be small, on average, if comparing truly matching cartridge cases.	111

4.7	We summarize classification accuracy, true negative, and true positive rates for both the training and testing results, represented as gray and black points/lines respectively, for 13 binary classifier models. Our primary interest is the test data results, but visualizing the training data results allows us to assess the generalizability of the models after training. In the first row, we consider a classifier based on a single feature, the Cluster Indicator feature C_0 , as a baseline. The remaining rows show results from training/testing classifiers based on a decision tree (CART), random forest (RF), and logistic regression (LR) under various feature sets and optimization critieria. The second and third rows show results based on a subset of seven features from the ACES feature set while the fourth and fifth rows show results using all 19 ACES features. We also compare results from optimizing either the overall Accuracy (third and fifth rows) or the area under the Receiver Operating Characterstic curve (AUC, second and fourth rows) during training.	115
4.8	A dot plot of the predicted match probabilities for the non-match and match comparisons in the test set based on a logistic regression model. As we expect, the non-match comparisons tend to have a low match probability. However, we see that there are many matching comparisons that also have a low match probability.	118

4.9	(Left) A dot plot of the predicted match probabilities for the match comparisons in the test set based on a logistic regression model, separated by firearm. We see that firearm T has more matching comparisons with low match probabilities than the other test firearms. (Right) Misclassifications divided by total number of pairwise comparisons for each pair of test firearms based on the same logistic regression model. For clarity, we do not show comparisons with 0 misclassifications. We note that the proportion of misclassified matching comparisons from firearm T of 45.9% is much higher than that of other comparisons.	119
4.10	Variable importance measures from fitting a random forest to the training data set, repeated 10 times under various random seeds. The top three features consist of one feature from each of the three feature groups defined in the Methods section. Points are plotted on a log scale and vertically jittered for visibility.	121
4.11	A scatterplot where points represent the cell-wise estimated translations faceted by rotation for a matching pair of cartridge cases. As evidenced by the tight cluster in the middle facet, it appears that multiple cells agree on a translation of $[m, n] \approx [17, -16]$ after rotating by 3° . Points are jittered for visibility.	132
4.12	Density plots of the Registration-Based features for 21,945 cartridge case pairs. The standard deviation of the cell-based registrations distinguish between match and non-match pairs better than the mean values.	133

4.13	Distributions of the density-based features for 21,945 cartridge case pairs. The Cluster Size and Estimated Translation Difference features may be missing (NA) if no DBSCAN cluster is identified, which commonly occurs for non-matching cartridge case pairs as evidenced by the stacked bar chart in the top left. This explains the near absence of non-matching comparisons from Cluster Size and Estimated Translation Difference plots. Whether a cluster is identified for a particular comparison strongly predicts whether it is a match or a non-match, which justifies the inclusion of the cluster indicator feature C_0	134
4.14	Full scan comparison plot.	135
4.15	Distributions of the visual diagnostic-based features for 21,945 cartridge case pairs. Matching comparisons tend to have smaller neighborhood sizes on average and higher correlation values than non-matches indicating their utility in a classifier.	136

ACKNOWLEDGEMENTS

Acknowledgements go here.

ABSTRACT

Algorithms to compare evidence are increasingly used in forensic examinations to supplement an examiner’s opinion with an objective measure of similarity. However, an algorithm must first be thoroughly tested under various conditions to identify its strengths and weaknesses. This experimentation is expedited for algorithms that are accessible to fellow researchers and practitioners. In this work, we discuss an algorithm to objectively measure the similarity between cartridge cases. We have designed this algorithm to be approachable for researchers and practitioners alike. Chapter 2 discusses a modularization of the algorithm into a “pipeline” that enables reproducibility, experimentation, and comprehension. Our goal in this modularization is to lay a foundation upon which improvements can be easily developed. Chapter 3 details a suite of diagnostic tools that illuminate the inner-workings of the algorithm and determine when and why the algorithm “works” correctly. These diagnostics will be useful for both researchers interested in correcting the algorithm’s behavior and for practitioners concerned with applying the algorithm to case work. Chapter 4 introduces novel pieces of the pipeline that we demonstrate are improvements to predominant methods. In particular, we introduce the Automatic Cartridge Evidence Scoring (ACES) algorithm that measures the similarity between two cartridge cases using a novel set of numeric features.

CHAPTER 1. Literature Review

1.1 Preliminaries: Forensic Examinations

A bullet casing is found at the scene of a murder. The bullet is recovered from the victim during autopsy. A handwritten letter threatening the victim is found in their pocket. The assailant’s shoeprints are discovered fleeing the area. Who left this evidence? Investigators obtain the gun, shoes, and handwriting samples of a suspect. This evidence, along with the crime scene evidence, is sent to a forensic laboratory for analysis. Forensic examiners compare the evidence to establish whether they share a common source. The suspect is charged after the examiners conclude that there is sufficient agreement between the crime scene and suspect’s samples.

The procedure described above, in which evidence is analyzed to determine its origin, is called the *source identification* problem (Ommen and Saunders, 2018). Historically, forensic examiners have relied on tools (e.g., microscopes), case facts, and experience to develop an opinion on the similarity of two pieces of evidence. More recently, algorithms to automatically compare evidence and provide an objective measure of similarity have been introduced. These algorithms can be used in a forensic examination to supplement and inform the examiner’s conclusion. We propose an automatic, objective solution to the source identification problem; specifically in the context of comparing fired *cartridge cases*. Cartridge case comparison is a sub-discipline of Firearm and Toolmark Identification, which is reviewed in the next section.

1.1.1 Firearm and Toolmark Identification

Firearm and toolmark identification involves studying markings or impressions left by a hard surface such as a firearm or screwdriver on a softer surface (Thompson, 2017). For example, a barrel's rifling leaves toolmarks on a bullet as it travels out of the gun.

1.1.1.1 The Firing Process

In this section, we describe the basic process of firing a cartridge out of a handgun or rifle. A *cartridge* consists of a metal casing containing primer, gunpowder, and a bullet. Figure 1.1 shows a cross-section of a cartridge featuring these components (30 Magazine Clip, 2017).

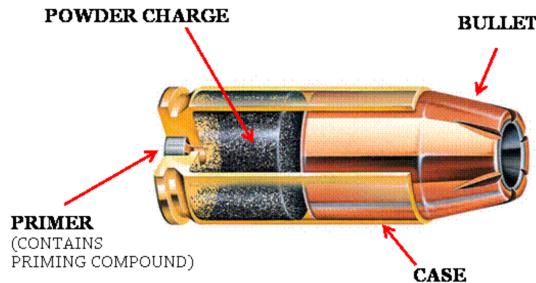


Figure 1.1: A cartridge containing primer, powder, and a bullet. The firing process is initiated by loading a cartridge into the barrel of a firearm.

First, a cartridge is loaded into the back of the barrel in an area called the *chamber*. Figure 1.2 shows an example of a cartridge loaded into the chamber of a pistol (Rattenbury, 2015). Note that the hammer of the pistol in Figure 1.2 is pulled to hold the firing pin under spring tension. Upon squeezing the trigger, the firing pin releases and travels forwards at a high velocity. The firing pin strikes the primer of the cartridge case, causing it to explode.

The explosion of the primer ignites the powder in the cartridge (Hampton, 2016). As shown in 1.3, gas rapidly expands in the cartridge that pushes the bullet down the barrel.

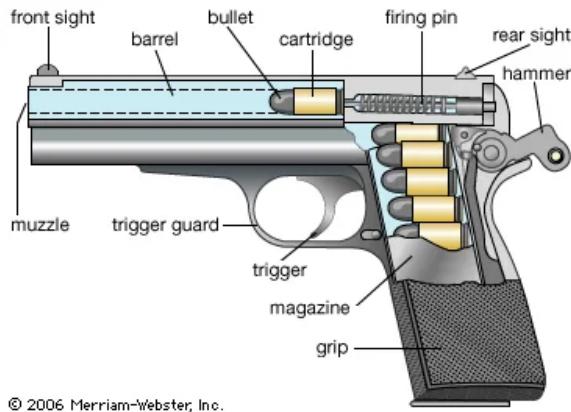


Figure 1.2: Cross-section of a pistol with a chambered cartridge and drawn-back hammer. Pulling the trigger releases the firing pin which strikes the cartridge case primer.

Simultaneously, the rest of the cartridge travels towards the back of the barrel and strikes the back wall of the barrel, known as the *breech face*, with considerable force.

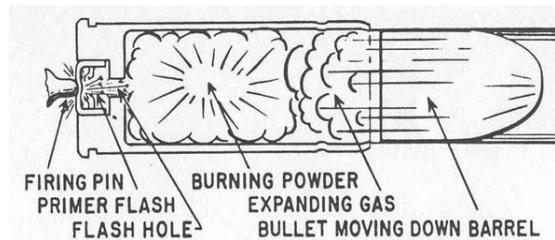


Figure 1.3: A cartridge after a firing pin has struck the primer. The explosion of the primer ignites the powder within the cartridge, causing gas to rapidly expand and force the bullet down the barrel.

Any markings on the breech face are imprinted onto the cartridge case, creating the so-called *breech face impressions*. These impressions are analogous to a barrel's "fingerprint" left on the cartridge case. Figure 1.4 shows cartoon examples of breech face markings that appear on cartridge cases (Hampton, 2016).

Figure 1.5 shows the base of a fired cartridge (Hampton, 2016). The hole to the south-east of the center of the primer is the impression left by the firing pin. Note the horizontal striated breech face markings on the primer to the left of the firing pin impression. We focus on the comparison of such markings.

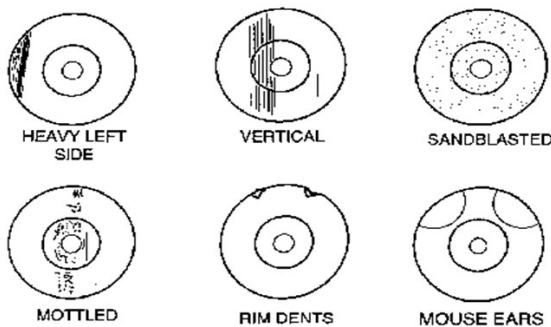


Figure 1.4: Examples of common breech face impression patterns. These are considered analogous to a breech face fingerprint left on the cartridge surface.



Figure 1.5: A fired 9mm Luger cartridge case with visible firing pin and breech face impressions.

1.1.1.2 An Overview of Firearm and Toolmark Examinations

Trained firearm and toolmark examiners use a *comparison microscope* like the one in Figure 1.6 to examine two pieces of evidence (Zheng et al., 2014). A comparison microscope combines the view of two compound microscopes into a single view via an *optical bridge*. This allows an examiner to view two microscope stages simultaneously under the same eyepiece. The right Figure 1.6 shows the view of two bullets under a comparison microscope. The white dotted line represents the split in the two fields of view. The goal of using a comparison microscope is to assess the “agreement” of the features on two pieces of evidence.

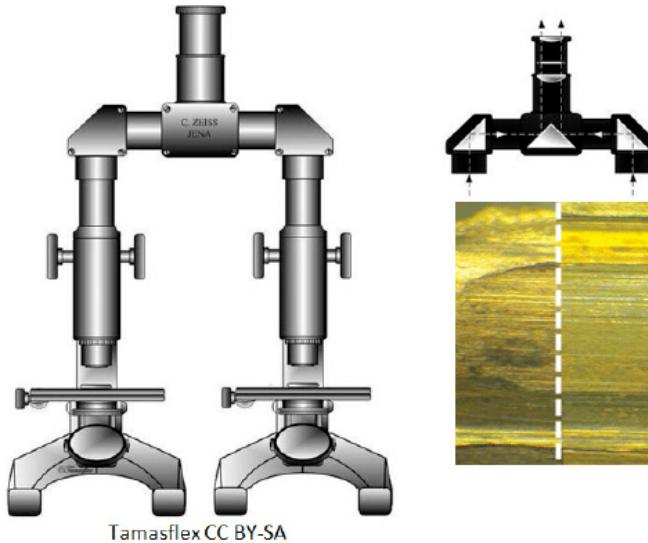


Figure 1.6: A comparison microscope consists of two stages upon which evidence is placed. These stages are placed under two compound microscopes that are joined together via an optical bridge and allow for viewing of both stages simultaneously under a single eyepiece. The image on the right shows an example of a bullet viewed under a comparison microscope.

Firearm examiners distinguish between three broad categories when characterizing the features of a fired bullet or cartridge case: class, subclass, and individual characteristics. *Class characteristics* are features associated with the manufacturing of the firearm such as the size of ammunition chambered by the firearm, the orientation of the extractor and ejector, or the width and twist direction of the barrel rifling. An early step in a forensic

examination is to determine the class characteristics of the firearm of origin as they can narrow the relevant population of potential firearm sources (Thompson, 2017). For example, a 9mm cartridge case must have been fired by a firearm that can chamber 9mm ammunition.

If the discernible class characteristics match between two pieces of evidence, then the examiner uses a comparison microscope to compare the *individual characteristics* of the evidence. Individual characteristics are markings attributed to imperfections on the firearm surface due to the manufacturing process, use, and wear of the tool. For example, markings on the breech face of a barrel often form after repeated fires of the firearm. Individual characteristics are assumed to occur randomly across different firearms and therefore can be used to distinguish between two firearms. The examiner independently rotates and translates the stages of a comparison microscope to find the position where the markings on the two pieces of evidence match (Zheng et al., 2014). An examiner concludes that the evidence originated from the same firearm if the individual characteristics are in “sufficient agreement” (AFTE Criteria for Identification Committee, 1992).

Subclass characteristics exist between the macro-level class and micro-level individual characteristics. These characteristics relate to markings reproduced across a subgroup of firearms. For example, breech faces manufactured by the same milling machine may share similar markings (Werner et al., 2021). It can be difficult to distinguish between individual and subclass characteristics during an examination. An examiner’s decision process may be affected if the existence of subclass characteristics is suspected.

Many firearm and toolmark examiners in the United States adhere to the Association of Firearm and Toolmark Examiners (AFTE) Range of Conclusions when making their evidentiary conclusions (AFTE Criteria for Identification Committee, 1992). According to these guidelines, six possible conclusions can be made in a firearm and toolmark examination:

1. **Identification:** Agreement of a combination of individual characteristics and all discernible class characteristics where the extent of agreement exceeds that which can

occur in the comparison of toolmarks made by different tools and is consistent with the agreement demonstrated by toolmarks known to have been produced by the same tool.

2. **Inconclusive:** there are three possible inconclusive decisions
 - 2.1 Some agreement of individual characteristics and all discernible class characteristics, but insufficient for an identification.
 - 2.2 Agreement of all discernible class characteristics without agreement or disagreement of individual characteristics due to an absence, insufficiency, or lack of reproducibility.
 - 2.3 Agreement of all discernible class characteristics and disagreement of individual characteristics, but insufficient for an elimination.
3. **Elimination:** Significant disagreement of discernible class characteristics and/or individual characteristics.
4. **Unsuitable:** Unsuitable for examination.

Forensic examinations first involve an examination of a “questioned” bullet or cartridge case for identifiable toolmarks (Thompson, 2017). The examiner classifies markings by their class, individual, and subclass characteristics. The examiner compares these characteristics to “known source” fires obtained from a suspect’s firearm if one is available. Otherwise, class characteristics from the questioned bullet can be used to narrow the relevant population and provide potential leads. An examiner’s decision may be used as part of an ongoing investigation or presented at trial as expert testimony.

Standard operating procedures for assessing and comparing evidence differ between forensic laboratories. For example, some labs collapse the three possible inconclusive deci-

sions into a single decision (Neuman et al., 2022) or prohibit examiners from making an elimination based on differences in individual characteristics (Duez et al., 2017).

1.1.2 Why Should Firearm and Toolmark Identification Change?

In 2009, the National Research Council released a report assessing a number of forensic disciplines including Firearm and Toolmark analysis. The report pointed out that firearm and toolmark analysis lacked a precisely defined process and that little research had been done to determine the reliability or repeatability of the methods. *Reliability* refers to the ability to correctly classify evidence as originating from the same source or not. *Repeatability* refers to the consistency of conclusions; for example, whether an examiner makes the same conclusion if presented with the same evidence on different occasions. Two of the recommendations from this study were to establish rigorously-validated laboratory procedures and “develop automated techniques capable of enhancing forensic technologies (National Research Council, 2009).”

A number of studies assess the reliability and repeatability of a firearm and toolmark examination (non-exhaustively: DeFrance and Arsdale (2003); Hamby et al. (2009); Fadul et al. (2011a); Stroman (2014); Baldwin et al. (2014); Smith et al. (2016); Mattijssen et al. (2020)). These studies indicate that examiners have a low error rate when comparing evidence obtained under controlled conditions (i.e., for which ground-truth is known). However, as pointed out in a 2016 report from the President’s Council of Advisors on Science and Technology, many of these studies, save Baldwin et al. (2014), were not “appropriately designed to test the foundational validity and estimate reliability (President’s Council of Advisors on Sci. & Tech., 2016).” The report called for more properly-designed studies to establish the scientific validity of the discipline.

Due to the opacity in the decision-making process, examiners are referred to as “black boxes” in a similar sense to black-box algorithms (OSAC Human Factors Committee, 2020).

Their evidentiary conclusions are fundamentally subjective and empirical evidence suggests that conclusions may differ if examiners are presented with the same evidence on different occasions (Ulery et al., 2011, 2012). Examiners rarely need to provide quantitative justification for their conclusion. Even for qualitative justifications, it can be difficult to determine what the examiner is actually “looking at” to arrive at their conclusion (Ulery et al., 2014). This suggests the need to supplement these black box decisions with transparent, objective techniques that quantitatively measure the similarity between pieces of evidence. As stated in President’s Council of Advisors on Sci. & Tech. (2016), efforts should be made to “convert firearms analysis from a subjective method to an objective method” including “developing and testing image-analysis algorithms for comparing the similarity of tool marks.” This work focuses on the development of an algorithm for comparing breech face impressions on cartridge cases.

1.2 Forensic Comparison Pipelines

Recent work in many forensic disciplines has focused on the development of algorithms to measure the similarity between pieces of evidence including glass (Curran et al., 2000a; Park and Tyner, 2019; Tyner et al., 2019), handwriting (Crawford, 2020), shoe prints (Park and Carriquiry, 2020), ballistics (Hare et al., 2017; Tai and Eddy, 2018), and toolmarks (Hadler and Morris, 2017; Krishnan and Hofmann, 2018). These algorithms often result in a numerical score for two pieces of evidence. A numerical score can add more nuance to an evidentiary conclusion beyond simply stating whether the evidence originated from the same source as would be the case in binary classification. For example, a larger similarity scores implies the evidence is more similar. However, an examiner must ultimately reach one of two conclusions (or three, if admitting inconclusives). Whether a conclusion should be based solely on an algorithm’s similarity score or if an examiner should incorporate the similarity score into their own decision-making process is still up for debate (Swofford and

Champod, 2021). In this work we view forensic comparison algorithms as a supplement to, rather than a replacement of, the traditional forensic examination.

We treat forensic comparison algorithms as evidence-to-classification “pipelines.” Broadly, the steps of the pipeline include Rice (2020):

1. capturing a digital representation of the evidence,
2. pre-processing this representation to isolate or emphasize a region of interest of the evidence,
3. comparing regions of interest from two different pieces of evidence to obtain a (perhaps high-dimensional) set of similarity features,
4. combining these features into a low-dimensional set of similarity scores, and
5. defining a classification rule based on these similarity features.

We add to this structure the emphasis that each step of the pipeline can be further broken-down into modularized pieces. For example, the pre-processing step may include multiple sub-procedures to isolate a region of interest of the evidence. Figure 1.7 shows two possible variations of the cartridge case comparison pipeline as well as the parameters requiring manual specification and alternative modules. The benefits of this modularization include easing the process of experimenting with different parameters/sub-procedures and improving the comprehensibility of the pipeline.

In the following sections, we detail recent advances to each of the five steps in the pipeline outlined above. We narrow our focus to advances made in comparing firearms evidence.

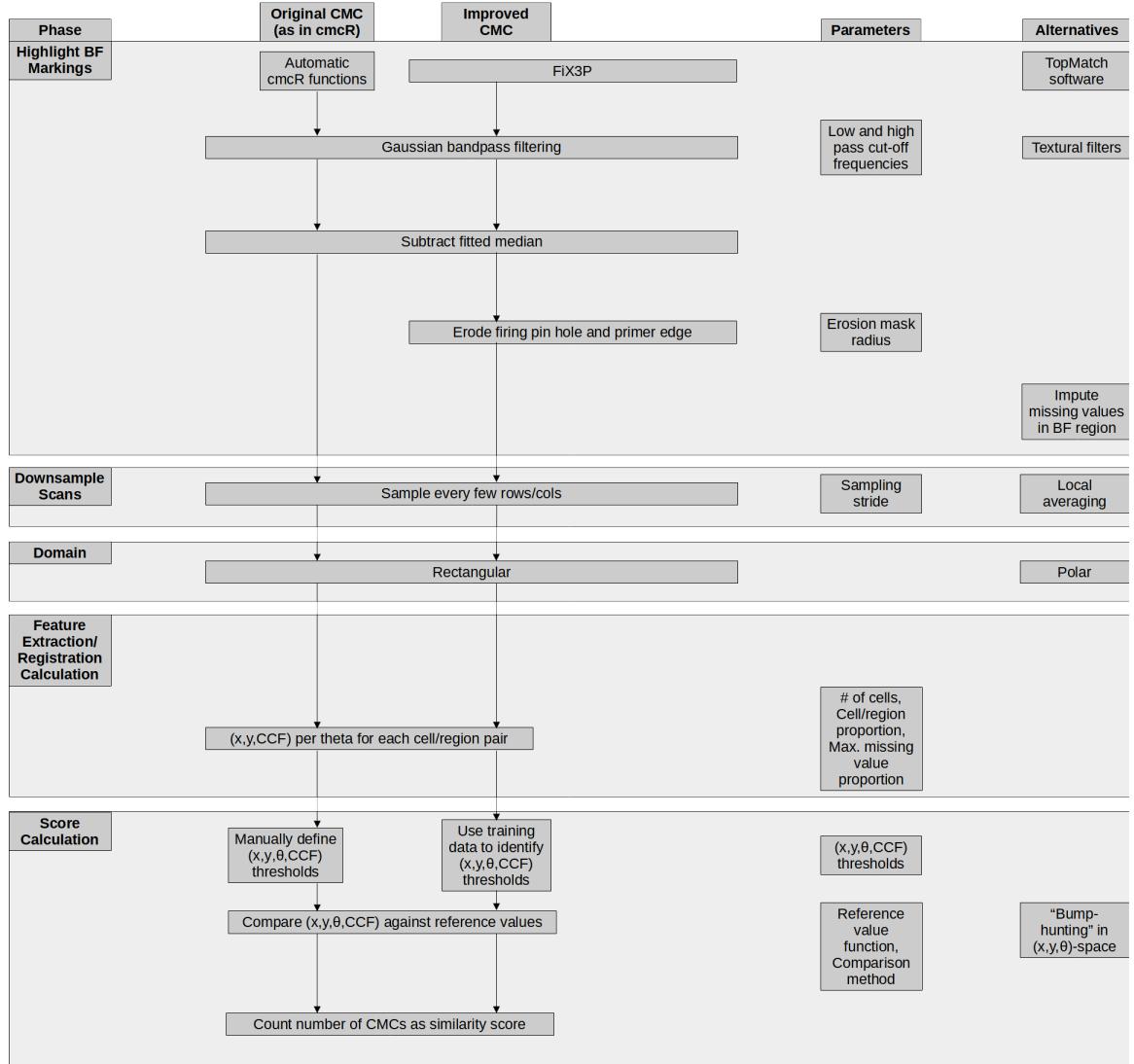


Figure 1.7: Variations upon the cartridge case comparison pipeline. The first two columns detail the pipeline with different sub-procedures. The third column shows the parameters that require manual specification at each step. The fourth column shows alternative processing steps that could replace steps in the existing pipeline.

1.2.1 Digital Representations of Evidence

Digital representations of cartridge case evidence commonly come in two forms: 2D optical images or 3D topographic scans. A common way to take 2D optical images is to take a picture of the cartridge case surface lit up under a microscope, implying a dependence on the lighting conditions under which the picture was taken. Some recent work has focused on comparing 2D optical images (Tai and Eddy, 2018; Tong et al., 2014), although the use of 3D microscopes has recently become more prevalent to capture the surface of ballistics evidence.

Using a 3D microscope allows for the scanning of surfaces at the micron (or micrometer) level under light-agnostic conditions (Weller et al., 2012). Figure 1.8 shows a 2D image and 3D topography of the same cartridge case primer from Fadul et al. (2011a).

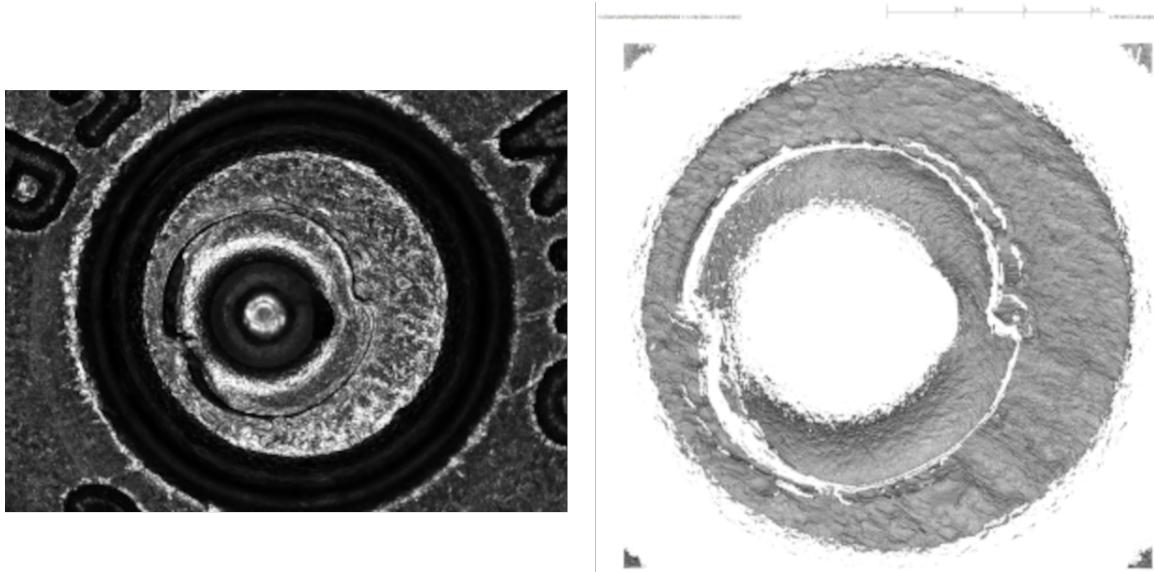


Figure 1.8: A cartridge case captured using 2D confocal reflectance microscopy (left) and 3D disc scanning confocal microscopy (right).

Recently, Cadre ForensicsTM introduced the TopMatch-3D High-Capacity Scanner (Weller et al., 2015). Figure 1.9 shows a TopMatch scanner with a tray of 15 fired cartridge cases (Cadre Forensics, 2019). This scanner collects images of a gel pad under

various lighting conditions into which the cartridge case surface is impressed. Proprietary algorithms combine these images into a regular 2D array called a *surface matrix*. Elements of the surface matrix represent the relative height value of the associated surface. The physical dimensions of these scans are about 5.5 mm^2 captured at a resolution of 1.84 microns per pixel (1000 microns equals 1 mm).

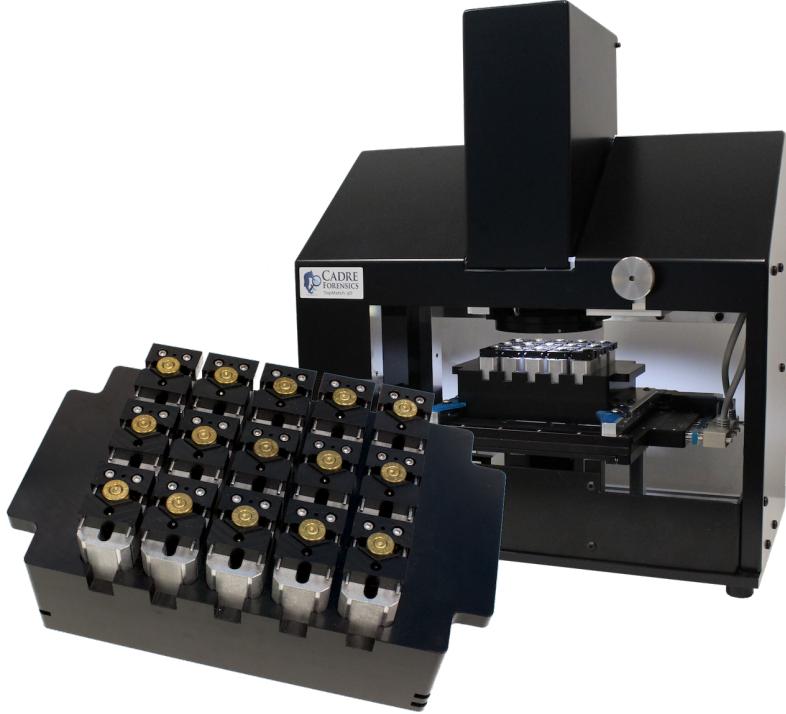


Figure 1.9: The TopMatch-3D High-Capacity Scanner from Cadre Forensics™. The scanner captures topographic scans of a gel pad into which a cartridge case surface is impressed.

The ISO standard x3p file format is commonly used to save 3D scans (ISO 25178-72(2017), 2017). An x3p is a container consisting of a single surface matrix representing the height values of the surface and metadata concerning the parameters under which the scan was taken as shown in Figure 1.10 (Zheng et al., 2020). A number of studies suggest that 3D topographic scans of cartridge case surfaces lead to more accurate classifications than 2D optical images of the same evidence (Tai, 2019; Tong et al., 2014; Song et al., 2014).

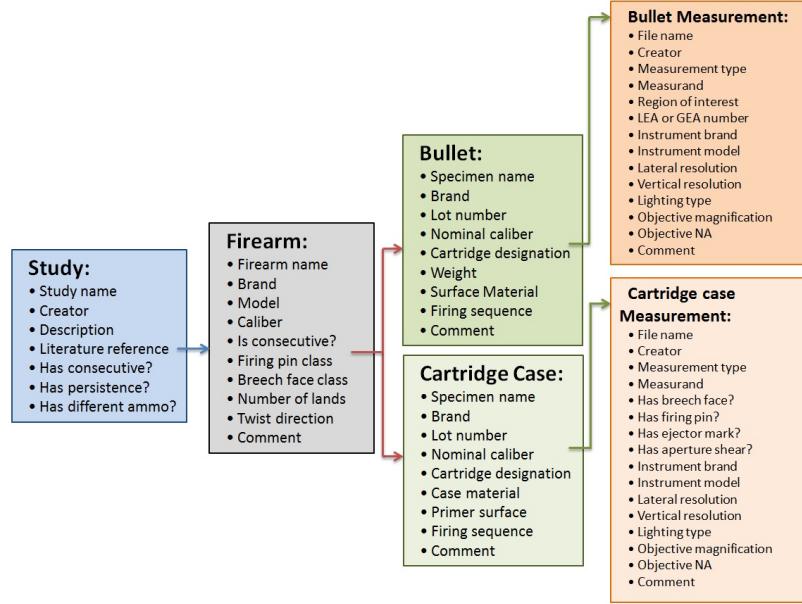


Figure 1.10: The hierarchy of information stored in the x3p file format for both bullet and cartridge case evidence.

1.2.2 Pre-processing Procedures for Forensic Data

After obtaining a surface's digital representation, we next want to isolate regions of the surface containing distinguishable markings. Figure 1.8 shows an example of a 2D image and 3D scan of the same cartridge case. In both representations, the corners of these arrays include regions of the cartridge case surface outside of the primer. The center of the cartridge case primer contains an impression left by the firing pin during the firing process. We wish to isolate the annular breech face region around the firing pin impression from the rest of the captured surface.

Both the 2D optical and 3D topographic representations of cartridge case surfaces are fundamentally pictorial in nature. As such, breech face impression isolation commonly relies on image processing and computer vision techniques. Tai and Eddy (2018) uses a combination of histogram equalization, Canny edge detection, and morphological operations to isolate breech face impressions in 2D images. A Gaussian filter is another common tool to emphasize breech face impressions. Tong et al. (2014) apply a low-pass Gaussian filter to

remove noise via a Gaussian-weighted moving average operation. Chu et al. (2013) and Song et al. (2018) use a bandpass Gaussian filter to simultaneously remove noise and unwanted global structure from the scan. Song et al. (2014) and Chen et al. (2017) use a “robust” variant of the Gaussian filter to omit outliers from the scan (ISO 16610-71(2014), 2014).

Instead of automatic procedures, others have used subjective human intervention to isolate the breech face impressions. For example, Song et al. (2018) performed “manually trimming to extract the breech face impression of interest” on a set of cartridge case scans. In Roth et al. (2015), examiners manually identify the borders of the breech face impression region by placing points around an image of the cartridge case primer.

1.2.3 Forensic Data Feature Extraction

After isolating the breech face impressions, we compare two pre-processed cartridge case scans and compute a set of similarity features. Because the cartridge cases at this point are represented as high-dimensional matrices, this process can be thought of as a dimensionality reduction of the high-dimensional surface arrays to a set of similarity statistics.

A variety of features have been proposed to quantify the similarity between two cartridge case surface arrays. Tai and Eddy (2018) propose calculating the cross-correlation function (CCF) value between two cartridge cases across a grid of rotations. The cross-correlation function measures the similarity between two matrices for every translation of one matrix against the other. For two matching cartridge cases, we assume that the CCF will be largest after aligning the cartridge cases surfaces by their shared breech face impressions. Conversely, we expect the CCF to be relatively small for two non-matching cartridge cases no matter the alignment. Riva and Champod (2014) propose combining the CCF between two aligned scans with the element-wise median Euclidean distance and median difference between the normal vectors at each point of the surface. Riva et al. (2016) and

Riva et al. (2020) applied Principal Component Analysis to reduce these three features down to two principal components for the sake of fitting a 2D kernel density estimator.

Pertinent to this work is the cell-based comparison procedure originally outlined in Song (2013). The underlying assumption of Song (2013) is similar to that of Tai and Eddy (2018): that two matching cartridge cases will exhibit higher similarity when they are close to being correctly aligned. While Tai and Eddy (2018) measured similarity using the CCF between the two full scans, Song (2013) proposes partitioning the scans into a grid of “correlation cells” and counting the number of similar cells between the two scans. The rationale behind this procedure is that many cartridge case scans have only a few regions with discriminatory markings. As such, comparing full scans may result in a lower correlation than if one were to focus on the highly-discriminatory regions. In theory, dividing the scans into cells allows for the identification of these regions. After breaking a scan into a grid of cells, each cell is compared to the other scan to identify the rotation and translation, known together as the *registration*, at which the cross-correlation is maximized. Song (2013) assumes that the cells from a truly matching pair of cartridge cases will “agree” on their registration in the other scan. Song (2013) referred to the procedure of counting the number of similar cells the “Congruent Matching Cells” method. Chapter 2 contains more details of this procedure.

1.2.4 Similarity Scores & Classification Rules for Forensic Data

Following feature extraction, the dimensionality of these features is further reduced to a low-dimensional, usually univariate, similarity score. We can define a decision boundary based on the value of the similarity score to classify cartridge case pairs as matching or non-matching.

After calculating the CCF across various possible registrations, Tai and Eddy (2018) propose using the maximum observed CCF value as the univariate similarity score. They

perform binary classifications by setting a CCF threshold above which pairs are classified as “matches” and below which as “non-matches.” Tai (2019) proposes setting a CCF cut-off that maximizes the precision and recall in a training set of pairwise comparisons.

Riva et al. (2016, 2020) use a training set to fit two 2D kernel density estimates to a set of features from matching and non-matching comparisons. Using these estimates, they compute a score-based likelihood ratio (SLR), which can be interpreted as a similarity score (Garton et al., 2020).

For the Congruent Matching cells method, Song (2013) proposes using the number of cells that agree on a registration, the “congruent matching” cells, as a similarity score. The criteria used to define “congruent matching” cells has changed across papers (Song et al., 2014; Tong et al., 2014, 2015; Chen et al., 2017) and will be discussed in greater detail in Chapter 2. The authors of these papers have consistently used six congruent matching cells as a decision boundary to distinguish matching and non-matching cartridge case pairs.

Zhang et al. (2020) applies the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm (Ester et al., 1996) to the features from the cell-based comparison procedure to determine if any clusters form amongst the per-cell estimated registration values. This is based on the assumption that any cells that come to a consensus on their registration should form a cluster in translation (x, y) and rotation θ space. Zhang et al. (2020) proposes a binary classifier based on whether any clusters are identified by the DBSCAN algorithm (Ester et al., 1996). If a cluster is found for a particular pairwise comparison, then that pair is classified as a “match” and otherwise as a “non-match.”

Apart from the algorithms described in Tai and Eddy (2018) and Tai (2019), the authors of these comparison algorithms have not provided publicly available code or data. As such, although the results reported in associated papers are promising, it is difficult or impossible for other researchers to verify or reproduce the findings. Results must be

reproducible to be accepted by others in any scientific domain. In the next section, we discuss recent challenges and opportunities in computationally reproducible research.

1.2.5 Reproducibility of Comparison Pipelines

National Academy of Sciences, Engineering, and Medicine (2019) defines *reproducibility* as “obtaining consistent computational results using the same input data, computational steps, methods, code, and conditions of analysis.” While not exact in their definition of “consistent,” the authors assert that, barring a few exceptions, it is reasonable to expect that the results obtained by a second researcher, after applying the exact same processing steps to the exact same data, be the exact same as the original results. In either case, they assert that “a study’s data and code have to be available in order for others to reproduce and confirm results.” Given data and code, researchers are able to verify the results, incorporate the materials into their own research, and improve or accelerate discovery (Stodden et al., 2018a).

A number of studies indicate that computationally reproducible research is sparse across various disciplines. Stodden et al. (2018a) and Stodden et al. (2018b) studied the reproducibility of articles sampled from the *Journal of Computational Physics* and the journal *Science*, respectively. In the former, Stodden et al. (2018a) found that zero of 306 randomly selected articles from the *Journal of Computational Physics* were “straightforward to reproduce with minimal effort” and, at best, that five articles were “reproducible after some tweaking.” In the latter, Stodden et al. (2018b) found that only 3 of 204 randomly selected articles from *Science* were “straightforward to reproduce with minimal effort;” despite a journal policy requiring that all code and data used in the paper be made available to any reader. Similar findings were found in Chang and Li (2022) (29 of 59 economic papers reproducible), Iqbal et al. (2016) (zero of 268 biomedical papers provided raw data and 1 in 268 linked to a full study protocol), Duvendack et al. (2015) (50% or more published articles include data or code in only 27 of 333 economics journals), and Gundersen et al. (2018) (24

of 400 AI conference papers included code). A common recommendation amongst these authors is to establish of rigorous tools and standards to promote reproducibility. This includes making code and data used in a paper easily-accessible to readers.

Infrastructure already exists to ease the process of developing, maintaining, and sharing open-source code and data. Data repositories such as the NIST Ballistics Toolmark Research Database (Zheng et al., 2020) provide open access to raw data. Grüning et al. (2018) discuss the use of package managers such as Conda (<https://anaconda.org/anaconda/conda>), container software such as Docker (<https://www.docker.com/>), and virtual machine software to preserve the entire data analysis environment in-perpetuity. For situations in which VMs or containers aren't available, software such as the `manager` R package allows users to “compare package inventories across machines, users, and time to identify changes in functions and objects (Rice, 2020).” Piccolo and Frampton (2016) reference repositories like Bioconductor (Huber et al., 2015) that make it easy to document and distribute code. Further, software such as the `knitr` R package (Xie, 2014a) enable “literate programming” in which prose and executed code can be interwoven to make it easier to understand the code’s function. These tools make data, code, and derivative research findings more accessible, in terms of both acquisition and comprehensibility, to consumers and fellow researchers.

1.3 Diagnostic Tools

Forensic examiners often provide expert testimony in court cases. As part of this testimony, an examiner is allowed to provide facts about the outcome of a forensic examination and their opinion about what the results mean. A party to a court may challenge the examiner on the validity of the underlying scientific method or whether they interpreted the results correctly (American Academy of Forensic Sciences, 2021). In these situations, examiners need to explain the process by which they reached an evidentiary conclusion to the fact finders of the case; namely, the judge or jury. As algorithms are more often used

in forensic examinations, the technical knowledge required to understand and explain an algorithm to lay-people has increased. Indeed, even the most effective algorithms may be moot if an examiner can't explain the algorithm in their testimony. While in some cases the authors of the algorithm have been willing to provide testimony to establish the validity of the algorithm (Indiana County Court of Common Pleas, 2009), this will become less viable as algorithms become more prevalent.

The resources required to educate examiners on the theory and implementation of highly technical algorithms makes additional training seem currently implausible. An alternative is to develop algorithms from the ground-up to be intuitive for examiners to understand and explain to others. *Explainability* refers to the ability to identify the factors that contributed to the results of an algorithm (Belle and Papantonis, 2021). For example, understanding why a classifier predicted one class over another. *Diagnostics* are tools to explain or justify the behavior of a model or algorithm in specific instances. Myriad diagnostic tools exist to explain the results of an algorithm. These range from identifying instances of the training set that illuminate how the model operates (Deng, 2018) to fitting more transparent models that accurately approximate the complex model (Puiutta and Veith, 2020) to explaining the behavior of the algorithm in a small region of interest of the prediction space (Ribeiro et al., 2016; Goode and Hofmann, 2021). Many of these methods require additional technical knowledge to interpret these explanations.

1.3.1 Visual Diagnostics

A less technical approach is to use visualizations that facilitate understanding of model behavior. Properly constructed visuals enable both exploratory data analysis and diagnostics (Buja et al., 2009), which are critical steps in the data analysis process for anticipating and assessing model fit. Given that many of the procedures by which cartridge case evidence is captured, processed, and compared are based on image processing techniques, a visual diagnostic is an intuitive mode of explanation for researchers and lay-people alike. As

stated in Cleveland (1994), “graphical methods tend to show data sets as a whole, allowing us to summarize the behavior and to study detail. This leads to much more thorough data analyses.”

Numerical statistics summarize the behavior of data, but miss the detail referenced in Cleveland’s quote (Telea, 2014). To illustrate this, consider the famous data sets from (Anscombe, 1973) known as Anscombe’s quartet. The two variables in each data set are plotted against one another in Figure 1.11. There are clear differences in the relationship between x and y across these four data sets.

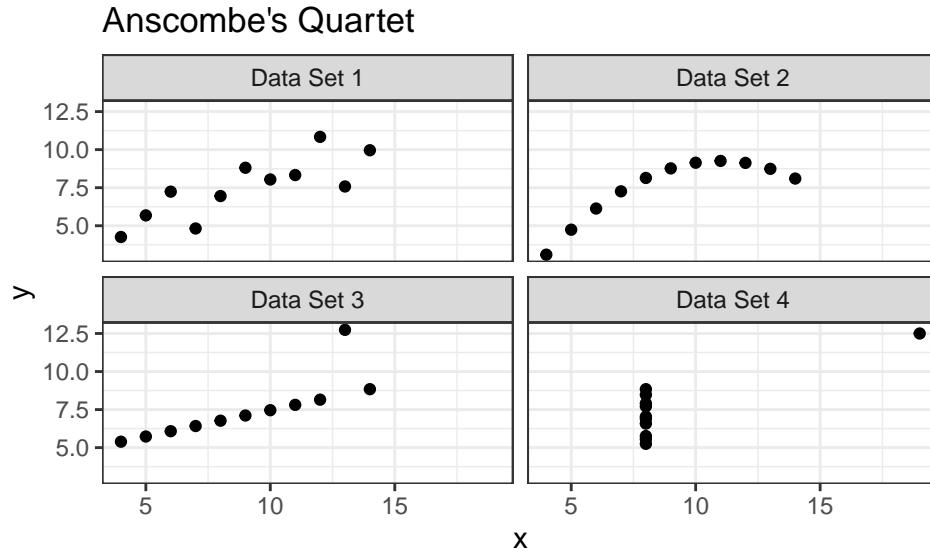


Figure 1.11: A visualization of Anscombe’s quartet. Despite there being obvious differences between these four data sets, their summary statistics are nearly identical

Despite these differences, Table 1.1 shows that summary statistics, namely the first two moments, are identical. This demonstrates that visual diagnostics can be more effective at uncovering data behavior than summary statistics (at least low-order moments).

Given the pivotal role that visual diagnostics play in the data analysis pipeline, we now consider best practices in creating data visualizations. Human brains are wired for seeing patterns and differences, and for understanding spatial relationships from this (Telea, 2014). As such, an effective visual diagnostic or data visualization is one that conveys patterns

Data Set	\bar{x}	S.D. x	\bar{y}	S.D. y
1	9	3.32	7.5	2.03
3	9	3.32	7.5	2.03
3	9	3.32	7.5	2.03
4	9	3.32	7.5	2.03

Table 1.1: Moments of the two variables in Anscombe's quartet.

quickly and easily, and with minimal scope for understanding. Arising originally from a psychological theory of perception, the Gestalt Laws of Perceptual Organization (Goldstein and Brockmole, 2016) summarize important considerations when constructing statistical graphics. The Gestalt laws are as follows:

- **Pragnanz - the law of simplicity:** Every stimulus pattern is seen in such a way that the resulting structure is as simple as possible.
- **Proximity:** Things that are near each other appear to be grouped together.
- **Good Continuation:** Points that, when connected, result in straight or smoothly curving lines are seen as belonging together, and the lines tend to be seen in such a way as to follow the smoothest path.
- **Similarity:** Similar things appear to be grouped together.
- **Common Region:** Elements that are within the same region of space appear to be grouped together.
- **Uniform Connectedness:** A connected region of visual properties, such as the lightness, color, texture, or motion, is perceived as a single unit.
- **Synchrony:** Visual events that occur at the same time are perceived as belonging together.
- **Common Fate:** Things that are moving in the same direction appear to be grouped together.

- **Familiarity:** Things that form patterns that are familiar or meaningful are likely to become grouped together.

These laws provide guidance on how to construct a visual that concisely conveys a pattern or difference in data. For data visualization, additional laws include (Midway, 2020):

- **Use and Effective Geometry:** Choose a geometry (shape and features of a statistical graphic) that is appropriate to the data.
- **Colors Always Mean Something:** Colors in visuals can convey groupings or a range of values.

Figure 1.12 depicts a case study of the Gestalt principles in practice. The plot shows the weight over time of chicks fed one of two experimental diets (Crowder and Hand, 1990). Individual points represent the weight of a single chick on a particular day. Connected points represent the weight for a single chick over time. This is an example of using an effective geometry (point & line graph to represent time series) along with the Gestalt law of Good Continuation. We further apply the Gestalt law of Common Region to facet the data set into plots based on diet. This implicitly communicates to the audience that the weights of two diet groups of chicks is expected to differ. Indeed, appealing to the Gestalt law of Uniform Connectedness, the “motion” of the grouped time series suggests that chicks given Diet 2 tend to gain weight more rapidly than those given Diet 1. This may suggest a particular modeling structure for these time series (e.g., diet fixed effect) or the need to assess the experimental design to ensure that the assumption that the chicks were randomly sampled from the same population is appropriate. We see how such a plot can be used for both exploratory data analysis or as a post-hoc diagnostic tool. Alternative to faceting, the time series from these two diet groups could be combined into a single plot and distinguished by color.

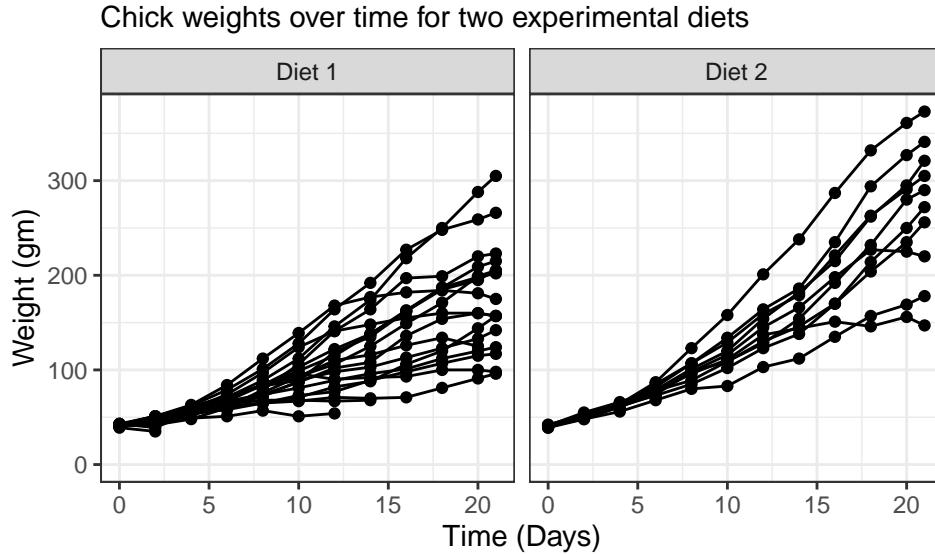


Figure 1.12: An example of a statistical graphic that uses the Gestalt Laws of Perceptual Organization to communicate data findings.

The R programming language (R Core Team, 2017) provides a variety of tools to create visual diagnostics. Among the most robust of these tools is the `ggplot2` package (Wickham, 2009). This package extends the “Grammar of Graphics” introduced in Wilkinson (2005) to provide a user-friendly structure to create statistical graphics. We use the `+` operator to “layer” features of a statistical graphic (e.g., elements, transformations, guides, labels) on a blank canvas. 1.13 along with the accompanying code chunk demonstrates how to create a residual plot from a simple linear regression using the `ggplot2` package. This visual diagnostic allows the analyst or audience to determine whether the homoscedasticity or linear form assumptions underlying simple linear regression are met. For those willing to learn the “grammar,” the code used to create these statistical graphics can easily be re-used and tweaked to fit a specific application. 1

```
lmFit <- lm(formula = rating ~ complaints, data = datasets::attitude)
```

```
library(ggplot2)
```

```
ggplot(data = data.frame(Complaints = datasets::attitude$complaints,
                        Residuals = lmFit$residuals)) +
  geom_point(aes(x = Complaints, y = Residuals)) +
  geom_hline(yintercept = 0, linetype = "dashed") +
  labs(x = "% in-favor of handling of employee complaints")
```

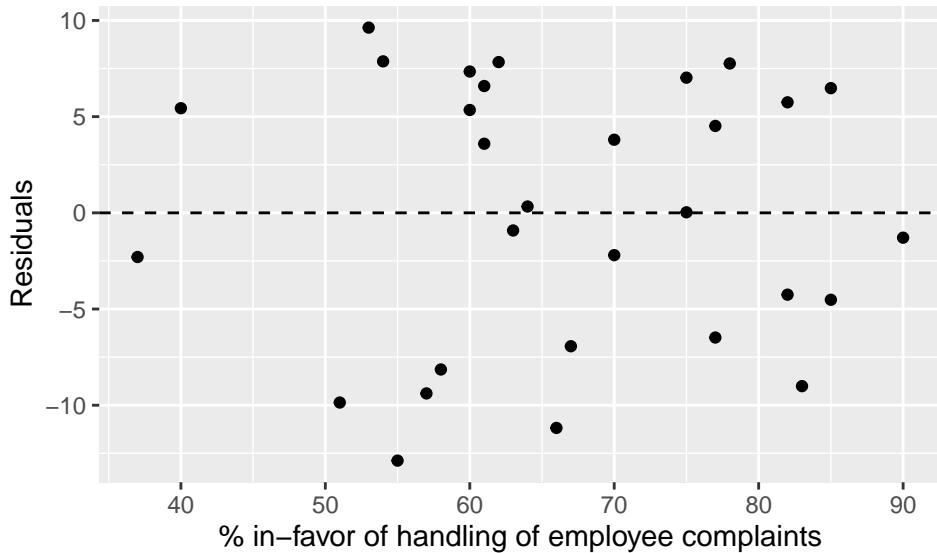


Figure 1.13: An example of using the ggplot2 package to construct a residual plot from a simple linear regression. The features of the statistical graphic are combined layer-by-layer using the `+` operator as we see in the accompany code chunk.

Properly constructed visual diagnostics provide the audience with a nuanced yet intuitive explanation of the behavior of a model or algorithm that summary diagnostic statistics may not convey. Tools like the ggplot2 package provide a coherent, thorough infrastructure for creating such visual diagnostics. However, the tools discussed thus far are useful for creating *static* visualizations. In the next section, we discuss the benefits of making a visual diagnostic interactive to user input.

1.3.2 Interactive Diagnostics

Interactive diagnostic tools encourage both expert and lay users to engage with an analysis pipeline that otherwise may be technically or conceptually inaccessible. Rather than answering a question posed by the author of a plot as a static plot does, such interactive diagnostic tools enable the audience to formulate and answer their own questions. This leads to deeper engagement with the data (Telea, 2014). While the `ggplot2` package eases the process of constructing visual diagnostics, software such as the `shiny` R package (Chang et al., 2021) enables the consumer of the diagnostic to interact with the visualizations and underlying data. The `shiny` package provides tools for using R to build web applications run on HTML, CSS, and JavaScript. Among other functionality, these applications allow users to upload or create their own data, set parameters for an analysis, interact with visualizations or data sets (e.g., by hovering to display a tooltip), and export their analyses in various file formats (Beeley and Sukhdeve, 2018).

Several recently-released software provide interactive diagnostic applications for firearms and toolmarks evidence. Most notable of these software is the Virtual Comparison Microscopy application from Cadre Forensics™. In contrast to traditional Light Comparison Microscopy (LCM) that uses a comparison microscope, this software displays digital representations of the cartridge case surface on a computer screen. Figure 1.14 shows a screenshot of comparing two cartridge case surfaces (Chapnick et al., 2020). The functionality shown allows the user to manually annotate the surfaces of the two cartridge cases to identify similar and different markings. For example, the user has selected a shade of blue to represent similarities between the two surfaces. Conversely, shades of yellow and red represent differences between the two surfaces. This sort of interactivity allows the user to customize their analysis more effectively than they could with a static visualization. Further, we can save a history of the annotations for further analysis. These annotations are a visual diagnostic tool that allows others to understand the specific patterns that the examiner looks at during an examination. Another major benefit of using VCM over LCM

is the ability to share scans over the internet rather than sending the physical specimen to another lab, which takes time and may damage the specimen. Duez et al. (2017), Chapnick et al. (2020), and Knowles et al. (2021) all demonstrate that performing forensic examinations using such VCM technology yields equally, if not more, accurate conclusions compared to traditional LCM methods.

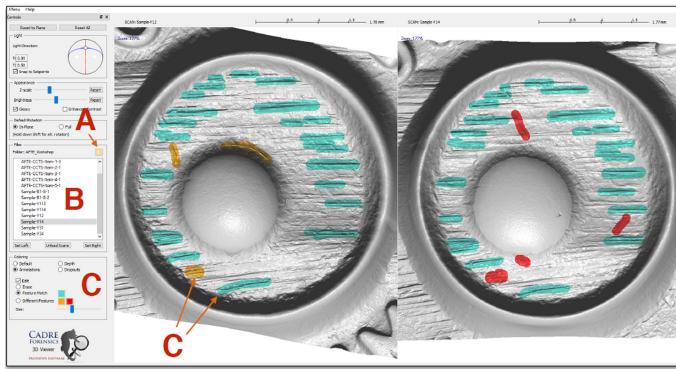


Figure 1.14: A screenshot of the TopMatch-3D™ Virtual Comparison Microscopy software. In this example, similar and different markings on the cartridge case scans are manually annotated by the user using shades of blue and yellow/red, respectively.

In Chapter 3, we introduce a suite of static and interactive visual diagnostic tools. We discuss how these visual diagnostic tools can be used by both researchers and practitioners to understand the behavior of automatic cartridge case comparison algorithms.

1.4 Automating and Improving the Cartridge Case Comparison Pipeline

In this section, we review preliminaries needed to understand various sub-routines of the cartridge case comparison pipeline.

1.4.1 Image Processing Techniques

We first review image processing and computer vision algorithms used in cartridge case comparison algorithms. Throughout this section, let $A, B \in \mathbb{R}^{k \times k}$ denote two images for $k > 0$. We use lowercase letters and subscripts to denote a particular value of a matrix: a_{ij}

is the value in the i -th row and j -th column, starting in the top-left corner, of matrix A . In our application, A and B represent the surface matrices of two cartridge cases.

1.4.1.1 Image Registration

Image registration involves transforming image B to align best with image A (or vice versa) (Brown, 1992). In our application, this transformation is composed of a discrete translation $(m^*, n^*) \in \mathbb{Z}^2$ and rotation by $\theta^* \in [-180^\circ, 180^\circ]$. Together, we refer to (m^*, n^*, θ^*) as the “registration” of image B to A . To determine the optimal registration, we calculate the *cross-correlation function* between A and B , denoted $(A \star B)$, which measures the similarity between A and B for every possible translation of B . The CCF between A and B is a 2D array of dimension $2k - 1 \times 2k - 1$ where the value of the m, n -th element is given by:

$$(a \star b)_{mn} = \sum_{i=1}^k \sum_{j=1}^k a_{mn} \cdot b_{i+m, j+n}$$

where $1 \leq m, n \leq 2k - 1$. The value $(a \star b)_{mn}$ quantifies the similarity between A and B after B is translated m elements horizontally and n elements vertically.

A natural choice for aligning A and B is the translation that maximizes the CCF. However, we must also consider that B may also need to be rotated to align optimally with A . We therefore compute the maximum CCF value across a range of rotations of B . Let B_θ denote B rotated by an angle θ and $b_{\theta mn}$ the m, n -th element of B_θ . Then the estimated registration (m^*, n^*, θ^*) is:

$$(m^*, n^*, \theta^*) = \arg \max_{m, n, \theta} (a \star b_\theta)_{mn}.$$

In practice, we consider a discrete range of rotations $\Theta \subset [-180^\circ, 180^\circ]$. The registration procedure is given by:

1. For each $\theta \in \Theta$:

1.1 Rotate image B by θ to obtain B_θ .

1.2 Calculate the CCF between A and B_θ .

1.3 Determine the translation $[m_\theta^*, n_\theta^*]$ at which the CCF is maximized. Also, record the CCF value associated with this translation.

2. Across all $\theta \in \Theta$, determine the rotation θ^* at which the largest CCF value is achieved.
3. The estimated registration consists of rotation θ^* and translation $[m^*, n^*] \equiv [m_{\theta^*}^*, n_{\theta^*}^*]$.

In this instance, we refer to image A as the “reference” and B , the image aligned to the reference, as the “target.” We represent the transformation to register B to A element-wise where the index i, j maps to i^*, j^* by:

$$\begin{pmatrix} j^* \\ i^* \end{pmatrix} = \begin{pmatrix} n^* \\ m^* \end{pmatrix} + \begin{pmatrix} \cos(\theta^*) & -\sin(\theta^*) \\ \sin(\theta^*) & \cos(\theta^*) \end{pmatrix} \begin{pmatrix} j \\ i \end{pmatrix}.$$

Under this transformation, the value b_{ij} now occupies the the i^*, j^* -th element. In practice, we use *nearest-neighbor interpolation* meaning i^* and j^* are rounded to the nearest integer.

Based on the definition given above, the CCF is computationally taxing. In image processing, it is common to use an implementation based on the Fast Fourier Transform (Brown, 1992). This implementation leverages the Cross-Correlation Theorem, which states that for images A and B the CCF can be expressed in terms of a frequency-domain pointwise product:

$$(A \star B)[m, n] = \mathcal{F}^{-1} \left(\overline{\mathcal{F}(A)} \odot \mathcal{F}(B) \right) [m, n]$$

where \mathcal{F} and \mathcal{F}^{-1} denote the discrete Fourier and inverse discrete Fourier transforms, respectively, and $\overline{\mathcal{F}(A)}$ denotes the complex conjugate (Brigham, 1988). Because the product on the right-hand side is calculated pointwise, this result allows us to trade the moving sum computations from the definition of the CCF for two forward Fourier transformations, a pointwise product, and an inverse Fourier transformation. The Fast Fourier Transform (FFT) algorithm can be used to reduce the computational load considerably.

Figure 1.15 shows an example of two images A and B of dimension 100×100 and 21×21 , respectively. The white boxes in both of the images are of dimension 10×10 . The box in image A is centered on index $[30,50]$ while the box in image B is centered on index $[11,11]$. The right image shows the result of calculating the CCF using image A as reference and B as template. The CCF achieves a maximum of 1, indicating a perfect match, at the translation value of $[m^*, n^*] = [22, -2]$. This means that if image B were overlaid onto image A such that their center indices coincided, then image B would need to be shifted 22 units “up” and 2 units “left” to match perfectly with image A .

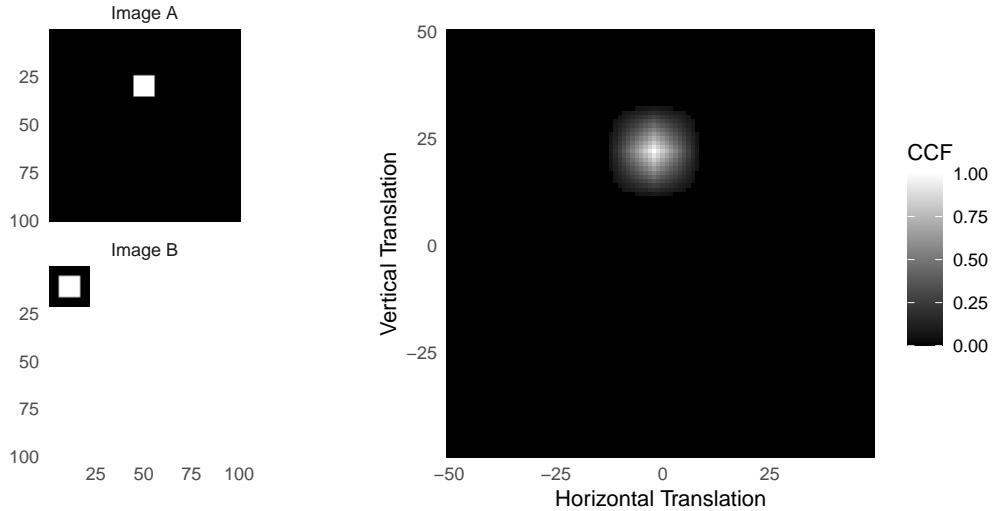


Figure 1.15: (Left) A reference image A and template image B both featuring a white box of dimension 10×10 . (Right) The cross-correlation function (CCF) between A and B . The index at which the CCF is maximized represents the translation at which A and B are most similar.

1.4.1.2 Gaussian Filters

In image processing, a Gaussian filter (equivalently, blur or smoother) is a mathematical operator that imputes the values in an image using a locally-weighted sum of surrounding values. We use a *lowpass* Gaussian filter to smooth the surface values of a cartridge case

scan. The weights are dictated according to the Gaussian function of a chosen standard deviation σ given by:

$$f(n, m; \sigma) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{1}{2\sigma^2}(n^2 + m^2)\right).$$

It is common to populate a 2D array with the values of the Gaussian function treating the center index as the origin. Such an array is called a *kernel*. An example of a 3×3 Gaussian kernel K with standard deviation $\sigma = 1$ is given below.

$$K = \begin{pmatrix} 0.075 & 0.124 & 0.075 \\ 0.124 & 0.204 & 0.124 \\ 0.075 & 0.124 & 0.075 \end{pmatrix}.$$

For an image A and Gaussian kernel K with standard deviation σ , the lowpass filtered version of A , denoted $A_{lp, \sigma}$ is given by:

$$A_{lp, \sigma}[m, n] = \mathcal{F}^{-1}(\mathcal{F}(A) \odot \mathcal{F}(K))[m, n].$$

This operation, known as *convolution*, is extremely similar to the calculation of the CCF defined in the Image Registration section (ISO 16610-21, 2011).

From left to right, Figure 1.16 shows an image A of a box injected with Gaussian noise (noise standard deviation $\sigma_n = 0.3$) followed by the application of various Gaussian filters. In the middle of Figure 1.16, we see that the lowpass filter (kernel standard deviation $\sigma_k = 2$) recovers some of the definition of the box by “smoothing” some of the Gaussian noise.

If a lowpass filter smooths values in an image, then a *highpass* filter performs a “sharpening” operation. For an image A and kernel standard deviation σ , the highpass filtered version A_{hp} can be defined as:

$$A_{hp, \sigma} = A - A_{lp, \sigma}.$$

The highpass filter therefore removes larger-scale (smooth) structure from an image and retains high-frequency structure such as noise or edges. The fourth facet of Figure 1.16

shows a highpass-filtered image A . The smooth interior of the box is effectively removed from the image while the edges are preserved.

Finally, a *bandpass* Gaussian filter simultaneously performs highpass sharpening and lowpass smoothing operations. Generally, the standard deviation of the highpass kernel will be considerably larger than that of the lowpass kernel. This leads to retaining sharp edges while also reducing noise. An example of a bandpass filtered image A is shown in Figure 1.16. The edges of the box are better-preserved compared to the lowpass filter figure while the interior of the box is better-preserved compared to the highpass filter figure.

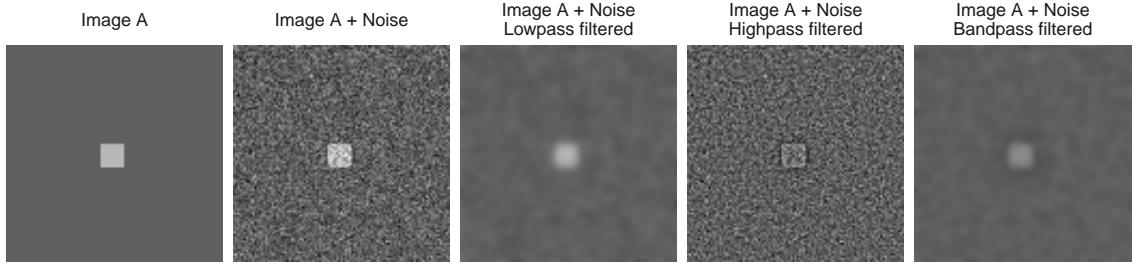


Figure 1.16: An image A of a box with Gaussian noise undergoing a lowpass, highpass, and bandpass filter operation.

Variations on the standard Gaussian filter include the “robust” Gaussian regression filter. This filter fluctuates between a filter step, which applies a Gaussian filter, and outlier step, which identifies and omits outlier observations from the next filter step (Brinkman and Bodschatwinna, 2003b). Another alternative, the “edge preserving” filter, adapts the kernel weights when approaching the boundary of an image to mitigate so-called *boundary effects* (Aurich and Weule, 1995).

We use Gaussian filters to change the values on the interior of a cartridge case surface to better emphasize breech face impressions. In the next section, we discuss applying morphological operations to change the values on the edges of a cartridge case surface.

1.4.1.3 Morphological Operations

Mathematical morphology refers to a theory and collection of image processing techniques for geometrical structures (Haralick et al., 1987). In our application, these geometrical structures are cartridge case scans; specifically, binarized versions of these scans representing whether a particular pixel contains part of the cartridge case surface. We discuss this in greater detail in Chapter 2.

Two fundamental operations in mathematical morphology are *dilation* and *erosion* (Haralick et al., 1987). For our purposes, these are both set operations on black and white, encoded as 0 and 1 respectively, images. We call the set of black and white pixels the “background” and “foreground” of the image, respectively. For an image A , let $W = \{(m, n) : A_{mn} = 1\}$ denote the foreground of A . An example of a 7×7 binary image A with $W = \{(3, 3), (3, 4), (3, 5), (4, 3), (4, 4), (4, 5), (5, 3), (5, 4), (5, 5)\}$ is given below.

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

A *structuring element* is a second, typically small, array B of ones that affects the amount of dilation or erosion applied to W within A . For simplicity, the indexing of the structuring element uses the center element as the index origin. For example, a 3×3 structuring

element is given by $B = \{(-1, -1), (-1, 0), (-1, 1), (-1, 0), (0, 0), (0, 1), (1, -1), (1, 0), (1, 1)\}$ or visually:

$$B = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

As the name suggests, a *dilation* grows the region W within image A by replacing 0-valued pixels that border W with 1. The structuring element B dictates which pixels are replaced with 1. We define the dilation of W by B , denoted $W \oplus B$, element-wise:

$$W \oplus B = \{[m, n] \in A : [m, n] = [i+k, j+l] \text{ for } [i, j] \in W \text{ and } [k, l] \in B\}$$

In our example,

$$W \oplus B = \{[3, 2], [3, 3], [3, 4], [3, 5], [3, 6], [4, 2], [4, 3], [4, 4], [4, 5], [4, 6], [5, 2], [5, 3], [5, 4], [5, 5], [5, 6]\}$$

or visually:

$$W \oplus B = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

The dilation operation by this B has the effect of growing the region W inside of A by one index in each direction.

In contrast, *erosion* has the effect of shrinking W . The erosion of W by B is:

$$A \ominus B = \{[m, n] \in A : [m, n] + [k, l] \in A \text{ for every } [k, l] \in B\}.$$

Using the same example as above, $W \ominus B = \{[3,3]\}$ or visually:

$$W \ominus B = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Erosion by this B shrinks the region W in A by one index in each direction.

Figure 1.17 shows our example represented using black and white pixels. In practice, the foreground set W may contain disconnected regions to which dilation or erosion can be independently applied.

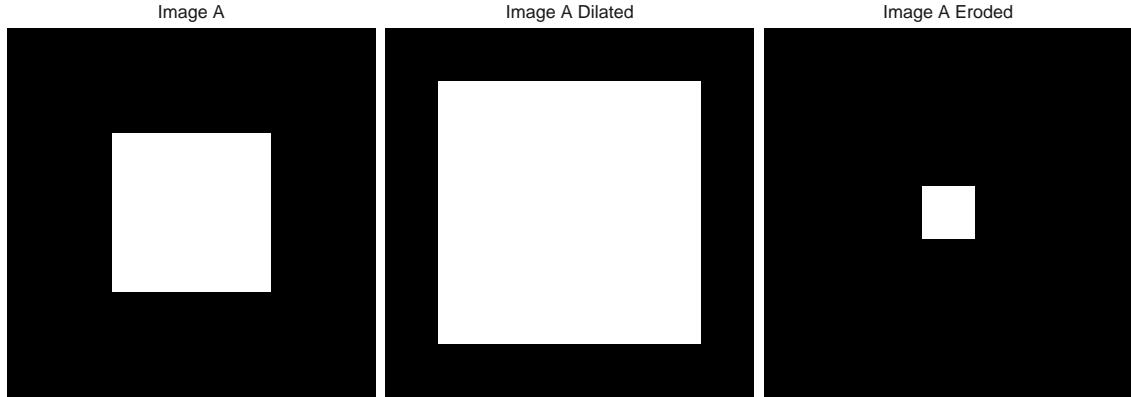


Figure 1.17: A 7×7 image A featuring a 3×3 box undergoing dilation and erosion by a 3×3 structuring element B .

This concludes our review of image processing techniques we use in subsequent chapters. Next, we discuss a clustering procedure used in Chapter 4 to calculate similarity features.

1.4.2 Density-Based Spatial Clustering of Applications with Noise

The Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm is a clustering procedure that assigns observations to clusters if they are in a region of high observation density (Ester et al., 1996). As we will see, the DBSCAN algorithm does not require the user to pre-specify the number of expected clusters as is required in common clustering algorithms like K-means. Further, the algorithm does not require that all points be assigned to a cluster.

Let D represent a $n \times p$ data set (n observations, each of dimension p) and let $x, y, z \in D$ denote three observations. The DBSCAN algorithm relies on the notion of ε -neighborhoods. Given some neighborhood radius $\varepsilon \in \mathbb{R}$ and distance metric d , y is in the ε -neighborhood of x if $d(x, y) \leq \varepsilon$. The ε -neighborhood of x is defined as the set $N_\varepsilon(x) = \{y \in D : d(x, y) \leq \varepsilon\}$. Given a minimum number of points $Minpts \in \mathbb{N}$ (notation used in (Ester et al., 1996)), observation x is called a *core point* with respect to ε and $Minpts$ if $|N_\varepsilon(x)| \geq Minpts$. Core points are treated as the “seeds” of clusters in the DBSCAN algorithm. The user must select values of ε and $Minpts$.

Figure 1.18 shows an example of a data set $D \in \mathbb{R}^{10 \times 2}$. We represent the 10 observations in D on the Cartesian plane. An ε -neighborhood using the Euclidean distance metric and $\varepsilon = 3$ is drawn around an observation x located at $(3, 2)$. Points inside the circle are neighbors of x . If, for example, $Minpts = 2$, then x would be considered a core point.

(Ester et al., 1996) introduces two relational notions, *density-reachability* and *density-connectivity*, to identify regions of high observation density. A point y is *directly density-reachable* to a point x if x is a core point and $y \in N_\varepsilon(x)$. In Figure 1.18, the observation located at $(1, 0)$ is directly density-reachable to the observation located at $(3, 2)$. More broadly, a point x_m is *density-reachable* to a point x_1 if there exists a chain of observations $x_1, x_2, \dots, x_{m-1}, x_m$ such that x_{i+1} is directly density-reachable from x_i , $i = 1, \dots, n$. Density

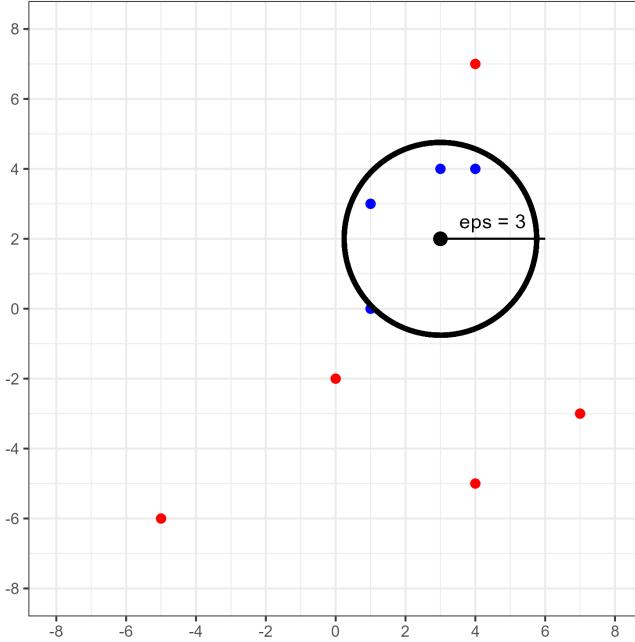


Figure 1.18: An ε -neighborhood around a observation located at $(3, 2)$ for $\varepsilon = 3$. Points are colored blue if they are neighbors to this observation and red otherwise.

reachability captures the notion of “neighbors of neighbors” for core points. The DBSCAN algorithm agglomerates density-reachable points into single clusters.

Figure 1.19 highlights three points $(1, 0)$, $(3, 2)$, and $(4, 4)$. Using $\varepsilon = 3$ and $Minpts = 2$, we see that all three of these points are core points. Further, the points at $(1, 0)$ and $(4, 4)$ are density-reachable by way of the point $(3, 2)$.

Finally, a point y is *density-connected* to a point x with respect to ε and $Minpts$ if there exists a point z such that both x and y are density-reachable to z (with respect to ε and $Minpts$). While density-reachability requires that all points in-between two points also be core points, density-connectivity extends the notion of “neighbors of neighbors” to include points that are merely within the neighborhood of density-reachable points. Figure 1.20 illustrates how the points located at $(4, 7)$ and $(0, -2)$ are density-connected but not density-reachable.

A *cluster* $C \subset D$ with respect to ε and $Minpts$ satisfies the following conditions:

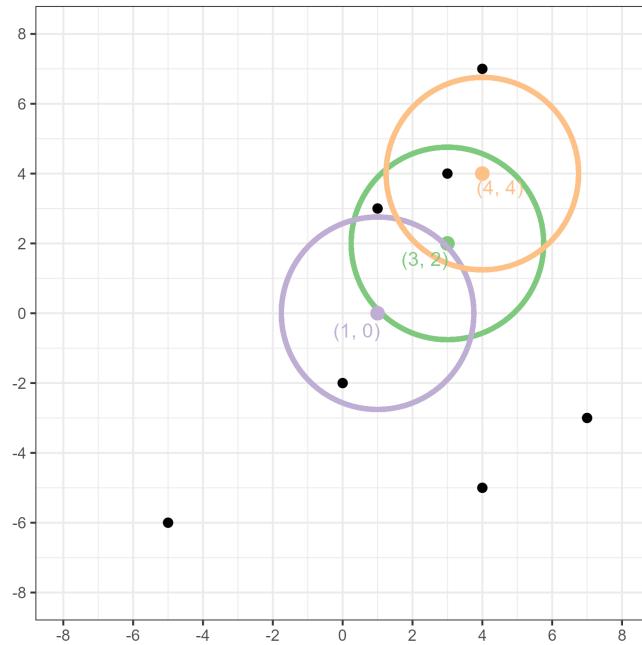


Figure 1.19: An example of three points that are density-reachable with respect to $\epsilon = 3$ and $Minpts = 2$.

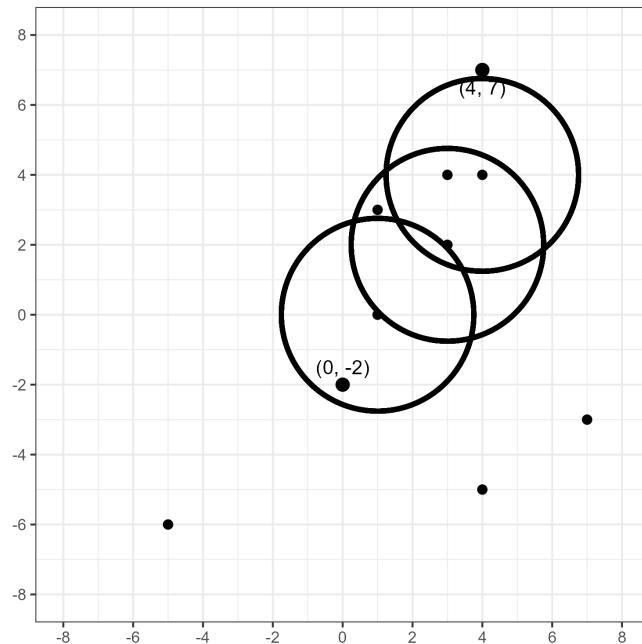


Figure 1.20: An example of two points that are density-connected, but not density-reachable, with respect to $\epsilon = 3$ and $Minpts = 2$.

1. $\forall x, y$: if $x \in C$ and y is density-reachable from x with respect to ϵ and $Minpts$, then $y \in C$.
2. $\forall x, y \in C$: x is density-connected to y with respect to ϵ and $Minpts$.

For a data set D , the DBSCAN algorithm determines clusters based on the above definition. Points not assigned to a cluster are classified as *noise points*. The algorithm halts once all points are assigned to a cluster or classified as noise.

Figure 1.21 shows the labels return by DBSCAN for the example considered above with respect to $\epsilon = 3$ and $Minpts = 2$. The algorithm finds a cluster of seven points, colored blue, and classifies three points as noise, colored red.

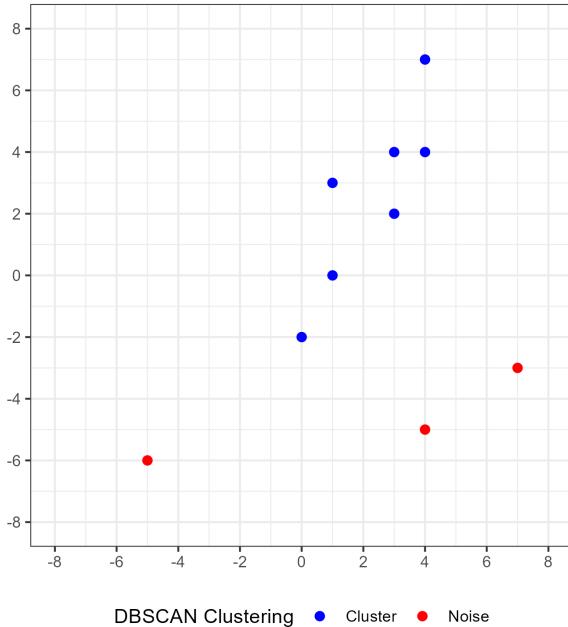


Figure 1.21: Cluster labeling for 10 data points using the DBSCAN algorithm with parameters $\epsilon = 3$ and $Minpts = 2$. Seven points are assigned to a single cluster and the remaining three are classified as noise.

1.4.3 Implementation Considerations

In the computational sciences, it is one thing to publish code along with research findings. Publicly-available code and data make results accessible in terms of acquisition. It is much more challenging to make code *conceptually* accessible to others. The former allows others to obtain the same results under the same programming conditions while the latter empowers others to actually engage with and potentially improve upon individual pieces of the algorithm. In any data analysis pipeline, the procedural details may be obscured as the goals of the analysis become more sophisticated. It is therefore worthwhile to design tools that make the data analysis process both easier to implement and understand (Wickham, 2014).

Our implementation of the cartridge case comparison pipeline adheres to the “tidy” principles of design (Wickham et al., 2019). The “tidyverse” is a collection of R packages that share an underlying design philosophy and structure. Knowledge and skills learned while using one tidy package can be applied to others. The four principles of a tidy API are:

1. *Reuse existing data structures.*

For example, users do not need to learn new data attributes or compatible functions if a package reuses existing data structures.

2. *Compose simple functions with the pipe.*

The pipe operator allows the output of one function to be passed as input to another without assigning a new variable. We incrementally transform data as they move from one function to another rather than drastically transforming the data in a single call.

3. Embrace functional programming.

The functional programming paradigm encourages immutability of objects, meaning data passed as input to a function are not changed. Rather, the function makes a copy of the input data, manipulates the copy, and returns the transformed copy as output. This differs from an “object-oriented” paradigm where functions have the ability to implicitly rewrite or change the state of the original data. It is easier to reason about a function that actually returns an object as output than one that changes the input object as a “side effect.”

4. Design for humans.

Designing a package for humans largely comes down to using consistent, explicit, and descriptive naming schemes for objects and functions.

Conceptualizing the cartridge case comparison procedure as a pipeline makes it easier to understand. We go one step further by actually implementing the procedure as a sequence of algorithms that are programatically connected together in the R statistical programming language (R Core Team, 2017). In particular, we utilize the pipe operator `%>%` available from the magrittr R package (Bache and Wickham, 2022). The pipe operator allows the user to think intuitively in terms of verbs applied to the data. Table 1.2 illustrates two pipelines that utilize the pipe operator. The left-hand example shows how an R data frame is manipulated by piping it between functions from the dplyr package. Functions like `group_by`, `summarize`, and `filter` are simple building blocks strung together to create complicated workflows. The right-hand example similarly illustrates a cartridge case object passing through a comparison pipeline. While the full comparison procedure is complex, the modularization to the `preProcess_`, `comparison_`, and `decision_` steps, which can further be broken-down into simpler functions, renders the process more understandable and flexible for the user.

Data Frame Manipulation Example	Cartridge Case Comparison Example
<pre>dataFrame %>% group_by(category) %>% summarize(x = summary(var)) %>% filter(x > 0) ...</pre>	<pre>cartridgeCase1 %>% preProcess_func(params1) %>% comparison_func(cartridgeCase2, params2) %>% decision_func(params3) ...</pre>

Table 1.2: Two examples of data analysis workflows that utilize the pipe operator. The left side shows a data frame manipulation while the right side shows a comparison of two cartridge cases.

Adherence to tidy principles makes it easier to engage with and understand the overall data analysis pipeline. In our application it also enables experimentation by making it easy to change one step of the pipeline and measure the downstream effects (Zimmerman et al., 2019). Each step of the cartridge case comparison pipeline requires the user to define parameters. These can range from minor to substantial, such as choosing the standard deviation used in a Gaussian filter to choosing the algorithm used to calculate a similarity score. So far, no consensus exists for the “best” parameter settings. A large amount of experimentation is yet required to establish these parameters. A tidy implementation of the cartridge case comparison pipeline allows more people to engage in the validation and improvement of the procedure.

Figure 1.22, Figure 1.23, Figure 1.24, and Figure 1.25 illustrate how various forensic comparison algorithms use a modularized structure to conceptualize their pre-processing procedures. In each figure, a sequence of modular procedures are applied to a piece of evidence. Figure 1.22 shows morphological and image processing procedures applied to a 2D image of a cartridge case to remove the firing pin region (Tai and Eddy, 2018).

Figure 1.23 shows the procedure by which a 2D “signature” of a bullet scan is extracted from a 3D topographical scan (Rice, 2020).

Figure 1.24 shows how an image of the written word “csafe” is processed using the handwritten R package to break the word into individual *graphemes* that can be further processed (Berry et al., 2021).

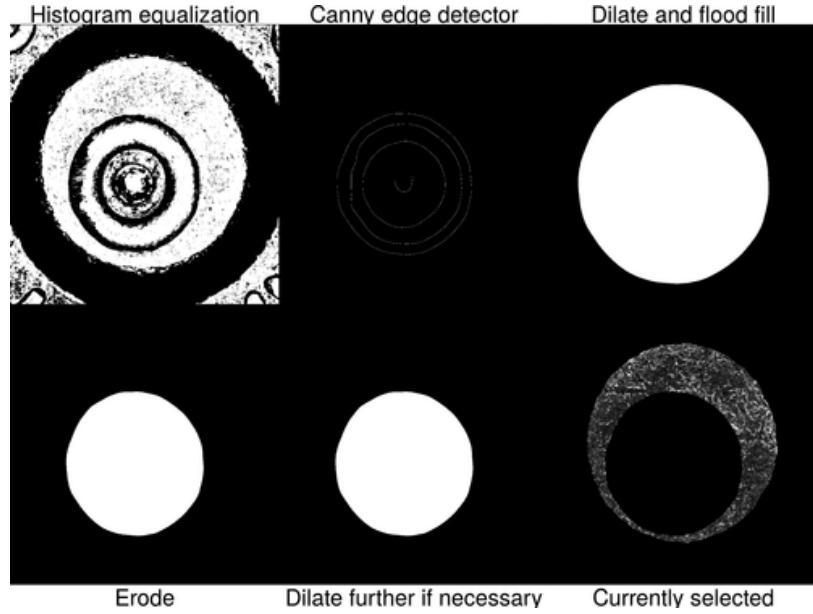


Figure 1.22: A pre-processing procedure applied to a 2D image of a cartridge case to identify the firing pin impression. The procedure results in a 2D image of a cartridge case without the firing pin impression region.

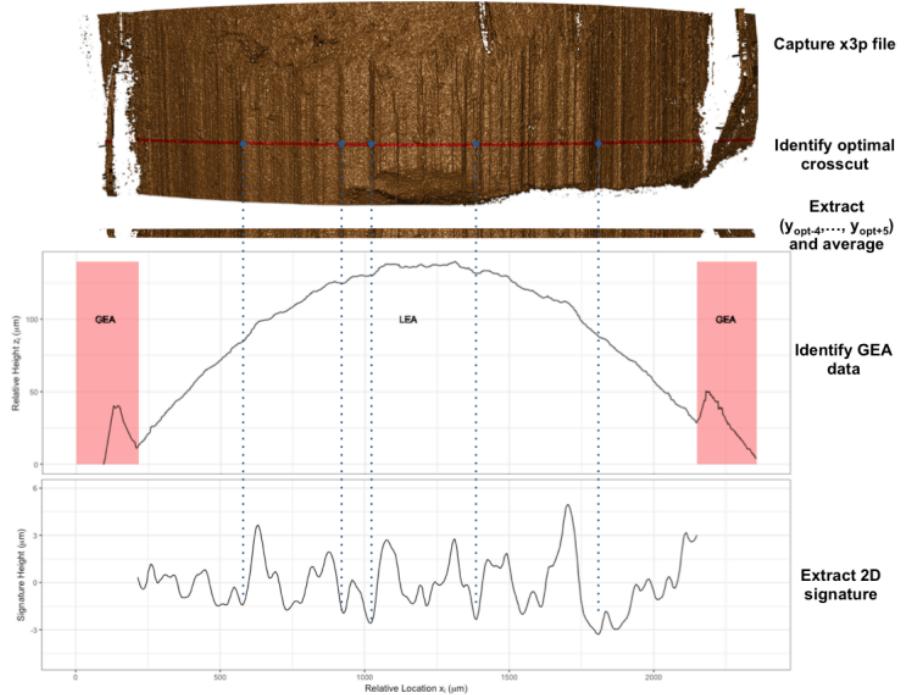


Figure 1.23: A pre-processing procedure for extracting 2D bullet "signatures" from a 3D topographic bullet scan. The procedure results in an ordered sequence of values representing the local variations in the surface of the bullet.



Figure 1.24: A pre-processing procedure applied to an image of the handwritten word "csafe." The procedure results in a skeletonized version of the word that has been separated into graphemes as represented by orange nodes.

Finally, Figure 1.25 shows a 3D topographical cartridge case scan undergoing various procedures to isolate and highlight the breech face impressions. These procedures are discussed in greater detail in Chapter 2.

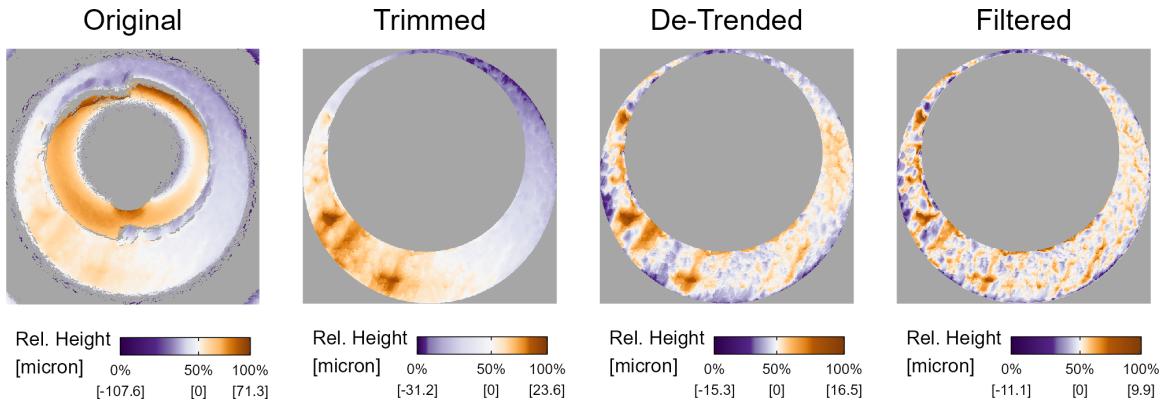


Figure 1.25: A cartridge case undergoing various pre-processing steps. The procedure results in a cartridge case scan in which the breech face impressions have been segmented and highlighted.

By breaking the broader pre-processing step into modularized pieces, we can devise other arrangements of these pre-processing procedures that may improve the segmenting or emphasizing of the region of interest. The modularity of the pipeline makes it easier to

understand what the algorithm is doing “under the hood.” A genuine modular implementation enables others to experiment with alternative versions of the pipeline, thus accelerating discovery and improvement.

CHAPTER 2. A Study in Reproducibility: The Congruent Matching Cells Algorithm and cmcR package

Abstract

Scientific research is driven by our ability to use methods, procedures, and materials from previous studies and further research by adding to it. As the need for computationally-intensive methods to analyze large amounts of data grows, the criteria needed to achieve reproducibility, specifically computational reproducibility, have become more sophisticated. In general, prosaic descriptions of algorithms are not detailed or precise enough to ensure complete reproducibility of a method. Results may be sensitive to conditions not commonly specified in written-word descriptions such as implicit parameter settings or the programming language used. To achieve true computational reproducibility, it is necessary to provide all intermediate data and code used to produce published results. In this paper, we consider a class of algorithms developed to perform firearm evidence identification on cartridge case evidence known as the *Congruent Matching Cells* (CMC) methods. To date, these algorithms have been published as textual descriptions only. We introduce the first open-source implementation of the Congruent Matching Cells methods in the R package cmcR. We have structured the cmcR package as a set of sequential, modularized functions intended to ease the process of parameter experimentation. We use cmcR and a novel variance ratio statistic to explore the CMC methodology and demonstrate how to fill in the gaps when provided with computationally ambiguous descriptions of algorithms.

2.1 Introduction

Forensic examinations are intended to provide an objective assessment of the probative value of a piece of evidence. Typically, this assessment of probative value is performed by a forensic examiner who visually inspects the evidence to determine whether it matches evidence found on a suspect. The process by which an examiner arrives at their evidentiary conclusion is largely opaque and has been criticized (President’s Council of Advisors on Sci. & Tech., 2016) because its subjectivity does not allow for an estimation of error rates. In response, National Research Council (2009) pushed to augment subjective decisions made by forensic examiners with automatic algorithms that objectively assess evidence and can be explained during court testimony. In addition to the objectivity of these algorithms, there is an additional benefit: we expect that an algorithm with the same random seed run on the same data multiple times will produce the same answer; that is, that the results are repeatable. This is extremely beneficial because it allows the prosecution and defense to come to the same conclusion given objective evidence or data.

2.1.1 Repeatability and reproducibility

Repeatability in forensic labs is enforced primarily using standard operating procedures (SOPs), which specify the steps taken for any given evaluation, along with the concentrations of any chemicals used, the range of acceptable machine settings, and any calibration procedures required to be completed before the evidence is evaluated. When labs use computational procedures, this SOP is augmented with specific algorithms, which are themselves SOPs intended for use by man and machine. Algorithms are generally described on two levels: we need both the conceptual description (intended for the human using the algorithm) and the procedural definition (which provides the computer hardware with a precise set of instructions). For scientific and forensic repeatability and reproducibility, it is essential to have both pieces: the algorithm description is critical for establishing human understanding

and justifying the method’s use in court, but no less important is the computer code which provides the higher degree of precision necessary to ensure the results obtained are similar no matter who evaluates the evidence. As with SOPs in lab settings, the code parameters function like specific chemical concentrations; without those details, the SOP would be incomplete and the results produced would be too variable to be accepted in court.

The National Academy of Sciences, Engineering, and Medicine (2019) defines *reproducibility* as “obtaining consistent computational results using the same input data, computational steps, methods, code, and conditions of analysis.” This form of reproducibility requires that the input data, code, method, and computational environment are all described and made available to the community. In many situations, this level of reproducibility is not provided – not just in forensics but in many other applied disciplines. In forensics in particular, it is easier to list the exceptions: reproducible algorithms have been proposed in sub-disciplines including DNA (Tvedebrink et al., 2020; Goor et al., 2020; Tyner et al., 2019), glass (Curran et al., 2000b; Park and Tyner, 2019), handwriting (Crawford, 2020), shoe prints (Park and Carriquiry, 2020), and ballistic evidence (Hare et al., 2017; Tai and Eddy, 2018).

We find it useful to instead consider a more inclusive hierarchy of reproducibility. Algorithms at higher tiers of the hierarchy are more easily reproducible in the sense that fewer resources are required to (re)-implement the algorithm.

Definition 1 *Hierarchy of Reproducibility*

Conceptual description *The algorithm is described and demonstrated in a scientific publication.*

Pseudocode *The algorithm is described at a high level of detail with pseudocode implementation provided, and results are demonstrated in a scientific publication.*

Reproducible data *The algorithm is described and demonstrated in a scientific publication, and input data are available in supplementary material.*

Comparable results *The algorithm is described and demonstrated in a scientific publication, and input data and numerical results are provided in supplementary material.*

Full reproducibility *The algorithm is described and demonstrated in a scientific publication, and the input data, source code, parameter settings, and numerical results are provided in supplementary material.*

To aid in comprehension of an algorithm, it is useful to supplement conceptual descriptions with pseudocode. However, a conceptual description and pseudocode alone do not contain sufficient detail (e.g., parameter settings) to ensure computational reproducibility. Implementing algorithms based on conceptual descriptions or pseudocode requires enumerating and testing possible parameter choices which, depending on their complexity, can be a lengthy and expensive process. In contrast, implementing fully reproducible algorithms requires only as much time as it takes to emulate the original development environment. Commonly identified reasons for unreproducible results include (1) ambiguity in how procedures were implemented, (2) missing or incomplete data, and (3) missing or incomplete computer code to replicate all statistical analyses (Leek and Jager, 2017). In particular, for statistical algorithms which depend on input data, we find that full reproducibility depends on the provision of both original data and any manual pre-processing applied to said data, as this manual process is not reproducible by itself. In combination with the code, the algorithm description, and the numerical results presented in the paper, it should be possible to fully reproduce the results of a paper.

In this paper, we demonstrate the importance of higher levels of reproducibility by examining the Congruent Matching Cells (CMC) algorithm for cartridge case comparisons

and developing an open-source, fully reproducible version for general use in the forensics community.

2.1.2 The Congruent Matching Cells algorithm

A *cartridge case* is the portion of firearm ammunition that encases a projectile (e.g., bullet, shots, or slug) along with the explosive used to propel the projectile through the firearm. When a firearm is discharged, the projectile is propelled down the barrel of the firearm, while the cartridge case is forced towards the back of the barrel. It strikes the back wall, known as the *breech face*, of the barrel with considerable force, thereby imprinting any markings on the breech face onto the cartridge case and creating the so-called *breech face impressions*. These markings are used in forensic examinations to determine whether two cartridge cases have been fired by the same firearm. During a forensic examination, two pieces of ballistic evidence are placed under a *comparison microscope*. Comparison microscopes allow for a side-by-side comparison of two objects within the same viewfinder, as seen in Figure 2.1. A pair of breech face images is aligned along the thin black line in the middle of the images. The degree to which these breech face markings can be aligned is used to determine whether the two cartridge cases came from the same source; i.e., were fired from the same firearm. These breech face impressions are considered analogous to a firearm’s “fingerprint” left on a cartridge case (Thompson, 2017).

The Congruent Matching Cells (CMC) pipeline is a collection of algorithms to process and compare cartridge case evidence (Song, 2013). Since its introduction, the pipeline and its extensions (Tong et al., 2015; Chen et al., 2017; Song et al., 2018) have shown promise in being able to differentiate between matching and non-matching cartridge cases. However, so far the CMC pipelines have only been introduced in the form of conceptual descriptions. Further, the cartridge case scans used to validate the pipelines are only available in their raw, unprocessed forms on the NIST Ballistics Toolmark Research Database (Zheng et al., 2016). While it is clear that the creators of the CMC pipeline have a working implementation, the

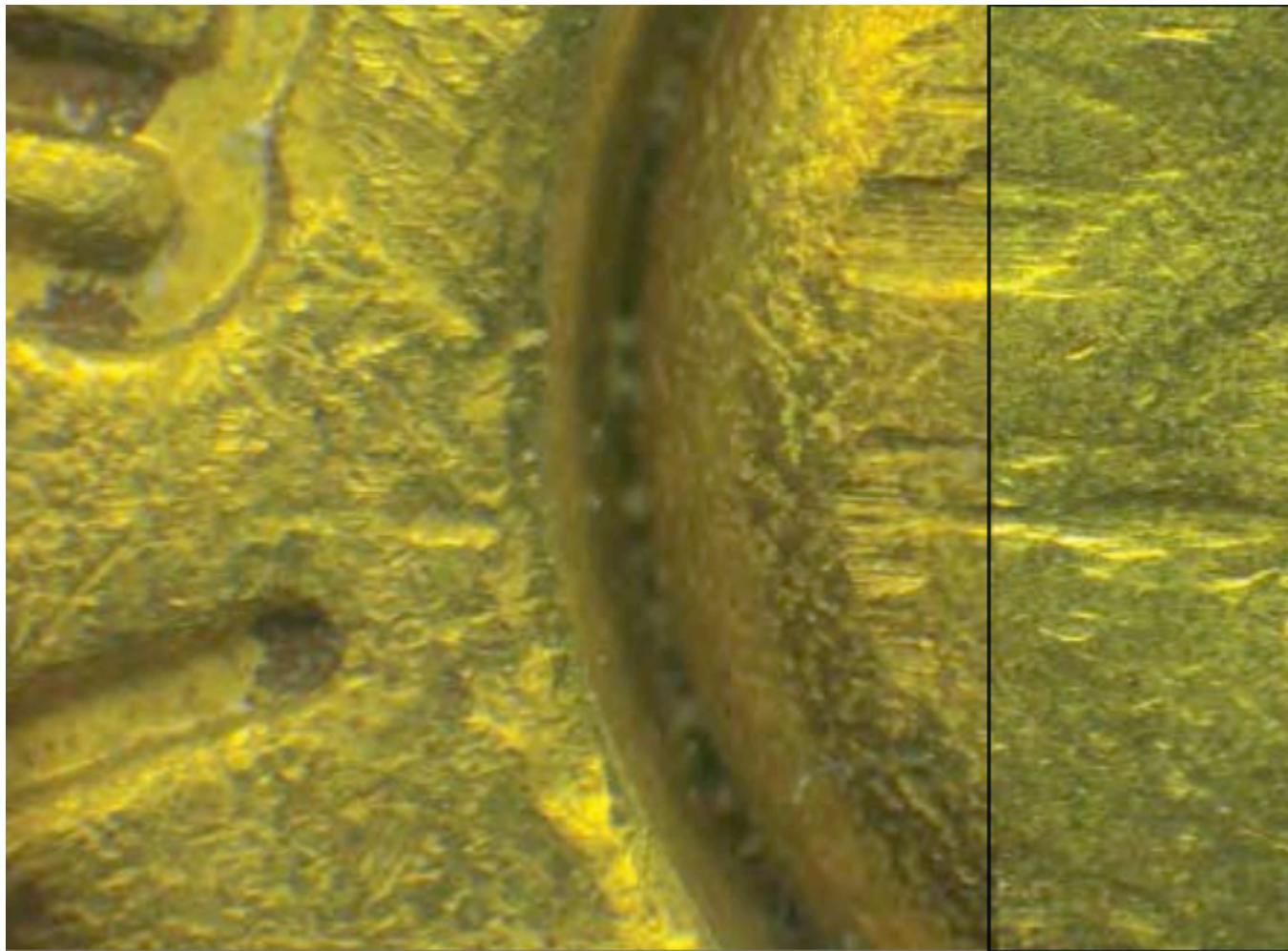


Figure 2.1: A cartridge case pair with visible breech face impressions under a microscope. A thin line can be seen separating the two views. The degree to which the markings coincide is used to conclude whether the pair comes from the same source.

wider forensic science community only has access to conceptual descriptions of the pipeline and summary statistics describing its performance. In our hierarchy of reproducibility, this puts the CMC algorithm somewhere between the conceptual description and reproducible data stage: the steps are described but no code is available, and the raw data are available but manual pre-processing steps make this raw data insufficient to replicate the pipeline even with newly written code.

The development of the CMC algorithm seems to be representative of how many forensic algorithms are developed: after an algorithm is introduced, researchers build upon the foundation laid by the original algorithm in subsequent papers. These changes are often incremental in nature and reflect a growing understanding of the algorithm’s behavior. While this cycle of scientific progress certainly is not unique to forensic algorithms, given the gravity of the application it is imperative that these incremental improvements not be unnecessarily delayed. As such, we believe that the forensic community at-large would benefit greatly by establishing an open-source foundation for their algorithms upon which additional improvements can be developed. Using open-source algorithms are cheaper to use than writing one’s own code, enables the process of peer review by providing an accessible benchmark, and helps other research groups or companies stay on the leading edge of technology development (The Linux Foundation, 2017).

Here, we describe the process of implementing the CMC pipeline for the comparison of marks on spent cartridge cases, using the descriptions from two published papers, Song et al. (2014) and Tong et al. (2015). Our R package, `cmcR`, provides an open-source implementation of the CMC pipeline. We use `cmcR` to illustrate how ambiguities in the textual description of an algorithm can lead to highly divergent results. In particular, our implementation highlights an extreme sensitivity to processing and parameter decisions that has not been discussed previously. Additionally, we argue that our implementation can be used as a template for future implementations of forensic pattern-matching algorithms to not

only ensure transparency and auditability, but also to facilitate incremental improvements in forensic algorithms.

In the remainder of this paper, we describe a general, reproducible, and open-source CMC pipeline which encompasses those discussed in Song (2013), Song et al. (2014), and Tong et al. (2015). Song (2013) lays out the conceptual framework for the original CMC pipeline later implemented in Song et al. (2014) and Tong et al. (2014). An improvement of the pipeline presented in Tong et al. (2015) and used in subsequent papers is referred to as the “High CMC” method (Chen et al., 2017). However, it should be noted that what the authors refer to as the original and High CMC decision rules are variations of one step of a larger CMC pipeline.

The cmcR package contains implementations designed for use with 3D topographical scans of the original decision rule described in Song (2013) and Song et al. (2014) and the High CMC decision rule described in Tong et al. (2015). The source code to the full cmcR package is accessible at <https://github.com/CSAFE-ISU/cmcR>.

2.2 The CMC pipeline

In this section, we examine the process of implementing the CMC pipeline for automatic comparisons of 3D cartridge case scans. At each step, we will discuss how we filled in the gaps of the original description during the creation of cmcR.

All of the CMC pipelines can be broken down into three broad stages: (1) pre-processing, (2) cell-based similarity feature extraction, and (3) application of a decision rule as illustrated in 2.2. In the following sections we break each of these stages further into a set of modular steps. One advantage of modularizing these algorithms is that we can implement an algorithm as a set of sequential procedures. This allows us to test new variations against the old implementation in a coherent, unified framework.

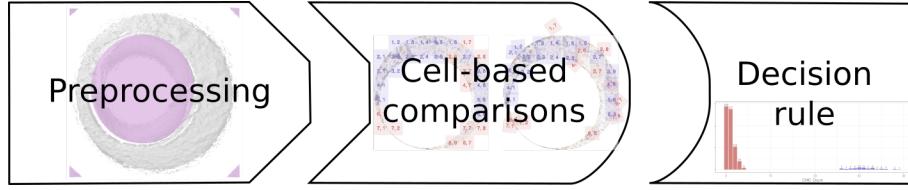


Figure 2.2: The stages of CMC pipelines. In the pre-processing stage, each scan is prepared for analysis, removing extraneous information and noise. Then, each scan is broken up into cells, which are numerically compared to cells in the other scan to determine an optimal alignment. Finally, each of the scores arising from the cells in the second stage are compared to a reference distribution to determine whether the scans originate from the same source or from different sources.

The primary difference between the two pipelines presented here, using the original and High CMC decision rules, lies in how the decision rules are utilized to separate matching vs. non-matching cartridge case pairs. In addition, there are also several small differences in the parameters used in the pre-processing and comparison procedures.

2.2.1 Initial data

Digital microscopy is capable of precision measurements of surface topology at high resolutions. Using a 3D microscope, we can obtain scans of breech face impressions at the micron level ($1\mu m = 10^{-3}mm = 10^{-6}m$). These 3D topological scans are used as input to automated comparison algorithms, such as the CMC pipeline originally proposed in Song (2013). We will use the same data set referenced in Song et al. (2014) and Tong et al. (2015) to illustrate usage of the cmcR package. These 3D scans of cartridge cases are available from the NIST Ballistics Toolmark Research Database (Zheng et al., 2016). The strings defined below refer to three cartridge case scans available on the NBTRD from Fadul et al. (2011b) and will be used throughout the remainder of this paper.

```
library(cmcR)

nbtrd_url <- "https://tsapps.nist.gov/NBTRD/Studies/CartridgeMeasurement"
```

```

x3p_ids <- c("DownloadMeasurement/2d9cc51f-6f66-40a0-973a-a9292dbe36d",
            "DownloadMeasurement/cb296c98-39f5-46eb-abff-320a2f5568e8",
            "DownloadMeasurement/8ae0b86d-210a-41fd-ad75-8212f9522f96")

file_names <- c("fadul1-1.x3p","fadul1-2.x3p","fadul2-1.x3p")

purrr::walk2(.x = x3p_ids,
             .y = file_names,
             .f = function(x3p_id,file_name){
               download.file(url = file.path(nbtrd_url, x3p_id),
                             destfile = paste0("data/",file_name),mode = "wb")
             })

```

Cartridge case scans are commonly stored in the ISO standard x3p file format (ISO 25178-72(2017), 2017). x3p is a container format which consists of a single surface matrix representing the height value of the breech face surface and metadata concerning the parameters under which the scan was taken (size, resolution, creator, microscope, microscopy software versions, etc.). The x3ptools package provides functionality to work with the format in R (Hofmann et al., 2020).

Figure 2.3 shows the surface matrices of a known match (KM) pair of cartridge cases from a study by Fadul et al. (2011b). In this study, a total of 40 cartridge cases were scanned with a lateral resolution of 6.25 microns (micrometers) per pixel. The surface matrices are approximately 1200×1200 pixels in size corresponding to an area of about $3.8 \times 3.8 \text{ mm}^2$.

Only certain regions of a cartridge case contain identifying breech face impression markings. Song (2013) defines “valid correlation regions” as regions where “the individual characteristics of the ballistics signature are found that can be used effectively for ballistics

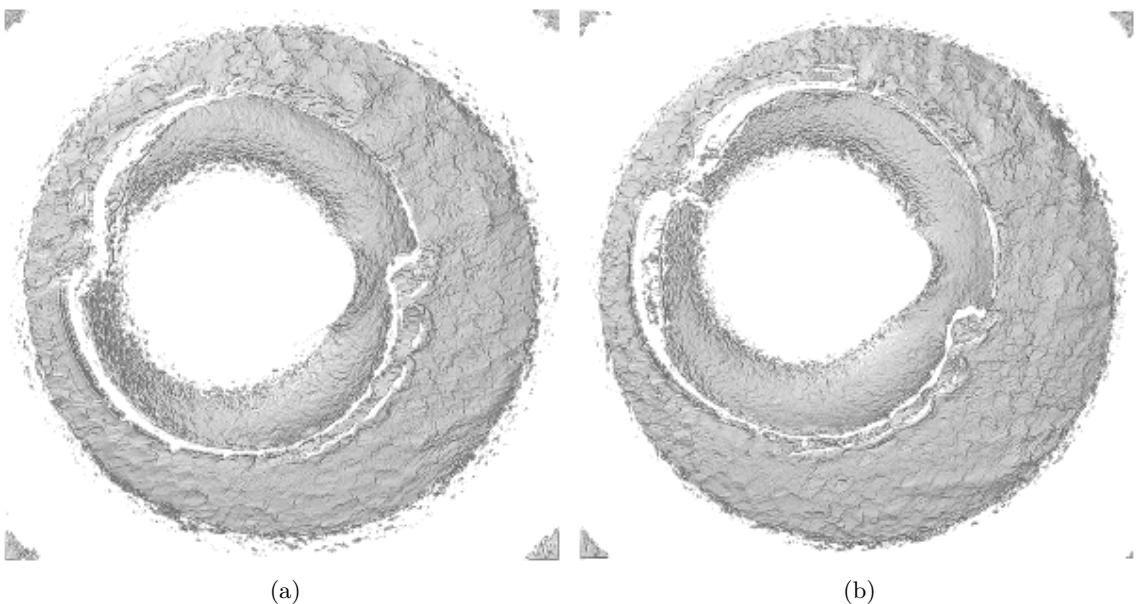


Figure 2.3: Unprocessed surface matrices of the known-match Fadul 1-1 and Fadul 1-2 Fadul et al. 2011. The observations in the corners of these surface matrices are artifacts of the staging area in which these scans were taken. The holes on the interior of the primer surfaces are caused by the firing pin striking the primer during the firing process. The region of the primer around this hole does not come into uniform contact with the breech face of the firearm.

identification.” Prior to applying the CMC comparison procedure, cartridge scans must undergo some pre-processing to isolate the valid correlation regions.

2.2.2 Pre-processing procedures

During the pre-processing stage, we apply sequential steps to prepare each cartridge case for analysis. The goal of this process is to remove the edges and center of the scan which did not come into contact with the breech face, as well as any artifacts of the scan and microscope staging which do not accurately represent the breech face surface. The various iterations of the CMC algorithm describe different variations of these steps. A summary of these steps is shown in Figure 2.4.

Translating the pre-processing steps in Figure 2.4 into an implementation requires the implementer to decide between potentially many implicit parameter choices. For example, Table 2.1 compares the pre-processing procedures as described in Song et al. (2014) to considerations that need to be made when implementing the procedures. Depending on one’s interpretation of the description, there are many possible implementations that satisfy the described procedure - in contrast, there was only one implementation that led to the original results. While not explicitly mentioned in Song et al. (2014), Song et al. (2018) indicates that the “trimming” of the unwanted regions of the scan is performed manually. It is difficult to replicate manual steps as part of a reproducible pipeline; the best solution is for the authors to provide intermediate data after the manual steps have been completed.

The pre-processing procedures are implemented via modularized functions of the form `preProcess_*`. Modularizing the steps of the pre-processing procedures makes the overall process easier to understand and allows for experimentation. Figure 2.5 shows an overview of the pre-processing framework for the Fadul 1-1 breech face from reading the scan (left) to an analysis-ready region (right). For each scan in Figure 2.5, eleven height value percentiles: the Minimum (0th), 1st, 2.5th, 10th, 25th, Median (50th), 75th, 90th, 97.5th, 99th, and

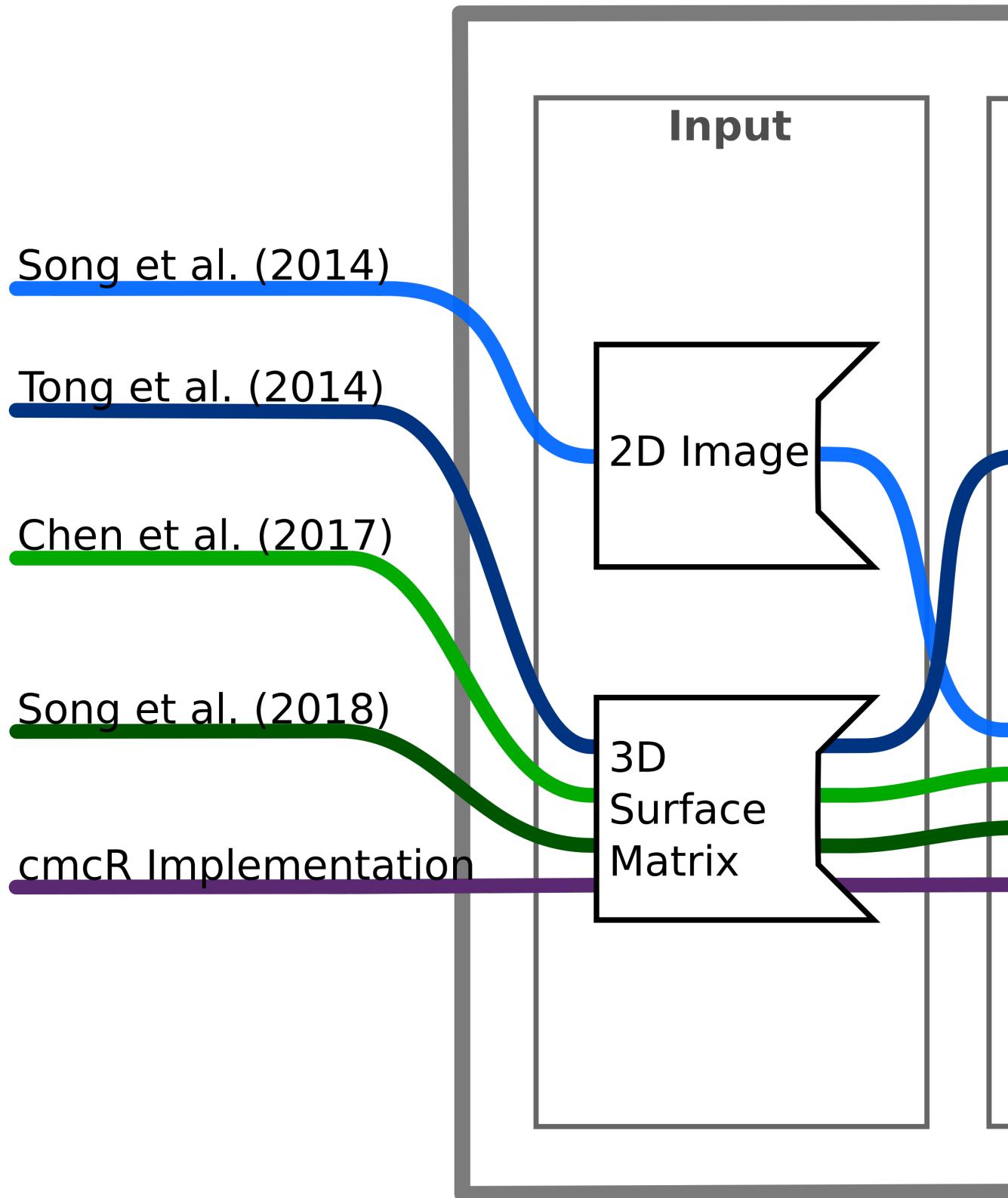


Figure 2.4: Overview of the set of pre-processing steps used in the CMC algorithms. Where a procedure step is not discussed or explicitly not applied in the paper, the path traverses empty space.

Table 2.1: Description of pre-processing procedures from Song et al. 2014 vs. considerations that need to be made when implementing these procedures. Each of these considerations requires the implementer to decide between potentially many choices.

Description from Song et al. (2014)	Implementation Considerations
”Trim off the inside firing pin surface and other areas outside the breech face mark, so that only breech face impression data remain for correlation.”	Removal of firing pin hole, primer exterior, global trend, and primer roll-off
”Identify and remove dropouts or outliers.”	Definition of outliers, what ”removal” of dropouts or outliers means
”Apply a band-pass Gaussian regression filter with 40 μm short cut-off length and 400 μm long cut-off length to remove low frequency components, including surface curvature, form error, waviness and high frequency components which mainly arise from the instrument noise.”	Wavelength cut-off parameters, specific implementation of the filter

Maximum (100th) are mapped to a purple-to-orange color gradient. This mapping is chosen to highlight the extreme values in each scan.

We demonstrate usage of the `preProcess_*` functions on the Fadul 1-1 scan. Each code chunk is followed up with an explanation of the functions used.

```
# Step (1)
fadul1.1 <- x3ptools::x3p_read("data/fadul1-1.x3p")
```

We begin with a 3D scan. Typically, we downsample scans to about 25% of their size by only retaining every other row and column in the surface matrix. The breech faces in Fadul et al. (2011b) were initially scanned at a resolution of 3.125 μm per pixel. Downsampling reduces the resolution to 6.25 μm per pixel. Step (1) in Figure 2.5 shows an unprocessed breech face scan.

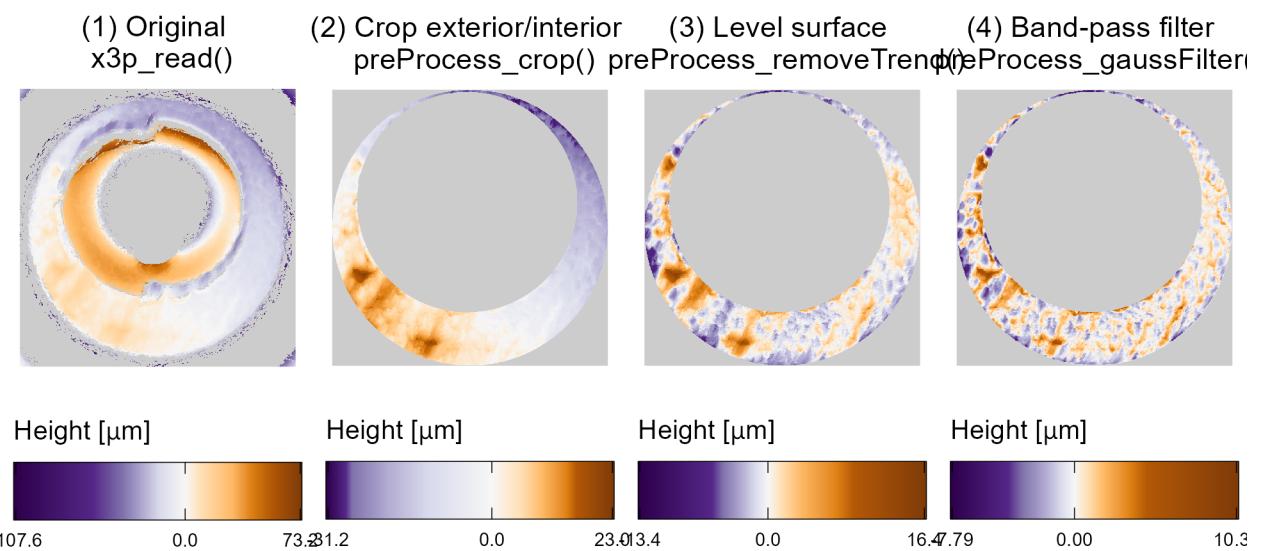


Figure 2.5: Illustration of the sequential application of pre-processing steps implemented in cmcR. We map the cartridge case surface height values to a divergent purple-white-orange color scale to emphasize deviations from the median height value (represented here as 0 micrometers). At each stage, the variability in height across the scan decreases as we emphasize the regions containing breech face impressions.

```
# Step (2)

fadul1.1_cropped <- fadul1.1 %>%
  cmcR::preProcess_crop(region = "exterior") %>%
  cmcR::preProcess_crop(region = "interior")
```

We then use a labeling algorithm to identify three major regions of the scan: the exterior of the cartridge case primer, the breech face impression region of interest, and the firing pin impression region in the center of the scan (Hesselink et al., 2001; Barthelme, 2019). We remove observations outside of the breech face impression region (i.e., replaced with NA). The resulting breech face scan, like the one shown in step (2) of Figure 2.5, is reproducible assuming the same parameters are used. The `preProcess_crop` function removes the exterior and firing pin impression region on the interior based on the `region` argument.

```
# Step (3)

fadul1.1_deTrended <- fadul1.1_cropped %>%
  preprocess_removeTrend(statistic = "quantile", tau = .5, method = "fn")
```

In step (3), we remove the southwest-to-northeast trend observable in steps (1) and (2) of Figure 2.5 by subtracting the estimated conditional median height value. The result of the `preProcess_removeTrend` function the median-leveled breech face scan in step (3) of Figure 2.5.

```
# Step (4)

fadul1.1_processed <- fadul1.1_deTrended %>%
  preprocess_gaussFilter(filtertype = "bp", wavelength = c(16,500)) %>%
  x3ptools::x3p_sample(m = 2)
```

Finally, we apply a band-pass Gaussian filter to the surface values to attenuate noise and unwanted large-scale structure. Step (4) of Figure 2.5 shows the effect of the `preProcess_gaussFilter` function. There is currently no determination or removal of outliers in the cmcR package's pre-processing procedures. Instead, we rely on the low-pass portion of the Gaussian filter to reduce the effects of any high-frequency noise.

Figure 2.6 displays the processed Fadul 1-1 and Fadul 1-2 scans; the second matrix is processed using the same parameters. Next, similarity features are extracted from a processed cartridge case pair in the cell-based comparison procedure.

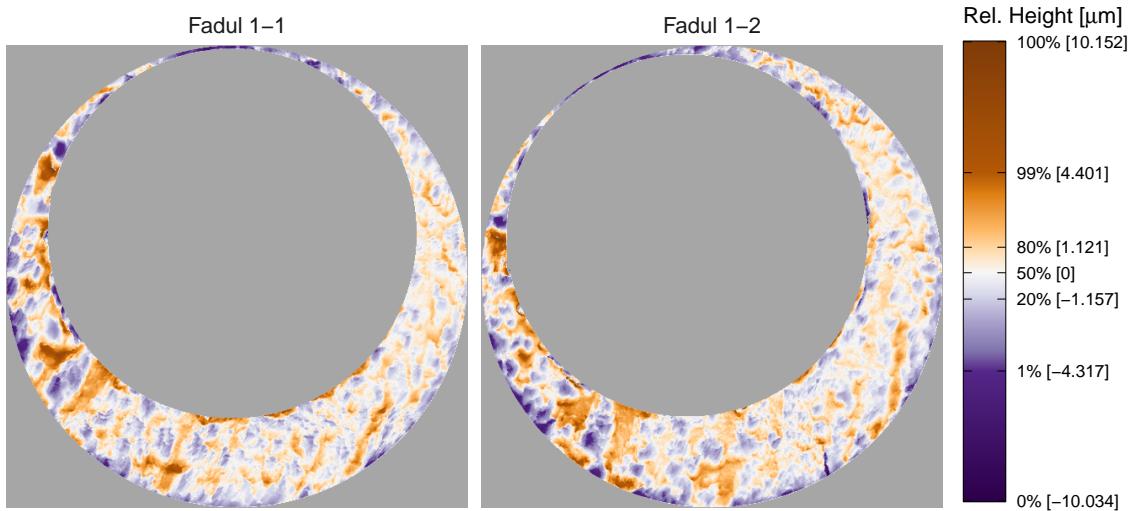


Figure 2.6: Fadul 1-1 and Fadul 1-2 after pre-processing. Similar striated markings are now easier to visually identify on both surfaces. It is now clearer that one of the scans needs to be rotated to align better with the other.

2.2.3 “Correlation cell” comparison procedure

As described in Song (2013), breech face markings are not uniformly impressed upon a cartridge case during the firing process. As such, only certain sections of the cartridge case are used in a comparison. In the CMC pipeline as proposed by Song (2013) two scans are compared by partitioning one breech face scan into a grid of so-called “correlation cells”. These cells are compared individually to their best-matching counterpart on the other scan. If a large proportion of these correlation cells are highly similar to their counterparts on the other breech face scan, this is considered as evidence that the markings on the two cartridge cases were made by the same source. The number of highly similar cells is defined as the *CMC count C* (Song, 2013) of the breech-face comparison. The CMC count is considered to be a more robust measure of similarity than the correlation calculated between two full scans.

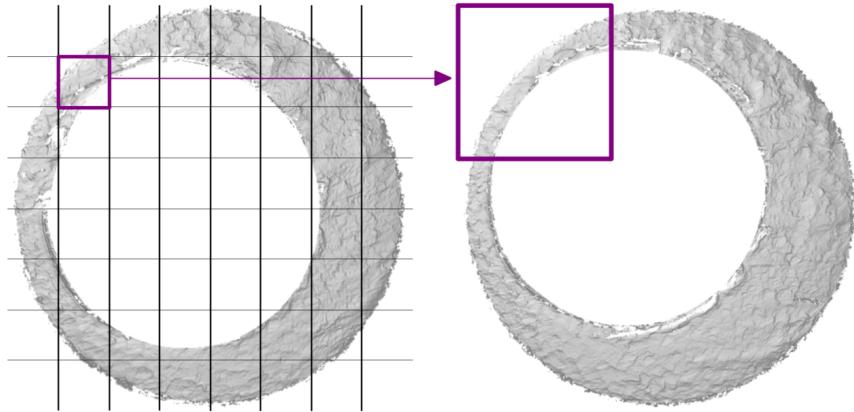


Figure 2.7: Illustration of comparing a cell in the reference cartridge case scan (left) to a larger region in a questioned cartridge case scan (right). Every one of the cells in the reference cartridge case is similarly paired with a region in the questioned cartridge case. To determine the rotation at which the two cartridge cases align, the cell-region pairs are compared for various rotations of the questioned cartridge case.

Figure 2.7 illustrates the cell-based comparison procedure between two cartridge case scans. The scan on the left serves as the reference; it is divided into a grid of 8×8 cells.

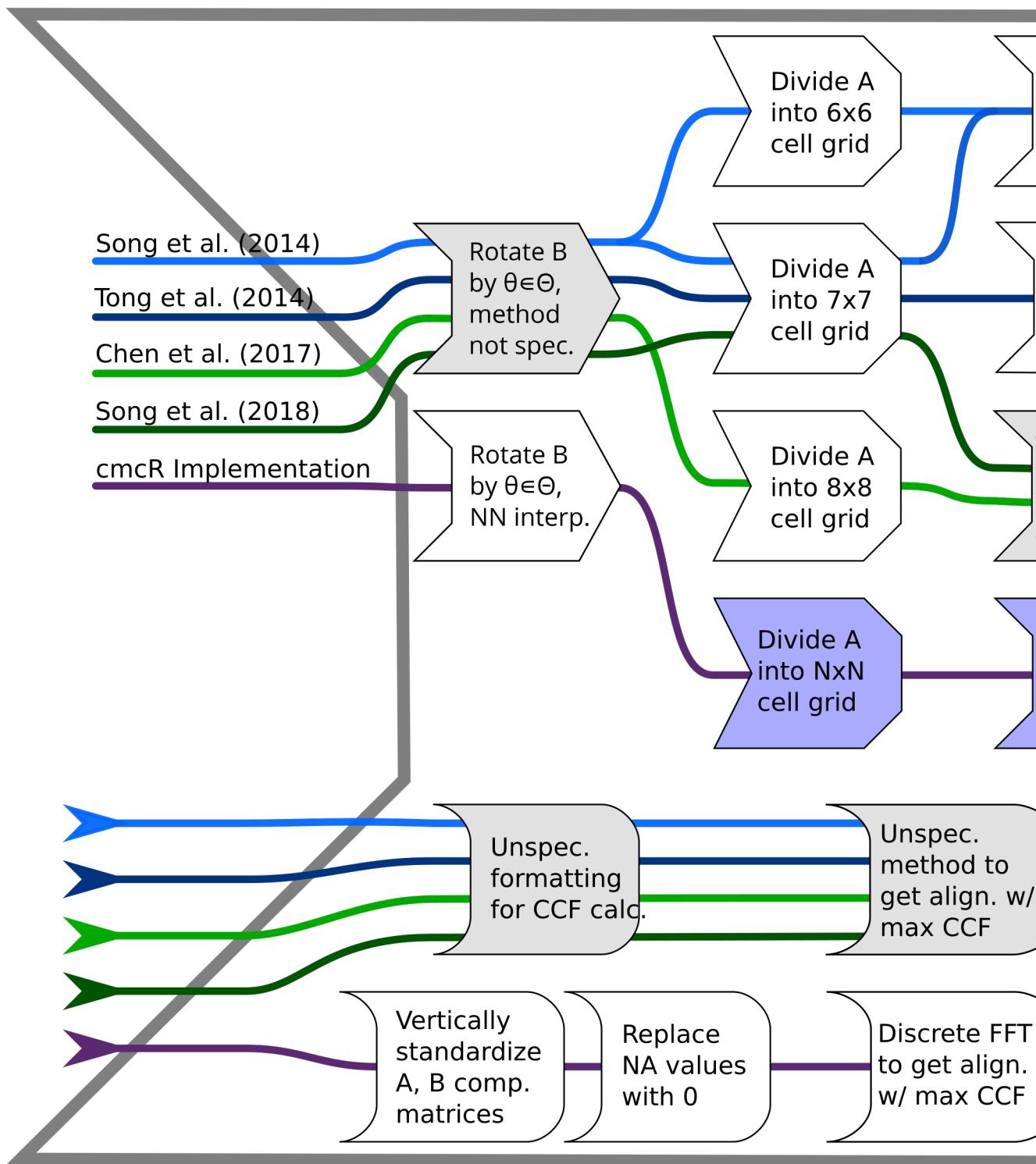


Figure 2.8: Each CMC implementation uses a slightly different procedure to obtain a similarity score between two cartridge cases. Steps which are implemented with additional user-specified parameters are shaded purple; steps which are described but without sufficient detail are shaded grey.

Figure 2.8 shows the steps of the correlation cell comparison process in each of the papers as well as the cmcR implementation. Each cell is paired with an associated larger region in the other scan. The absolute location of each cell and region in their respective surface matrices remain constant. However, the scan on the right is rotated to determine the rotation at which the two scans are the most “similar,” as quantified by the *cross-correlation function* (CCF).

For real-valued matrices A and B of dimension $M \times N$ and $P \times Q$, respectively, the cross-correlation function, denoted $(A \star B)$ is defined as

$$(A \star B)[m, n] = \sum_{i=1}^M \sum_{j=1}^N A[i, j]B[(i+m), (j+n)],$$

where $1 \leq m \leq M+P-1$ and $1 \leq n \leq N+Q-1$. By this definition, the $[m, n]$ th element of the resulting $M+P-1 \times N+Q-1$ CCF matrix quantifies the similarity between matrices A and B for a translation of matrix B by m pixels horizontally and n pixel vertically. The index at which the CCF attains a maximum represents the optimal translation needed to align B with A . The CCF as defined need not be bounded between -1 and 1 . However, it is common to normalize the CCF for interpretability, and this is the convention adopted in the cmcR package.

Prior to calculating the CCF, the matrices A and B are standardized through subtraction of their respective means and division by their respective standard deviations. This is referred to as the *Areal Cross-Correlation Function* (ACCF) in some CMC papers (Ott et al., 2017). A direct calculation of the CCF for breech face scans based on the definition above is prohibitively slow. While computationally feasible alternatives exist, Song (2013) and other CMC papers do not specify the algorithm used to calculate the CCF.

Published descriptions of the CMC algorithm do not detail how the CCF is calculated. In image processing, it is common to use an implementation based on the Fast Fourier Transform (Brown, 1992). This implementation leverages the Cross-Correlation Theorem, which states that for matrices A and B , the CCF can be expressed in terms of a frequency-

domain pointwise product:

$$(A \star B)[m, n] = \mathcal{F}^{-1} \left(\overline{\mathcal{F}(A)} \odot \mathcal{F}(B) \right) [m, n],$$

where \mathcal{F} and \mathcal{F}^{-1} denote the discrete Fourier and inverse discrete Fourier transforms, respectively, and $\overline{\mathcal{F}(A)}$ denotes the complex conjugate (Brigham, 1988). Because the product on the right-hand side is calculated pointwise, we trade the moving sum computations from the definition of the CCF for two forward Fourier transformations, a pointwise product, and an inverse Fourier transformation. The Fast Fourier Transform (FFT) algorithm can be used to reduce the computational load considerably. Our implementation of this FFT-based CCF calculation is adapted from the cartridges3D package (Tai, 2021).

No computational shortcut comes without some trade-offs, though, and this FFT-based CCF calculation is no different. The FFT does not tolerate missing values, and breech faces are not continuous surfaces – all of the white regions in Figure 2.7 correspond to missing values. While it is unclear how the CCF is implemented in the CMC papers, the cmcR package adopts the following conventions:

- Only cells with a minimum proportion of non-missing pixels are assessed. This minimum threshold differs across CMC papers (15% in Chen et al. (2017) vs. 10% in Song et al. (2018), as shown in Figure 2.8), and is referenced but not specified in several other papers (Tong et al., 2014; Song et al., 2014; Chu et al., 2013). The `comparison_calcPropMissing` function computes the proportion of a matrix that is missing (`NA`-valued).
- Missing values are replaced with the overall mean value when the FFT-based CCF is computed (using function `comparison_replaceMissing`).
- The optimal translation is determined using the FFT-based CCF (using `comparison_fft_ccf`).

- Based on the optimal translation determined from the FFT-based CCF, we compute the pairwise complete CCF directly, avoiding any distortion of the CCF computation based on compensation for missing values (using function `comparison_cor`).

All of the steps dealing with cell-based comparisons are implemented as functions of the form `comparison_*`. Similar to the `preProcess_*` functions, the `comparison_*` functions can be chained together through a sequence of pipes. Below, we use the `comparison_allTogether` function to perform the entire cell-based comparison procedure in one call. The comparison procedure is performed twice: once with Fadul 1-1 considered the “reference” scan divided into cells that are compared to the “target” scan Fadul 1-2 and again with the roles reversed.

```
# Fill in most of the arguments first
comp_w_pars <- purrr::partial(.f = comparison_allTogether,
                                numCells = c(8,8), maxMissingProp = .85)

# Then, map the remaining values to theta
kmComparisonFeatures <- purrr::map_dfr(
  seq(-30,30,by = 3),
  ~comp_w_pars(reference = fadul1.1, target = fadul1.2, theta = .))

kmComparisonFeatures_rev <- purrr::map_dfr(
  seq(-30,30,by = 3),
  ~comp_w_pars(reference = fadul1.2, target = fadul1.1, theta = .))
```

The `comparison_allTogether` function consists of the following steps wrapped into a single convenience function:

- `comparison_cellDivision`: Divide the reference scan into cells

- `comparison_getTargetRegions`: Extract regions associated with each reference cell from the target scan
- `comparison_calcPropMissing`: Compute missing proportions and filter out cells with a proportion of missing values above the threshold.
- `comparison_standardizeHeights`: Standardize height values
- `comparison_replaceMissing`: Replace missing values
- `comparison_fft_ccf`: Compute CCF and estimated translations using FFT
- `comparison_alignedTargetCell`: Extract a matrix from the target scan corresponding to the region of the target scan to which the reference cell aligns
- `cor`: Calculate the pairwise-complete correlation between each cell pair

The `comparison_allTogether` is called repeatedly while rotating the target scan by a set of rotation angles. When implementing the High CMC decision rule (Tong et al., 2015), both combinations of reference and target scan are examined (e.g. A-B and B-A).

Table 2.2 shows several rows of the data frame output of the `comparison_allTogether` function for the comparison of Fadul 1-1 vs. Fadul 1-2 considering Fadul 1-1 as the reference scan. Although we used a grid of 8×8 cells, there were only 26 cell-region pairs that contained a sufficient proportion of non-missing values (15% in this example). The features derived from the correlation cell procedure (CCF_{max} , Δx , Δy , θ) are then used to measure the similarity between scans.

2.2.4 Decision rule

For each cell on the reference scan, we calculate the translation ($\Delta x, \Delta y$) and cross-correlation across rotations by a set of angles θ of the target scan. The task is to determine whether multiple cells come to a “consensus” on a particular translation and rotation. If such a consensus is reached, then there is evidence that a true aligning translation and rotation exists and the cartridge cases match. The CMC decision rules principally differ in

Table 2.2: Example of output from correlation cell comparison procedure between Fadul 1-1 and Fadul 1-2 rotated by -24 degrees. Due to the large proportion of missing values that are replaced to compute the FFT-based correlation, the pairwise-complete correlation is most often greater than the FFT-based correlation.

Cell Index	Pairwise-comp. corr.	FFT-based corr.	Δx	Δy	θ
2, 7	0.432	0.228	-14	-33	-24
2, 8	0.464	0.176	9	-44	-24
3, 1	0.841	0.478	-7	15	-24
3, 8	0.699	0.277	-7	5	-24
4, 1	0.850	0.375	-4	11	-24

how they identify consensus among the $\Delta x, \Delta y, \theta$ values. Here, we describe the two pipelines implemented in the cmcR package: using the original decision rule described in Song et al. (2014) and the High CMC decision rule proposed in Tong et al. (2015).

2.2.4.1 The Original CMC decision rule

This section briefly describes the decision rule used in the first CMC paper (Song, 2013). For a thorough explanation of the procedure, refer to the [CMC Decision Rule Description vignette](#) of the cmcR package.

Let x_i, y_i, θ_i denote the translation and rotation parameters which produce the highest CCF for the alignment of cell-region pair i , $i = 1, \dots, n$ where n is the total number of cell-region pairs containing a sufficient proportion of non-missing values. Song (2013) propose the median as a consensus $(x_{\text{ref}}, y_{\text{ref}}, \theta_{\text{ref}})$ across the cell-region pairs. Then, the distance between each (x_i, y_i, θ_i) and $(x_{\text{ref}}, y_{\text{ref}}, \theta_{\text{ref}})$ is compared to thresholds $T_x, T_y, T_\theta, T_{\text{CCF}}$. A cell-region pair i is declared a “match” if all of the following conditions hold:

Table 2.3: Different thresholds for translation, rotation, and CCF_{\max} are used across different papers. The range in CCF_{\max} is particularly notable.

Paper	Translation T_x, T_y (in pixels)	Rotation θ (in degrees)	CCF_{\max}
Song et al. (2014)	20	6	0.60
Tong et al. (2014)	30	3	0.25
Tong et al. (2015)	15	3	0.55
Chen et al. (2017)	20	3	0.40
Song et al. (2018)	20	6	0.50

$$|x_i - x_{\text{ref}}| \leq T_x, \quad (2.1)$$

$$|y_i - y_{\text{ref}}| \leq T_y,$$

$$|\theta_i - \theta_{\text{ref}}| \leq T_\theta,$$

$$CCF_{\max,i} \geq T_{CCF}.$$

The number of matching cell-region pairs, the “CMC count,” is used as a measure of similarity between the two cartridge cases. Song et al. (2014) indicate that the thresholds $T_x, T_y, T_\theta, T_{CCF}$ need to be determined experimentally. Table 2.3 summarizes the thresholds used in various CMC papers.

Unlike the original CMC pipeline, the High CMC decision rule considers multiple rotations for each cell-region pair.

2.2.4.2 The High CMC decision rule

For the High CMC decision rule, two scans are compared in both directions - i.e., each scan takes on the role of the reference scan that is partitioned into a grid of cells. Tong et al. (2015) claim that some matching cell-region pairs “may be mistakenly excluded from the CMC count” under the original decision rule because they attain the largest CCF at a rotation outside the range allowed by T_θ “by chance.”

Tong et al. (2015) introduce consensus values across all cell-region pairs for each rotation angle θ and calculate a θ -dependent CMC count as the sum of matches observed. Under the High CMC rule, a cell-region pair i is defined as a match conditional on a particular rotation θ if it satisfies the following three conditions:

$$\begin{aligned} |x_{i,\theta} - x_{ref,\theta}| &\leq T_x \\ |y_{i,\theta} - y_{ref,\theta}| &\leq T_y \\ \text{CCF}_{i,\theta} &\geq T_{\text{CCF}}. \end{aligned} \tag{2.2}$$

The θ -dependent CMC count, CMC_θ , is defined as the sum of matching cell-region pairs.

Tong et al. (2015) assert that for a truly matching cartridge case pair, the relationship between θ and CMC_θ should exhibit a “prominent peak” near the true rotation value. That is, CMC_θ should be largest when the scans are close to being correctly aligned. Further, non-matching pairs should exhibit a “relatively flat and random [...] pattern” across the CMC_θ values.

To determine whether a “prominent peak” exists in the relationship between θ and CMC_θ , Tong et al. (2015) consider an interval of rotation angles with large associated CMC_θ values. Let $\text{CMC}_{\max} = \max_\theta \text{CMC}_\theta$ be the maximum CMC_θ count across all rotation angles. For $\tau > 0$, define $S(\tau) = \{\theta : \text{CMC}_\theta > (\text{CMC}_{\max} - \tau)\}$ as the set of rotations with “large” CMC_θ

values. Tong et al. (2015) consider the “angular range” as $R(\tau) = |\max_\theta S(\tau) - \min_\theta S(\tau)|$. If $R(\tau)$ is small, then there is evidence that many cells agree on a single rotation and that the scans match. To arrive at a CMC count similarity score, Tong et al. (2015) suggest a value for τ of 1 and determine:

If the angular range of the “high CMCs” is within the range T_θ , identify the CMCs for each rotation angle in this range and combine them to give the number of CMCs for this comparison in place of the original CMC number.

If the angular range is larger than T_θ , we say that the cartridge case pair “fails” the High CMC criteria and the original CMC number is used. The High CMC decision rule returns a CMC count at least as large as the original decision rule.

2.2.4.3 Implementation of decision rules

In this section, we implement the decision rules in cmcR for both the original and High CMC decision rules. For illustrative purposes, we consider a set of thresholds: $T_x = T_y = 20$, $T_\theta = 6$, and $T_{CCF} = 0.5$.

Decision rules in cmcR are implemented as functions of the form `decision_*`. In particular, the `decision_CMC` function applies both the original and High CMC decision rules depending on if the parameter τ is set. The code below demonstrates the use of `decision_CMC` on the features `kmComparisonFeatures`, extracted from the comparison of scans Fadul 1-1 vs. Fadul 1-2. Conversely, `kmComparisonFeatures_rev` contains the features from a comparison of Fadul 1-2 vs. Fadul 1-1. For comparison, we also compute the CMCs under both decision rules for the comparison between the non-match pair Fadul 1-1 and Fadul 2-1 (not shown to avoid redundancy).

```

kmComparison_cmcs <- kmComparisonFeatures %>% mutate(
  originalMethodClassif =
    decision_CMC(cellIndex = cellIndex, x = x, y = y, theta = theta,
      corr = pairwiseCompCor, xThresh = 20, thetaThresh = 6,
      corrThresh = .5),
  highCMCClassif =
    decision_CMC(cellIndex = cellIndex, x = x, y = y, theta = theta,
      corr = pairwiseCompCor, xThresh = 20, thetaThresh = 6,
      corrThresh = .5, tau = 1))

```

We use the `cmcPlot` function to visualize congruent matching cells (CMCs) and non-congruent matching cells (non-CMCs). Figure 2.9 shows the CMCs and non-CMCs in blue and red, respectively, based on the original decision rule. The (red) non-CMC patches are shown in the position where the maximum CCF value in the target scan is attained. The top row shows 18 CMCs in blue and 8 non-CMCs in red when Fadul 1-1 is treated as the reference and Fadul 1-2 the target. The bottom row shows the 17 CMCs and 13 non-CMCs when the roles are reversed. There is no discussion in Song (2013) about combining the results from these two comparison directions, but Tong et al. (2015) propose using the minimum of the two CMC counts (17 in this example).

Similarly, CMCs and non-CMCs determined under the High CMC decision rule are shown in Figure 2.10. Treating Fadul 1-1 and Fadul 1-2 as the reference scan yields 20 and 18 CMCs, respectively. Combining the results as described above, the final High CMC count is 24.

In contrast, Figure 2.11 shows the CMC results for a comparison between Fadul 1-1 and a known non-match scan, Fadul 2-1, under the exact same processing conditions. Only two cells are classified as congruent matching cells under the original decision rule when Fadul 1-1 is the reference scan. No cells are classified as CMCs in the other direction. While

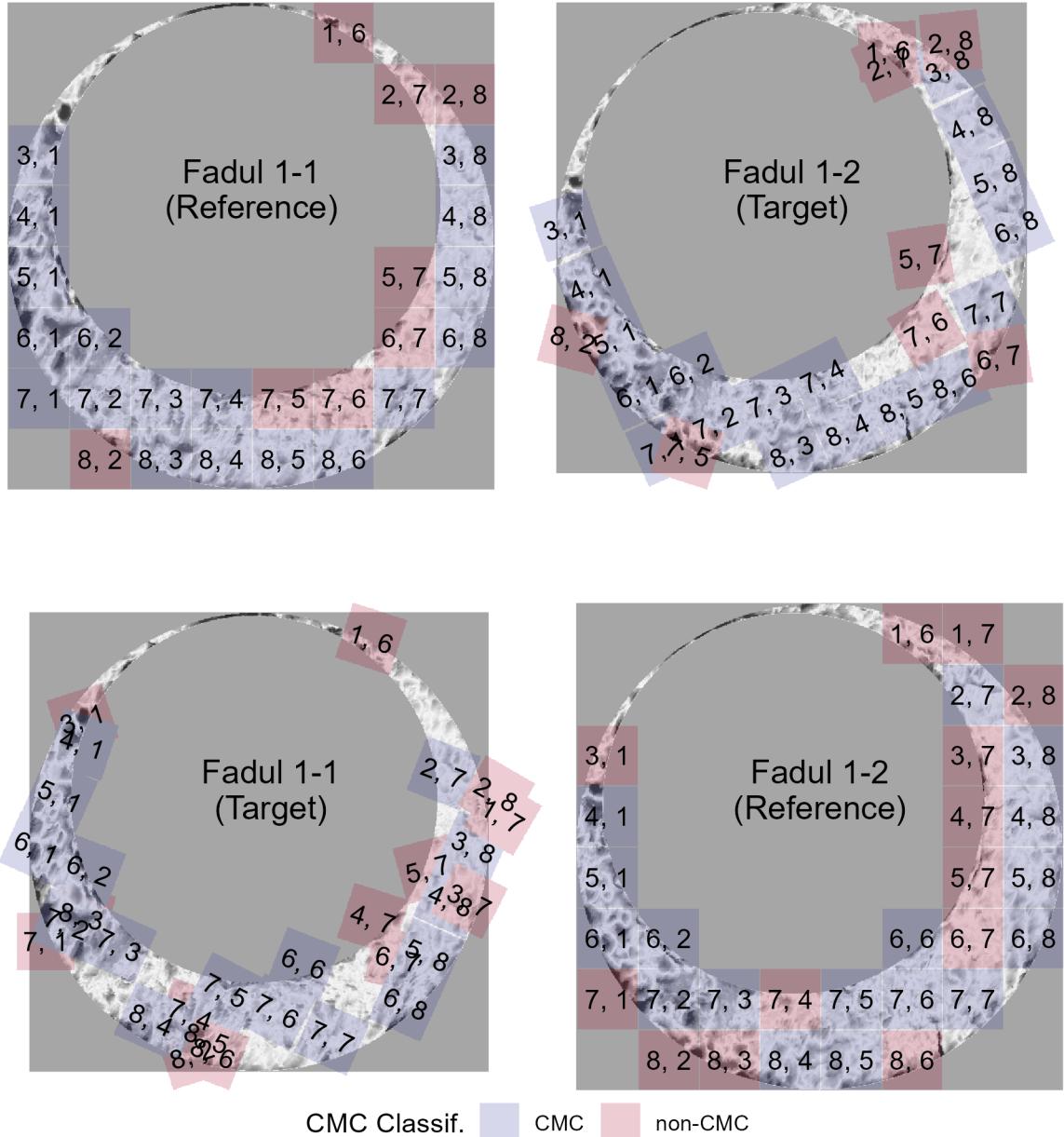


Figure 2.9: CMC results for the comparison between Fadul 1-1 and Fadul 1-2 using the original decision rule. The two plots in the top row show the 18 CMCs when Fadul 1-1 is treated as the "reference" cartridge case to which Fadul 1-2 (the "target") is compared. The second row shows the 17 CMCs when the roles are reversed. Red cells indicate where cells not identified as congruent achieve the maximum pairwise-complete correlation across all rotations of the target scan.

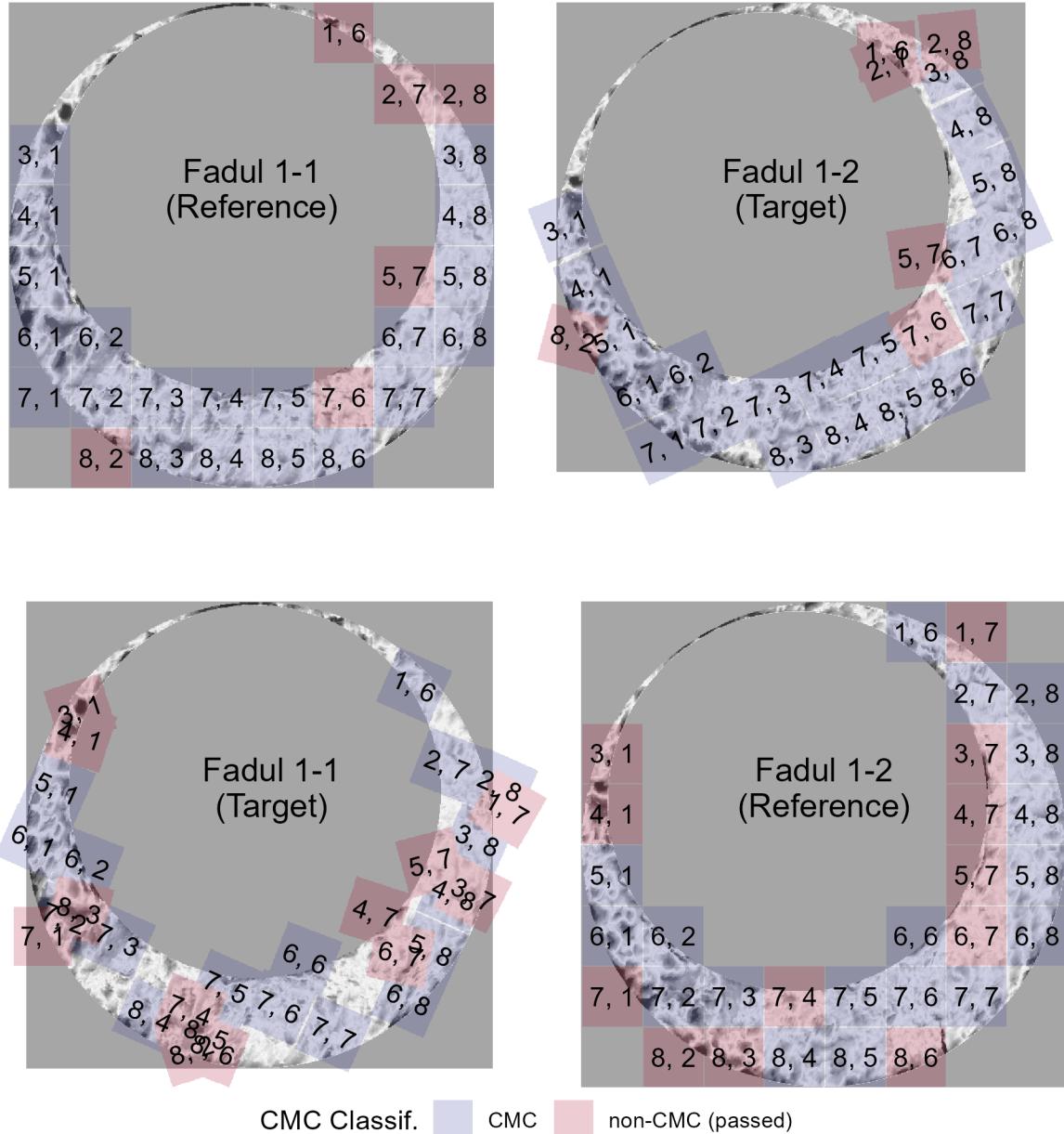


Figure 2.10: Applying the High CMC decision rule to the comparison of Fadul 1-1 and Fadul 1-2 results in 20 CMCs when Fadul 1-1 is treated as the reference (top) and 18 CMCs when Fadul 1-2 is treated as the reference (bottom). Although the individual comparisons do not yield considerably more CMCs than under the original CMC pipeline, Tong et al. (2015) indicate that the High CMCs from both comparisons are combined as the final High CMC count (each cell is counted at most once). Combining the results means that the High CMC decision rule tends to produce higher CMC counts than the original CMC pipeline. In this example, the combined High CMC count is 24 CMCs.

not shown, this pair fails the High CMC criteria and thus was assigned 0 CMCs under the High CMC decision rule.

2.3 Discussion

2.3.1 Ambiguity in algorithmic descriptions

During the implementation process we encountered ambiguous descriptions of the various CMC pipelines. We include the pre-processing and cell-based comparison procedures in the description of CMC methodology to emphasize how sensitive the final results are to decisions made in these first two steps. The pre-processing and cell-based comparison procedures are discussed only briefly, if at all, in Song et al. (2014), Tong et al. (2014), Tong et al. (2015), or Chen et al. (2017). However, the results reported often indicate a sensitivity to these procedures. Ambiguities range from minor implicit parameter choices (e.g., the convergence criteria for the robust Gaussian regression filter (Brinkman and Bodschwinna, 2003a)) to procedures that are fundamental to feature calculation (e.g., how the cross-correlation is calculated). We bring up these ambiguities to demonstrate the difficulties that we faced when translating the conceptual description of the CMC pipeline into an actual pipeline. While many of these choices are unlikely to affect the results dramatically, we believe that any amount of variability that exists solely because of uncertainty in how the method was intended to be implemented is both unnecessary and dangerous in this application.

The only solution to such ambiguity is to enumerate, implement, and pare-down the possible choices that could have been made to arrive to published results. Unsurprisingly, this process takes a considerable amount of time and resources that would be better spent furthering the state of the field. During the creation of the `cmcR` package, the process of re-implementing the comparison and decision steps of the pipeline was fairly straightforward. Emulating the pre-processing procedures used, on the other hand, took months of trial and

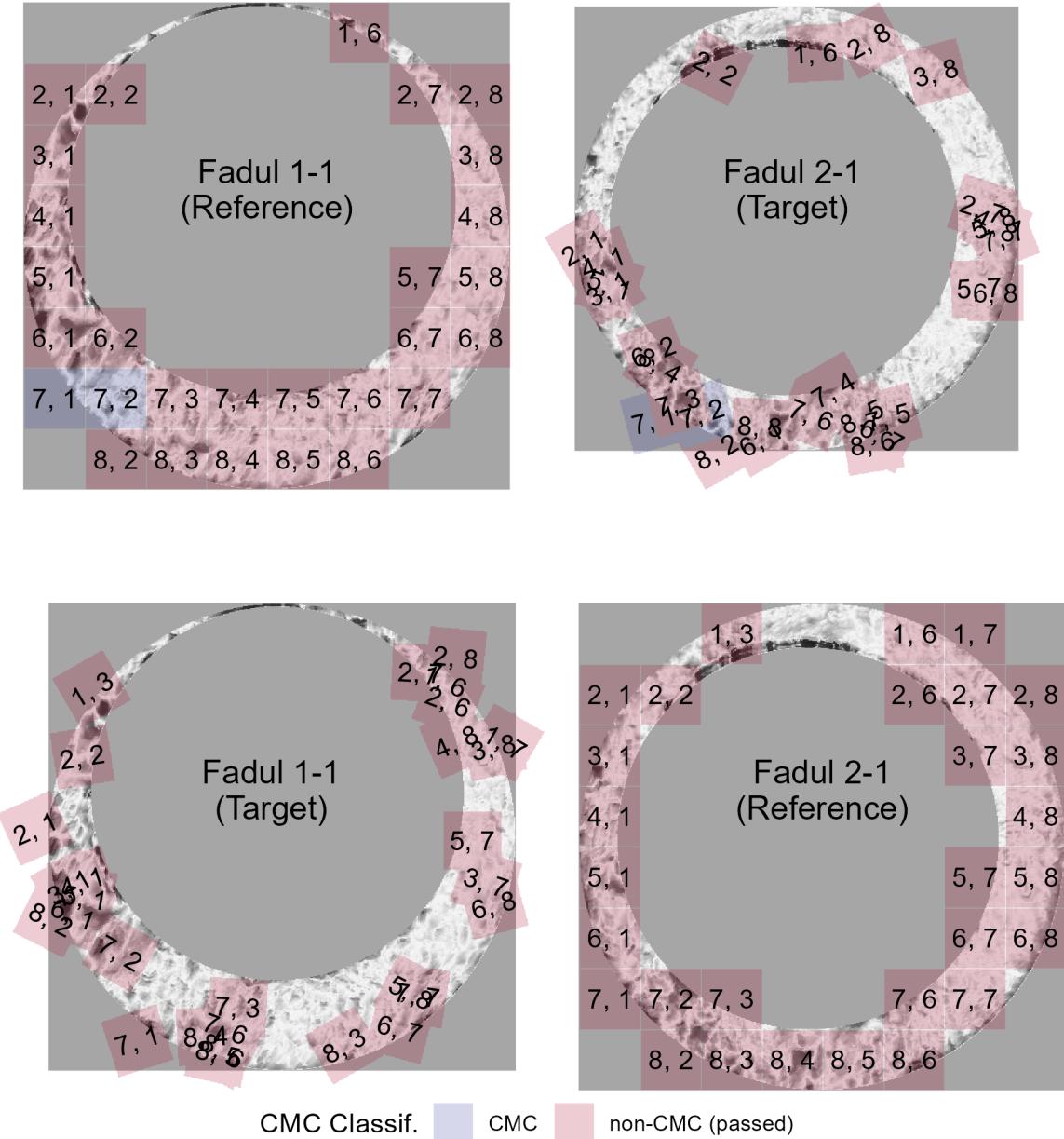


Figure 2.11: Applying both decision rules to the comparison between the non-match pair Fadul 1-1 and Fadul 2-1 results in 2 CMCs under the original decision rule (shown above) and 0 CMCs under the High CMC decision rule (not shown). The seemingly random behavior of the red cells exemplifies the assumption that cells in a non-match comparison do not exhibit an observable pattern. Random chance should be the prevailing factor in classifying non-match cells as CMCs.

error. Even after this effort, we still have no assurances that our implementation would match the results of the original implementation if applied to other data sets.

In the next section, we describe the process of resolving these ambiguities in the CMC pipeline descriptions. In doing so, we abstract a set of principles by which pipelines and results can be rendered both computationally reproducible and more thoroughly understood.

2.3.2 CMC pattern matching pipeline

As described in the initial data section, the set of cartridge case scans from Fadul et al. (2011b) is commonly used to compare the performance of various classification methods (Song et al., 2014; Tong et al., 2015; Chen et al., 2017). This set consists of 40 cartridge cases and 780 total comparisons: 63 known match comparisons and 717 known non-match comparisons. Scans of each breech face impression were taken with a Nanofocus Confocal Light Microscope at 10 fold magnification for a nominal lateral resolution of 3.125 microns per pixel and published to the NBTRD (Zheng et al., 2016). We also use the Weller et al. (2012) data set of 95 cartridge cases for comparison. For the Weller et al. (2012) dataset, we manually isolated the breech face impression regions using the FiX3P software (accessible here: <https://github.com/talenfisher/fix3p>). We compare results from the cmcR package to published results using processed scans available through the Iowa State University DataShare repository (Zemmels et al., 2022b). Our goal is to show that results obtained from cmcR are similar, at least qualitatively, to previously published results. However, justification for any differences will ultimately involve educated guesses due to the closed-source nature of the original implementations.

For each cartridge case pair, we calculate CMC counts under both the original and High CMC decision rules. In practice, we classify a cartridge case pair as “matching” if its CMC count surpasses some threshold; 6 CMCs being the generally accepted threshold in many papers (Tong et al., 2015; Song et al., 2018; Song, 2013). However, this threshold

has been shown to not generalize well to all proposed methods and cartridge case data sets (Chen et al., 2017). We instead use an optimization criterion to select parameters. In doing so, we will demonstrate the sensitivity of the pipeline to parameter choice. Additionally, we introduce a set of principles designed to reduce the need for brute-force searches across parameter settings when re-implementing algorithms without accompanying code. Adherence to these principles yields not only computationally reproducible results, but also improves a reader’s understanding of a proposed pipeline.

2.3.3 Processing condition sensitivity

Choosing threshold values $T_x, T_y, T_\theta, T_{CCF}$ for translation, rotation, and maximum cross-correlation is crucial in declaring a particular cell-region pair “congruent.” However, many combinations of these thresholds yield perfect separation between the matching and non-matching CMC count distributions. Therefore, choosing parameters based on maximizing classification accuracy does not lead to an obvious, single set of parameters. We instead consider the ratio of between- and within-group variability to measure separation between match and non-match CMC counts.

Let C_{ij} denote the CMC count assigned to the j th cartridge case pair, $j = 1, \dots, n_i$ from the i th group, $i = 1, 2$ representing matches and non-matches, respectively. For each set of thresholds we calculate the **Variance Ratio r** as:

$$r = r(T_x, T_y, T_\theta, T_{CCF}) = \frac{\sum_{i=1}^2 (\bar{C}_{i\cdot} - \bar{C}_{..})^2}{\sum_{i=1}^2 \frac{1}{n_i-1} \sum_{j=1}^{n_i} (C_{ij} - \bar{C}_{i\cdot})^2},$$

where $\bar{C}_{i\cdot}$ denotes the within-group CMC count average and $\bar{C}_{..}$ denotes the grand CMC count average. Greater separation between and less variability within the match and non-match CMC count distributions yields larger r values.

For example, Figure 2.12 shows results for the original decision rule and the High CMC decision rule for parameters $T_x = 20 = T_y$ pixels, $T_{CCF} = 0.5$, and $T_\theta = 6$. Despite both decision rules resulting in separation between the matching and non-matching CMC count

distributions, the High CMC decision rule yields greater separation as evidenced by the larger r value.

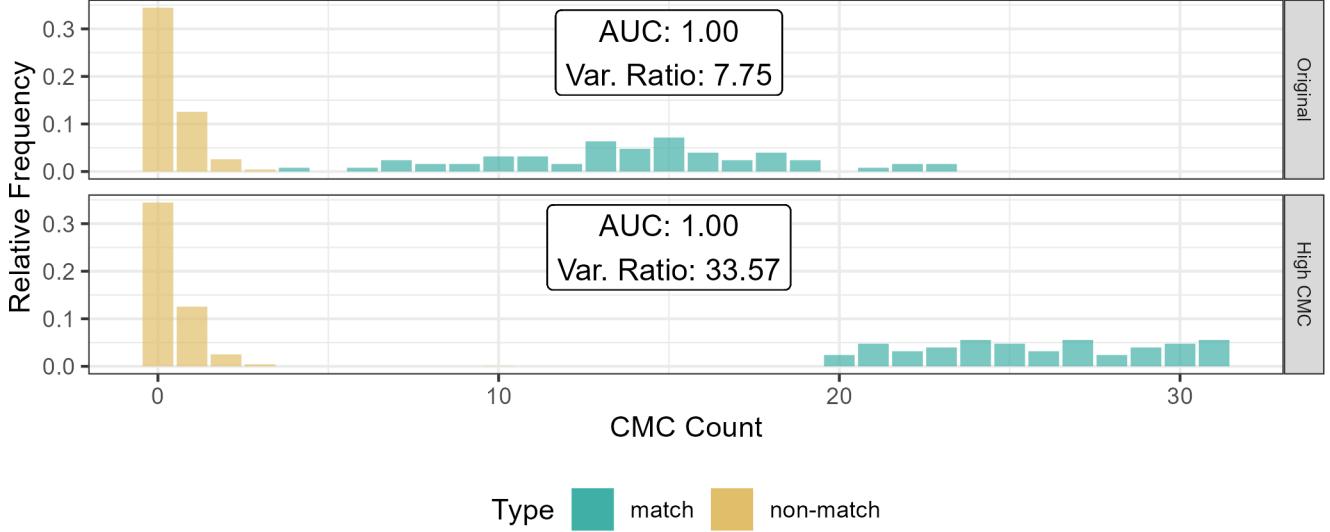


Figure 2.12: CMC count relative frequencies under the original decision rule and the High CMC decision rule for $T_{\Delta x} = 20 = T_{\Delta y}$ pixels, $T_{CCF} = 0.5$, and $T_\theta = 6$ degrees. An $AUC = 1$ corresponds to perfect separation of the match and non-match CMC count distributions. We can see that, for this set of processing parameters, the High CMC decision rule yields higher CMC counts for known matches than the original decision rule while known non-matches have the same distribution under both methods.

To explore the pipeline’s sensitivity, we consider five dimensions that have a demonstrable impact on CMC counts:

- the decision rule (original or High CMC) used,
- whether the global trend is removed during pre-processing, and
- choice of congruency thresholds: translation T_x, T_y , rotation T_θ , and cross-correlation T_{CCF} .

Choosing a single parameter setting that results in perfect identification is not enough to generally understand the algorithm. Instead, we use the variance ratio r to identify promising ranges of parameters. Figure 2.13 shows the value of the variance ratio under

different parameter settings. We see that the High CMC decision rule yields better separation than the original decision rule under any parameter setting. The largest variance ratio values are achieved for thresholds $T_x, T_y \in [10, 20]$, $T_\theta = 6$, and $T_{CCF} \in [0.4, 0.5]$. Interestingly, considering Table 2.3, only the parameters used in Song et al. (2018) fall into these ranges.

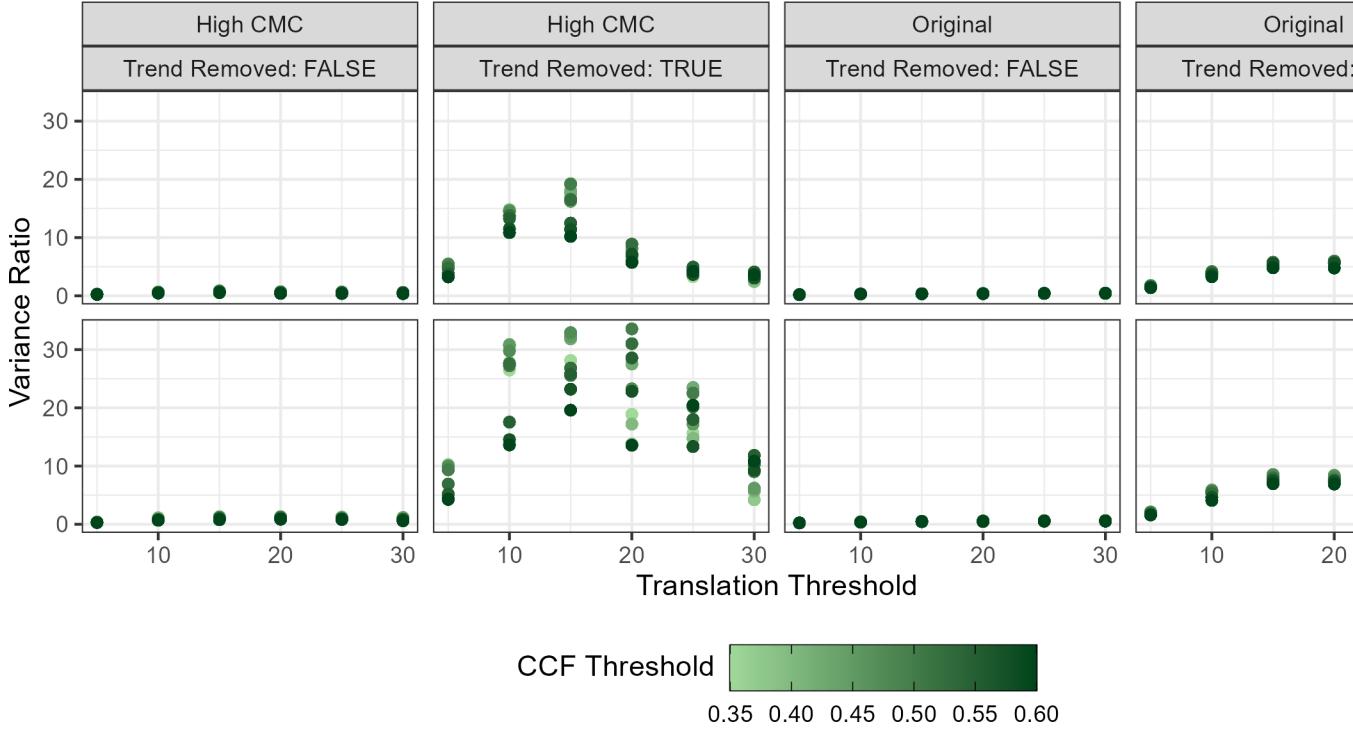


Figure 2.13: Variance ratio values are plotted for different parameter settings. High variance ratios are indicative of a good separation between CMC counts for known matching pairs and known-non matching pairs. The High CMC decision rule generally performs better than the original decision rule. Removing the trend during pre-processing has a major impact on the effectiveness of the CMC pipeline. In this setting, translation thresholds $T_x, T_y \in [15, 20]$, a rotation threshold $T_\theta = 6$, and a CCF threshold $T_{CCF} \in [0.4, 0.5]$ lead to a separation of results.

As shown in Figure 2.13, de-trending breech-scans in the pre-processing stage emerges as a critical step to achieve good algorithmic results. This step is not explicitly mentioned in the written-word descriptions of the algorithm in Song (2013), Tong et al. (2014), Tong et al. (2015), Chen et al. (2017), or Song et al. (2018), though it appears from their examples that it was used in the process. Figure 2.13 also illustrates how breaking a pipeline up into modularized steps eases experimentation. We will expand upon this idea in the next section.

We compare the best results from cmcR to results presented in previous papers. In particular, we have calculated variance ratio statistics shown in Figure 2.14 based on CMC counts reported in Song (2013), Tong et al. (2014), Tong et al. (2015), Chen et al. (2017), and Song et al. (2018). The last row in each facet shows the variance ratio values obtained from cmcR. We see that the implementation provided in cmcR yields comparable results to previous CMC papers.

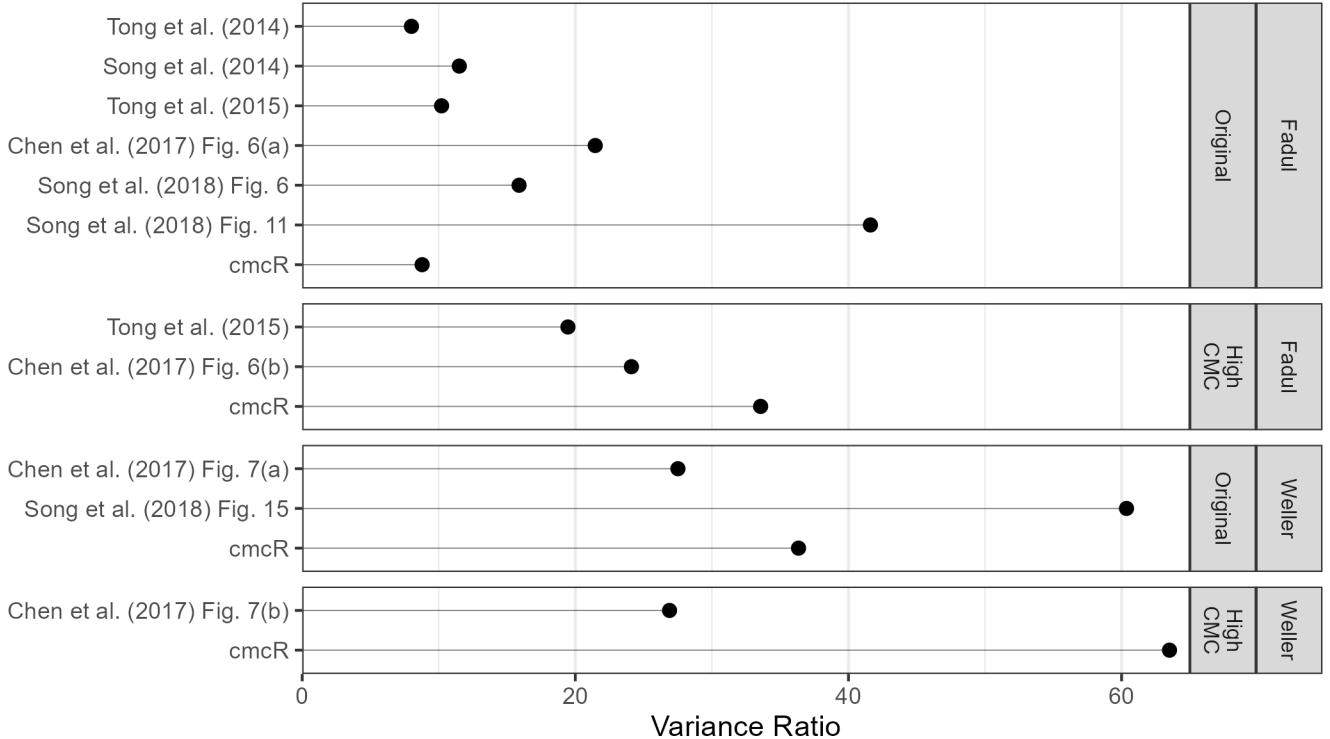


Figure 2.14: Variance ratios based on results reported in various CMC papers. The High CMC decision rule tends to outperform the original decision rule. However, it should be emphasized that each paper uses very different processing and parameter settings meaning the results are difficult to compare. The values labeled "cmcR" show the largest variance ratio values for the original and High CMC decision rules based on a limited grid search. These results indicate that the CMC pipeline implementation provided in cmcR yields comparable results to previous CMC papers.

2.4 Conclusion

Reproducibility is an indispensable component of scientific validity (Goodman et al., 2016). In this paper, we demonstrate at least three ways reproducibility can go awry: ambiguity in procedural implementation, missing or incomplete data, and missing or incomplete code. In forensics, many matching algorithms are commonly presented in the form of conceptual descriptions with accompanying results. There is sound reasoning to this; conceptual descriptions are more easily understood by humans compared to computer code. However, using the CMC pipelines as an example we have observed the gaps that can exist when translating a conceptual description of an algorithm to a genuine implementation. This is largely due to the fact that conceptual descriptions rarely detail implicit parameter choices required to run an algorithm. Consequently, there are multiple choices that are compatible with the description of an algorithm in a publication. This is dangerous in a forensics context because if many parameter settings are valid but only a narrow range lead to the same conclusion, it is entirely possible that witnesses for the prosecution and defense come to different conclusions. In order to prevent such misunderstandings, it is not enough to have guidelines for parameter settings and/or a sensitivity study – it is also necessary to standardize the specific computer code. The parameter values are only useful within the context of a single software package or pipeline.

These principles of open, accessible, interoperable code are also critical for a fair (in the legal sense) justice system: the defense has access to the code to understand the evidence against them, lawyers and examiners can assess the utility of the analysis method, and judges can determine whether a method is admissible in court. Transparent and intuitive open-source algorithms, such as cmcR, should be considered the gold standard in allowing the forensic science community to validate a pipeline.

Our contribution to the CMC literature is the open-source implementation, which fills the gaps in the human-friendly descriptions in the original papers. In addition, because we

have structured the cmcR implementation as a modular pipeline, it is easier to improve upon the CMC method and document the effects of specific changes to the algorithm compared to previous versions. The modularization creates an explicit framework to assess the utility and effectiveness of each piece of the algorithm, and allows us to independently manipulate each step while monitoring the downstream impact on the results. Additionally, it allows future collaborators to improve on pieces of the pipeline, adding new options and improving the method without having to re-invent the wheel. Indeed, re-implementing steps of the pipeline is at best a useful academic exercise and at worst a waste of time and resources that could be spent actually improving the pipeline. Even after many months of trial and error, although we have succeeded in obtaining qualitatively similar results on two data sets, it is difficult to know whether our implementation will behave the same as previous implementations on external data sets. Generalizability is an important assessment for any computational algorithm (Vanderplas et al., 2020).

Our application is far from unique: some journals have adopted policies encouraging or requiring that authors provide code and data sufficient to reproduce the statistical analyses, with the goal of building a “culture of reproducibility” in their respective fields (Peng, 2009, 2011; Stodden et al., 2013). Peer-review and scientific progress in the truest sense requires that *all* pre-processed data, code, and results be made openly available (Kwong, 2017; Desai and Kroll, 2017). Our experience with the CMC algorithm suggests that these standards should be adopted by the forensic science community, leveraging open-source ecosystems like R and software sharing platforms such as Github. We firmly believe that the forensic community should not go only halfway, trading a subjective, human black box for objective, proprietary algorithms that are similarly opaque and unauditible. Open, fully reproducible packages like cmcR allow research groups to make incremental changes, compare different approaches, and accelerate the pace of research and development.

2.5 Acknowledgement

This work was partially funded by the Center for Statistics and Applications in Forensic Evidence (CSAFE) through Cooperative Agreement 70NANB20H019 between NIST and Iowa State University, which includes activities carried out at Carnegie Mellon University, Duke University, University of California Irvine, University of Virginia, West Virginia University, University of Pennsylvania, Swarthmore College and University of Nebraska, Lincoln.

We greatly appreciate the constructive feedback from the two anonymous reviewers. Special thanks also to all the developers and open-source contributors of R, knitr (Xie, 2015, 2014b), rctiles (Allaire et al., 2021), and the tidyverse (Wickham et al., 2019), without whom this project would not have been possible.

2.6 Computational details

```
#> R version 4.2.2 (2022-10-31 ucrt)
#> Platform: x86_64-w64-mingw32/x64 (64-bit)
#> Running under: Windows 10 x64 (build 22621)
#>
#> Matrix products: default
#>
#> locale:
#> [1] LC_COLLATE=English_United States.utf8
#> [2] LC_CTYPE=English_United States.utf8
#> [3] LC_MONETARY=English_United States.utf8
#> [4] LC_NUMERIC=C
#> [5] LC_TIME=English_United States.utf8
#>
```

```

#> attached base packages:

#> [1] stats      graphics   grDevices datasets  utils      methods
#> [7] base

#>

#> other attached packages:

#> [1] patchwork_1.1.2 magrittr_2.0.3  rgl_1.0.1       x3ptools_0.0.3
#> [5] lubridate_1.9.2forcats_1.0.0  stringr_1.5.0    dplyr_1.1.0
#> [9] purrrr_1.0.1   readr_2.1.4     tidyverse_2.0.0 cmcR_0.1.11   ggplot2_3.4.1
#> [13] tidyverse_2.0.0 cmcR_0.1.11   ggplot2_3.4.1
#>

#> loaded via a namespace (and not attached):

#> [1] httr_1.4.5        jsonlite_1.8.4    viridisLite_0.4.0
#> [4] splines_4.2.2     assertthat_0.2.1  yulab.utils_0.0.6
#> [7] renv_0.17.0       tiff_0.1-11      yaml_2.3.5
#> [10] pillar_1.8.1     lattice_0.21-8   quantreg_5.94
#> [13] glue_1.6.2       digest_0.6.29    rvest_1.0.3
#> [16] colorspace_2.0-3 readbitmap_0.1.5 cowplot_1.1.1
#> [19] htmltools_0.5.4  Matrix_1.5-3     ggnewscale_0.4.8
#> [22] pkgconfig_2.0.3  SparseM_1.81    magick_2.7.4
#> [25] bookdown_0.26    scales_1.2.0     webshot_0.5.4
#> [28] ggplotify_0.1.0  svglite_2.1.1   jpeg_0.1-10
#> [31] tzdb_0.3.0       pracma_2.4.2    MatrixModels_0.5-1
#> [34] timechange_0.2.0  generics_0.1.2   farver_2.1.0
#> [37] ellipsis_0.3.2   withr_2.5.0     cli_3.6.0
#> [40] survival_3.5-5   evaluate_0.15   fansi_1.0.3
#> [43] MASS_7.3-55      imager_0.42.18  xml2_1.3.3
#> [46] textshaping_0.3.6 tools_4.2.2     hms_1.1.2

```

```
#> [49] bmp_0.3                 lifecycle_1.0.3    munsell_0.5.0
#> [52] kableExtra_1.3.4       compiler_4.2.2     gridGraphics_0.5-1
#> [55] systemfonts_1.0.4      rlang_1.0.6       grid_4.2.2
#> [58] rstudioapi_0.14        htmlwidgets_1.6.1  igraph_1.4.1
#> [61] labeling_0.4.2          base64enc_0.1-3   rmarkdown_2.14
#> [64] gtable_0.3.0           codetools_0.2-18  R6_2.5.1
#> [67] gridExtra_2.3           zoo_1.8-11       knitr_1.39
#> [70] fastmap_1.1.0          utf8_1.2.2       ragg_1.2.5
#> [73] stringi_1.7.6          Rcpp_1.0.10      vctrs_0.5.2
#> [76] png_0.1-8              tidyselect_1.2.0  xfun_0.38
```

CHAPTER 3. Diagnostic Tools for Cartridge Case Comparison Algorithms

Abstract

Diagnostics are useful for explaining the behavior of a complex model or algorithm. In this paper, we introduce a suite of visual and interactive diagnostic tools to assess algorithms that automatically compare forensic cartridge case evidence. These tools are useful to both forensic researchers and practitioners: Researchers can analyze and correct the (mis)behavior of the algorithm forensic practitioners can more easily understand and explain the algorithm. First, we develop a collection of tools to diagnose each step of a cartridge case comparison algorithm in the `impression` R package. We then implement these visual diagnostics in an interactive web application called `cartridgeInvestigatR` to provide greater flexibility to the user.

3.1 Introduction

Start of paper.

CHAPTER 4. Automatic Matching of Cartridge Case Impressions

Abstract

Forensic examinations attempt to solve the binary classification problem of whether two pieces of evidence originated from the same source. For example, a cartridge case found at a crime scene may be compared to a cartridge case fired from a suspect's firearm. Forensic examiners traditionally rely on high-powered comparison microscopes, case facts, and their own experience to arrive at a source conclusion. Automatic comparison algorithms have grown in prevalence in a number of forensic disciplines following the reports from National Research Council (2009) and President's Council of Advisors on Sci. & Tech. (2016). Many of these algorithms objectively measure the similarity between evidence, such as two fired cartridge cases, based on markings left on their surface, such as impressions left by a firearm's breech face during the firing process. We introduce the Automatic Cartridge Evidence Scoring (ACES) algorithm to compare pairs of three-dimensional topographical surface scans of breech face impressions. The ACES algorithm pre-processes the scans, extracts numeric features, and returns a similarity score indicating whether two cartridge cases were fired from the same firearm. The numeric features are computed based on a cell-by-cell registration procedure, results from a density-based unsupervised clustering algorithm, and derived from visual diagnostic tools we developed to investigate the performance of cartridge case comparison algorithms. We use scans taken at the Roy J Carver High Resolution Microscopy Facility of cartridge cases collected by Baldwin et al. (2014) to train and test the ACES algorithm. The performance of ACES compares favorably to several

other methods, such as random forests on smaller feature sets, logistic regressions, decision trees, and some variants of previous Congruent Matching Cells methods (Song, 2013; Zhang et al., 2020). The ACES algorithm is implemented in a free, opensource interactive web application called cartridgeInvestigatR.

4.1 Introduction

A *cartridge case* is the part of firearm ammunition that houses the projectile and propulsive device. When a firearm is discharged and the projectile travels down the barrel, the cartridge case moves in the opposite direction and slams against the back wall, the *breech face*, of the firearm. Markings on the breech face are “stamped” into the surface of the cartridge case leaving so-called *breech face impressions*.

In a traditional examination, forensic examiners use these impressions analogous to a fingerprint to determine whether two cartridge cases were fired from the same firearm. The top of Figure 4.1 illustrates this procedure (Xiao Hui Tai, 2018; Zheng et al., 2014; Vorburger et al., 2015). First, two cartridge cases are collected - perhaps one is from a crime scene and the other is collected from a suspect’s gun. An examiner places the two cartridge cases beneath a “comparison microscope” that merges the views of two compound microscopes into a single split view, similar to the side-by-side cartridge case image in Figure 4.1. The examiner assesses the degree of similarity between the markings on the cartridge cases and chooses one of four conclusions AFTE Criteria for Identification Committee (1992):

1. **Identification:** cartridge cases were fired from the same firearm
2. **Elimination:** cartridge cases were not fired from the same firearm
3. **Inconclusive:** the evidence is insufficient to make an identification or elimination
4. **Unsuitable:** the evidence is unsuitable for examination

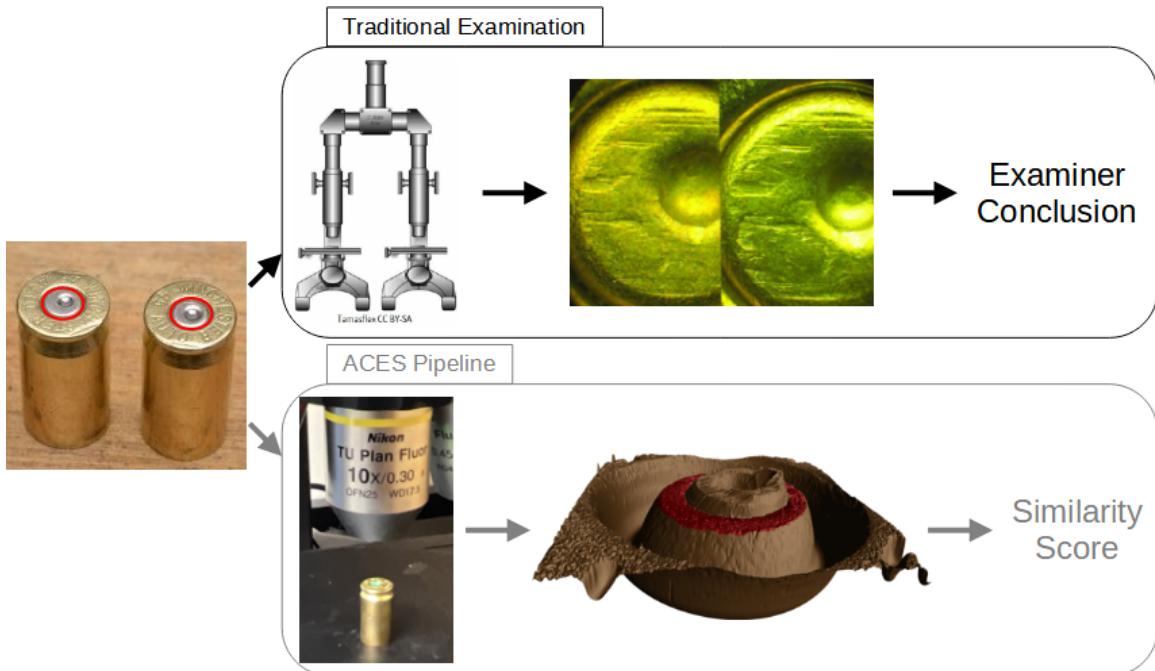


Figure 4.1: Comparison of the traditional examination vs. the currently proposed method for comparing cartridge cases. Both start with two fired cartridge cases. In traditional examination, an examiner uses a microscope to assess the "agreement" of markings on the two cartridge case surfaces. They decide whether or not the cartridge cases were fired from the same firearm, or if there is inconclusive evidence to decide. In the ACES algorithm, we take a topographical scan of the cartridge case surfaces and manually identify the regions containing distinguishable markings (shown in red). We pass these scans to the ACES algorithm, which processes and compares the two scans. The final result is a numerical measure of similarity of the two cartridge cases.

Critics of traditional forensic examinations cite a lack of “foundational validity” underlying the procedures used by firearm and toolmark examiners (National Research Council, 2009; President’s Council of Advisors on Sci. & Tech., 2016). In particular, examiners rely largely on their subjective findings rather than on a well-defined procedure to measure similarity. President’s Council of Advisors on Sci. & Tech. (2016) pushed for “developing and testing image-analysis algorithms” to objectively measure the similarity between cartridge cases. An automatic comparison algorithm could supplement, inform, or dictate an examiner’s conclusion (Swofford and Champod, 2021).

We introduce a novel Automatic Cartridge Evidence Scoring (ACES) algorithm to objectively compare cartridge case evidence based on their breech face impressions. Our algorithm encompasses all stages of the comparison procedure after collecting a scan of the cartridge case surface including pre-processing, comparing, and scoring. Our ACES algorithm is available open-source as part of the [scored](#) R package.

In the following sections, we first review recently proposed algorithms to compare firearm evidence. We then discuss the data collection procedure to obtain 510 cartridge scans used in training and validating the ACES algorithm. To our knowledge, this is the largest published study of a cartridge case comparison algorithm to-date, with the next largest analyzing four different data sets totaling 195 cartridge cases (Chen et al., 2017). After describing the ACES algorithm, we present summary results from training and testing three binary classifier models: base on a random forest, logistic regression, and classification tree. We discuss the strengths and weaknesses of the three classifier models and compare the relative importance of the ACES features. We also argue that the ACES algorithm combines the classification rules of previously proposed cartridge case comparison algorithms while incorporating additional nuance. We conclude with our thoughts on how cartridge case comparison algorithms should be developed, validated, and shared going forward.

4.1.1 Previous Work

Recent proposals for automatic cartridge case scoring algorithms borrow from image processing and computer vision techniques. For example, Vorburger et al. (2007) proposed using the cross-correlation function (CCF) to compare images or scans of cartridge case surfaces. The CCF measures the similarity between two matrices for all possible translations of one matrix against the other. Calculating the CCF while rotating one of the scans therefore allows for estimation of the optimal translation and rotation, together referred to as the *registration*, between the two scans; simply choose the rotation/translation at which the CCF is maximized. Hare et al. (2017) used the CCF, among other features, to compare scans of bullets. Tai and Eddy (2018) developed an open-source cartridge case comparison pipeline that compared cartridge case images using the CCF.

Song (2013) noted that two matching cartridge cases often share similar impressions in specific regions, so calculating the CCF between two full scans may not highlight their similarities. Instead, Song (2013) proposed partitioning one cartridge case scan into a grid of “cells” and calculating the CCF between each cell and the other scan. If two cartridge cases are truly matching, then the maximum CCF value between each cell and the other scan, particularly the cells containing distinguishable breech face impressions, should be relatively large. Furthermore, the cells should “agree” on the registration at which the CCF is maximized. Song (2013) outlined the “Congruent Matching Cells” algorithm to determine the number of cells that agree on a particular registration. A cell is classified as a Congruent Matching Cell (CMC) if its estimated registration is within some threshold of the median registration across all cells and its CCF value is above some threshold. A number of follow-up papers proposed alterations to the the original CMC method (Tong et al., 2015; Chen et al., 2017). Zemmels et al. (2022a) introduced an open-source implementation of the CMC method in the cmcR R package. As an alternative to defining Congruent Matching Cells, Zhang et al. (2020) proposed using a clustering algorithm from Ester et al. (1996) to determine the number of cells in agreement on a specific registration.

Currently, there is no rigorous procedure for comparing different cartridge case comparison algorithms. This includes selecting optimal parameters for a specific algorithm. Zemmels et al. (2023) proposed an optimization criterion to select parameters for the CMC algorithm. Analogously, Hare et al. (2017) developed a validation procedure to select parameters for a bullet comparison algorithm. In this work, we introduce a novel cross-validation procedure to learn and test optimal parameters for the ACES algorithm.

4.2 Cartridge Case Data

We use 510 cartridge cases collected as part of a study by Baldwin et al. (2014). The authors of the original study fired 800 Remington 9mm pistol cartridge cases from each of 25 new Ruger SR9, 9mm Luger centerfire pistols.. They separated the collected cartridge cases into 15 sets of four to be sent to each of 218 forensic examiner participants. Each set of four consisted of three cartridge cases labeled as originating from the same firearm, the “known-match” cartridge cases. Participants performed an examination to determine whether a fourth “questioned” cartridge case shared a common source with the known-match triplet (or whether the evidence was inconclusive).

Across all 218 examiners, the true positive rate - proportion of correctly classified matching sets - was reported to be 99.6%. The reported true negative rate - the proportion of correctly classified non-matching sets - was 65.2% The discrepancy between the true positive and true negative rates can be partially explained by the number of “inconclusive” decisions made by the examiners. Examiners reach an inconclusive decision when there is some agreement or disagreement in the characteristics between two cartridge cases, but not enough to make a match or non-match conclusion (AFTE Criteria for Identification Committee, 1992). Roughly one out of five comparisons, 22.9%, were reported as inconclusive. The vast majority, 98.5%, of these inconclusives were truly non-matching comparisons, which justifies the true negative rate of 65.2%. There has recently been some debate about

how to incorporate inconclusive decisions into accuracy/error rate estimation (Hofmann et al., 2021), so we do not report an overall accuracy here.

We scanned the 510 cartridge cases using the CadreTM 3D-TopMatch High Capacity Scanner. Briefly, this scanner collects images under various lighting conditions of a gel pad into which the base of a cartridge case is impressed. Proprietary software that accompanies this scanner combines these images into a 2D array called a *surface matrix*. The elements of a surface matrix represent the relative height values of the associated cartridge case surface. This surface matrix, along with metadata concerning parameters under which the scan was taken (dimension, resolution, author, etc.), are stored in the ISO standard XML 3D Surface Profile (x3p) file type (ISO 25178-72(2017), 2017).

As discussed in the next section, our design differs from that used in Baldwin et al. (2014). Rather than basing error rates on the comparison of three known-match cartridge cases to one questioned cartridge case (3 to 1), we consider the classification error rate of pairwise comparisons (1 to 1). Further, we split the 510 cartridge cases by randomly selecting 10 of the 25 firearms for training and use the remaining 15 firearms for testing. This resulted in a training set of 210 cartridge cases, $\binom{210}{2} = 21,945$ pairwise comparisons, and a testing set of 300 cartridge cases, $\binom{300}{2} = 44,850$ pairwise comparisons.

[cite eventual DFSC data-in-brief or ISU datashare repo?]

4.3 Methods

We now discuss the methods behind the Automatic Cartridge Evidence Scoring (ACES) algorithm. We divide the methods into three stages:

1. **Pre-processing:** prepare cartridge case scans for comparison
2. **Comparing:** compare two cartridge cases and compute similarity features

3. **Scoring:** measure the similarity between the two cartridge cases using a trained classifier

The following sections detail each of these stages. Throughout, we treat “surface matrix” and “scan” synonymously.

The bottom of Figure 4.1 shows a summary of our procedure. After taking a topographical scan of the cartridge case surfaces, we manually annotate the breech face impression region (shown in red). ACES automatically pre-processes and compares the scans resulting in a similarity score, either a binary classification or class probability, derived from a classifier model.

4.3.1 Pre-processing

We first use the open-source FiX3P web application [[cite Talen Fisher](#)] to manually annotate the breech face impression region. An example of a manually-annotated cartridge case scan is shown in Figure 4.1. The FiX3P software includes functionality to “paint” the surface of a cartridge case using a computer cursor and save the painted regions to a *mask*. A mask is a 2D array of hexadecimal color values of the same dimension as its associated surface matrix. When initialized, every element of a mask is a shade of brown (#cd7f32) by default. Any elements painted over by the user will be replaced with the user’s selected color value. In Figure 4.1, the breech face impression region was manually annotated using a shade of red (#ff0000).

We pre-process the raw scans by applying a sequence of functions available in the R packages `x3ptools` (Hofmann et al., 2020) and `cmcR` (Zemmels et al., 2022a). Figure 4.2 shows the effect that each function has on the scan surface values. Gray pixels in each plot represent missing values in the surface matrix. The `x3p_delete` function removes values in the scan based on the associated mask. Next, the `preProcess_removeTrend` function subtracts a fitted conditional median plane from the surface values to “level-out” any global

tilt in the scan. The `preProcess_gaussFilter()` function applies a bandpass Gaussian filter to remove small-scale noise and other large-scale structure, which better highlights the medium-scale breech face impressions. Finally, the `preProcess_erode()` function applies the morphological operation of erosion on the edge of the non-missing surface values [cite erosion reference]. This has the effect of shaving off values on the interior and exterior edge of the surface, which are often extreme “roll-off” values that unduly affect the comparing stage if not removed. The final result is a cartridge case surface matrix with emphasized breech face impressions.

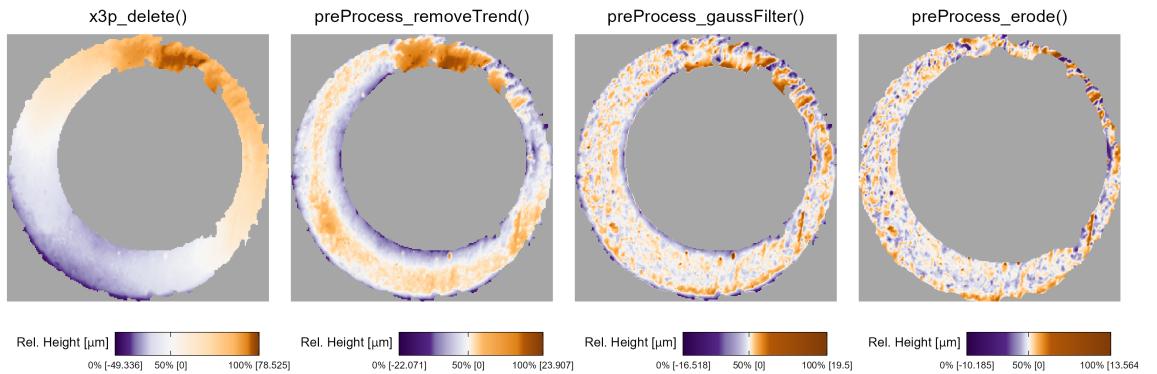


Figure 4.2: We apply a sequence of pre-processing functions to each scan. Each pre-processing step further emphasizes the breech face impressions in the scan.

Next, we compute a set of similarity features for two pre-processed cartridge case scans.

4.3.2 Comparing

In this section, we introduce a set of similarity features for two cartridge case scans. We calculate features at two scales: between two full scans and between individual cells. Analogous to how a forensic examiner uses a comparison microscope with different magnification levels, this allows us to assess the similarity between two scans at the macro and micro levels.

4.3.2.1 Notational Conventions

First, we introduce notation that will be used to define the features. Let A and B denote two surfaces matrices that we wish to compare. For simplicity, we assume that $A, B \in \mathbb{R}^{k \times k}$ for $k > 0$.¹ We use lowercase letters and subscripts to denote a particular value of a matrix: a_{ij} is the value in the i -th row and j -th column, starting from the top-left corner, of matrix A . We refer to the two known-match cartridge cases in Figure 4.3 as exemplar matrices A and B .

To accommodate structurally missing values, we adapt standard matrix algebra as follows: if an element of either matrix A or B is missing, then any element-wise operation including this element is also missing. Standard matrix algebra holds for non-missing elements. For example, the addition operator is defined as:

$$A \oplus_{NA} B = (a_{ij} \oplus_{NA} b_{ij})_{1 \leq i,j \leq k} = \begin{cases} a_{ij} + b_{ij} & \text{if both } a_{ij} \text{ and } b_{ij} \text{ are numbers} \\ NA & \text{otherwise} \end{cases}$$

Other element-wise operations such as \ominus_{NA} are defined similarly. For readability, we will use standard operator notation $+, -, >, <, I(\cdot), \dots$ and assume the extended, element-wise operations as defined above.

4.3.2.2 Registration Estimation

A critical step in comparing A and B is to find a transformation of B such that it aligns best to A (or vice versa). In image processing, this is called *image registration*. Noting that A and B are essentially grayscale images with structurally missing values, we rely on a standard image registration technique (Brown, 1992).

In our application, a registration is composed of a discrete translation by $(m, n) \in \mathbb{Z}^2$ and rotation by $\theta \in [-180^\circ, 180^\circ]$. To determine the optimal registration, we calculate the

¹This assumption of equally-sized, square matrices is easily enforced by padding the matrices with additional missing values. Due to the presence of (structurally) missing values around the breech face impression region, additional padding does not interfere with the structure of the scan.

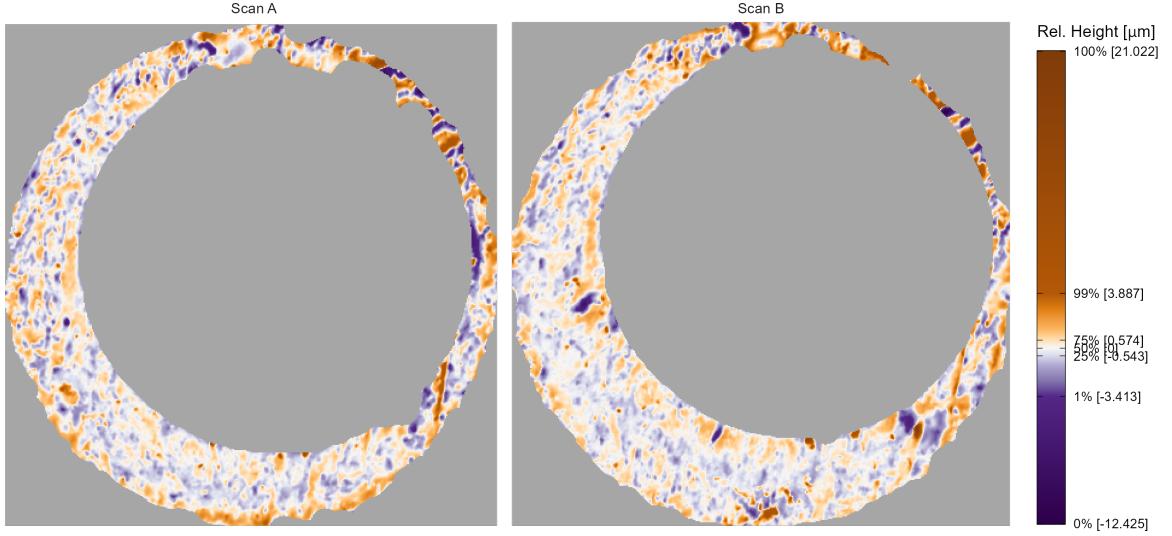


Figure 4.3: A matching pair of processed cartridge case scans. We measure the similarity between these cartridge cases using the distinguishable breech face impressions on their surfaces.

cross-correlation function (CCF) between A and B , which measures the similarity between A and B for every possible translation of B , denoted $(A \star B)$. We estimate the registration by calculating the maximum CCF value across a range of rotations of matrix B . Let B_θ denote B rotated by an angle $\theta \in [-180^\circ, 180^\circ]$ and $b_{\theta mn}$ the m, n -th element of B_θ . Then the estimated registration (m^*, n^*, θ^*) is:

$$(m^*, n^*, \theta^*) = \arg \max_{m, n, \theta} (a \star b_\theta)_{mn}.$$

In practice we consider a discrete grid of rotations $\Theta \subset [-180^\circ, 180^\circ]$. The registration procedure is outlined in Image Registration Algorithm. We refer to the matrix that is rotated as the “target.” The result is the estimated registration of the target matrix to the “source” matrix.

Image Registration Algorithm

Data: Source matrix A , target matrix B , and rotation grid Θ

Result: Estimated registration of B to A , (m^*, n^*, θ^*) , and cross-correlation function maximum CCF_{\max}

for $\theta \in \Theta$ **do**

 Rotate B by θ to obtain B_θ ;

 Calculate $CCF_{\max,\theta} = \max_{m,n}(a \star b)_{mn}$;

 Calculate translation $[m_\theta^*, n_\theta^*] = \arg \max_{m,n}(a \star b_\theta)_{mn}$;

end

Calculate overall maximum correlation $CCF_{\max} = \max_\theta \{CCF_{\max,\theta} : \theta \in \Theta\}$;

Calculate rotation $\theta^* = \arg \max_\theta \{CCF_{\max,\theta} : \theta \in \Theta\}$;

return Estimated rotation θ^* , translation $m^* = m_{\theta^*}^*$, and $n^* = n_{\theta^*}^*$, and CCF_{\max}

To accommodate missing values, we also compute the *pairwise-complete correlation* using only the complete value pairs, meaning neither value is missing, between A and B .

4.3.2.3 Registration-Based Features

4.3.2.3.1 Full-Scan Registration

We first estimate the registration between two full scans A and B using Image Registration Algorithm with a rotation grid $\Theta = \{-30^\circ, -27^\circ, \dots, 27^\circ, 30^\circ\}$. This results in an estimated registration (m^*, n^*, θ^*) and similarity measure CCF_{\max} . We also perform Image Registration Algorithm with the roles of A and B reversed, meaning the target scan A is aligned to source scan B .

To accommodate these two comparison directions, we introduce a new subscript $d = A, B$, referring to the source scan in Image Registration Algorithm. Consequently, we obtain two sets of sets of estimated registrations, $(m_d^*, n_d^*, \theta_d^*)$ and $CCF_{\max,d}$, for $d = A, B$.² For $d = A$,

²In reality, the true aligning registrations in the two comparison directions are opposites of each other. However, because we compare discretely-indexed arrays using a nearest-neighbor interpolation scheme, the estimated registrations differ slightly.

we then apply the registration transformation $(m_A^*, n_A^*, \theta_A^*)$ to B to obtain B^* and compute the pairwise-complete correlation, $\text{cor}_{\text{full},A}$, between A and B^* . We repeat this in the other comparison direction to obtain $\text{cor}_{\text{full},B}$ and average the two:

$$\text{cor}_{\text{full}} = \frac{1}{2} (\text{cor}_{A,\text{full}} + \text{cor}_{B,\text{full}}).$$

We assume that the **full-scan pairwise-complete correlation** is large for truly matching cartridge cases.

4.3.2.3.2 Cell-Based Registration

We next perform a cell-based comparison procedure, which begins with selecting one of the matrices, say A , as the “source” matrix that is partitioned into a grid of cells. The left side of Figure 4.4 shows an example of such a cell grid overlaid on a scan. Each of these source cells will be compared to the “target” matrix, in this case B^* . Because A and B^* are already partially aligned from the full-scan registration procedure, we compare each source cell to B^* using a new rotation grid of $\Theta'_A = \{\theta_A^* - 2^\circ, \theta_A^* - 1^\circ, \theta_A^*, \theta_A^* + 1^\circ, \theta_A^* + 2^\circ\}$.

We now extend the surface matrix notation introduced previously to accommodate cells. Let A_t denote the t -th cell of matrix A , $t = 1, \dots, T_A$ where T_A is the total number of cells containing non-missing values in scan A (e.g., $T_A = 43$ in Figure 4.4) and let $(a_t)_{ij}$ denote the i, j -th element of A_t .

The cell-based comparison procedure is outlined in Cell-Based Comparison Algorithm.

Cell-Based Comparison Algorithm

Data: Source matrix A , target matrix B^* , grid size $R \times C$, and rotation grid Θ'_A

Result: Estimated translations and CCF_{\max} values per cell, per rotation

Partition A into a grid of $R \times C$ cells;

Discard cells containing only missing values, leaving T_A remaining cells;

```

for  $\theta \in \Theta'_A$  do
    Rotate  $B^*$  by  $\theta$  to obtain  $B_\theta^*$ ;
    for  $t = 1, \dots, T_A$  do
        Calculate  $CCF_{\max,A,t,\theta} = \max_{m,n} (a_t \star b_\theta^*)_{mn}$ ;
        Calculate translation  $[m_{A,t,\theta}^*, n_{A,t,\theta}^*] = \arg \max_{m,n} (a_t \star b_\theta^*)_{mn}$ ;
    end
    end
return  $\mathbf{F}_A = \{(m_{A,t,\theta}^*, n_{A,t,\theta}^*, CCF_{\max,A,t,\theta}, \theta) : \theta \in \Theta'_A, t = 1, \dots, T_A\}$ 

```

Rather than exclusively returning the registration that maximizes the overall CCF as in Image Registration Algorithm, Cell-Based Comparison Algorithm returns the set \mathbf{F}_A of translations and CCF values for each of the T_A cells and each rotation in Θ'_A .

Figure 4.4 shows the estimated registrations of cells between two non-match cartridge cases. We magnify the surface values captured by cell pairs 5, 1 and 7, 7 and note the similarities in the surface values; for example, the dark purple region in the middle of the cell 7, 7 pair.

Just as with the whole-scan registration, we calculate the pairwise-complete correlation between each cell A_t and a matrix $B_{\theta,t}^*$ of the same size extracted from B_θ^* after translating by $[m_{A,\theta}^*, n_{A,\theta}^*]$. From this we obtain a set of pairwise-complete correlations for each cell and rotation: $\{cor_{A,t,\theta} : t = 1, \dots, T_A, \theta \in \Theta'_A\}$.

We repeat Cell-Based Comparison Algorithm and the pairwise-complete correlation calculation using B as the source scan and A^* as the target, resulting in cell-based registration set \mathbf{F}_B and pairwise-complete correlations $\{cor_{B,t,\theta} : t = 1, \dots, T_B, \theta \in \Theta'_B\}$.

For $d = A, B$ and $t = 1, \dots, T_d$, define the cell-wise maximum pairwise-complete correlation as:

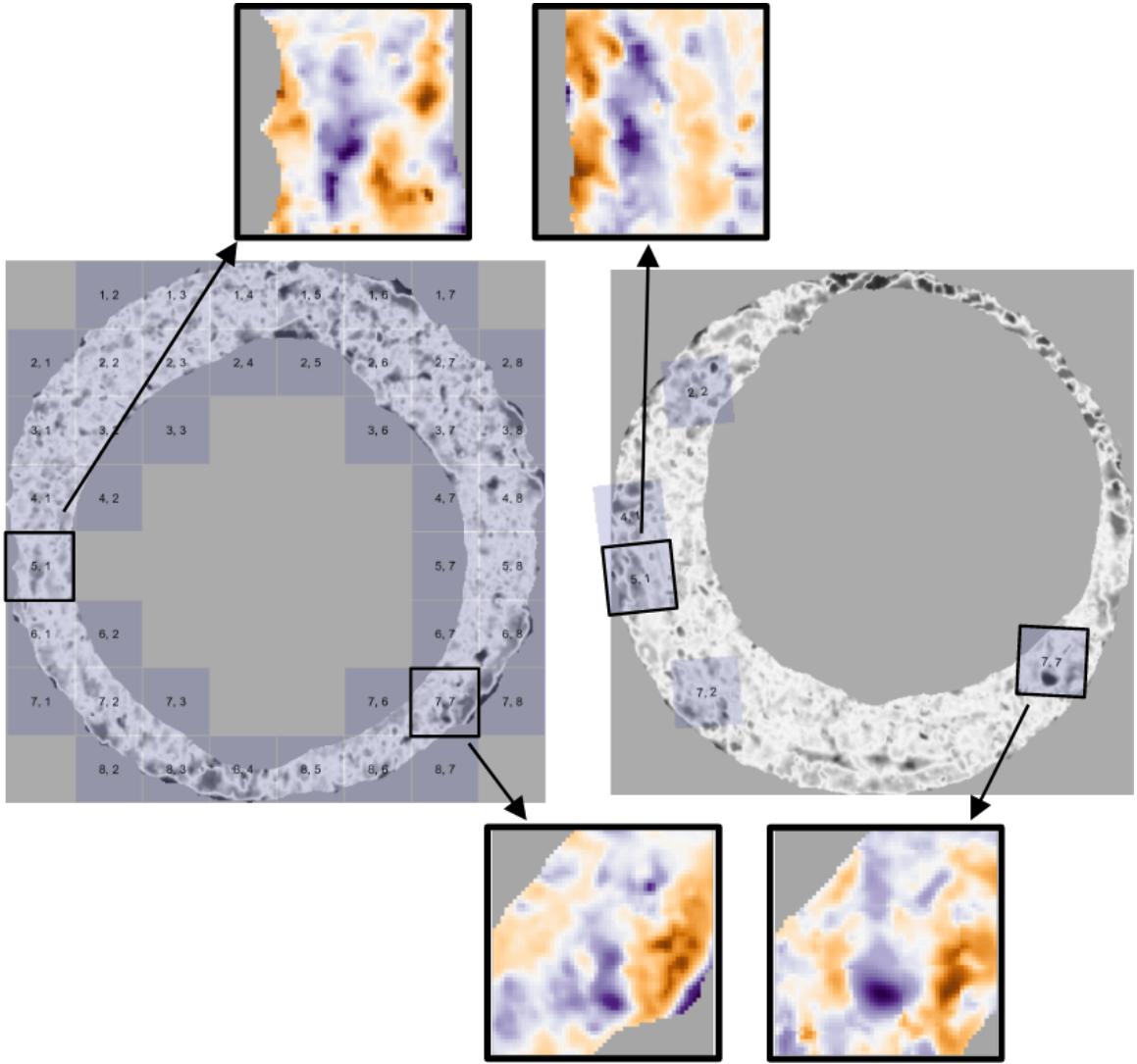


Figure 4.4: Estimated registrations of cells from a non-match pair of cartridge cases. A source scan (left) is separated into an 8×8 grid of cells. We exclude cells containing only missing values (visualized here as gray pixels). Each source cell is compared to a target scan (right) to estimate where it aligns best. We show a handful of cells at their estimated alignment in the target scan and magnify the surfaces captured by cell pairs 5, 1 and 7, 7. Although the cartridge case pair is non-matching, we note that there are similarities in the surface markings for these cell pairs.

$$cor_{d,t} = \max_{\theta} \{ cor_{d,t,\theta} : \theta \in \Theta'_d \}$$

We compute two features, the **average** and **standard deviation of the cell-based pairwise-complete correlations**, using the correlation data:

$$\bar{cor}_{\text{cell}} = \frac{1}{T_A + T_B} \sum_{d \in \{A,B\}} \sum_{t=1}^{T_d} cor_{d,t}$$

$$s_{cor} = \sqrt{\frac{1}{T_A + T_B - 1} \sum_{d \in \{A,B\}} \sum_{t=1}^{T_d} (cor_{d,t} - \bar{cor}_{\text{cell}})^2}$$

We expect \bar{cor}_{cell} and s_{cor} to be large for truly matching cartridge case pairs relative to non-matching pairs.

For $d = A, B$ and $t = 1, \dots, T_d$, define the per-cell estimated translations and rotation as:

$$\theta_{d,t}^* = \arg \max_{\theta} \{ CCF_{\max,d,t,\theta} : \theta \in \Theta'_d \}$$

$$m_{d,t}^* = m_{\theta_{d,t}^*, d, t}^*$$

$$n_{d,t}^* = n_{\theta_{d,t}^*, d, t}^*$$

We compute the **standard deviation of the cell-based estimated registrations** using the estimated translations and rotations:

$$s_{\theta^*} = \sqrt{\frac{1}{T_A + T_B - 1} \sum_{d \in \{A,B\}} \sum_{t=1}^{T_d} (\theta_{d,t}^* - \bar{\theta}^*)^2}$$

$$s_{m^*} = \sqrt{\frac{1}{T_A + T_B - 1} \sum_{d \in \{A,B\}} \sum_{t=1}^{T_d} (m_{d,t}^* - \bar{m}^*)^2}$$

$$s_{n^*} = \sqrt{\frac{1}{T_A + T_B - 1} \sum_{d \in \{A,B\}} \sum_{t=1}^{T_d} (n_{d,t}^* - \bar{n}^*)^2}$$

where

$$\begin{aligned}\bar{m}^* &= \frac{1}{T_A + T_B} \sum_{d \in \{A, B\}} \sum_{t=1}^{T_d} m_{d,t}^* \\ \bar{n}^* &= \frac{1}{T_A + T_B} \sum_{d \in \{A, B\}} \sum_{t=1}^{T_d} n_{d,t}^* \\ \bar{\theta}^* &= \frac{1}{T_A + T_B} \sum_{d \in \{A, B\}} \sum_{t=1}^{T_d} \theta_{d,t}^*. \end{aligned}$$

We expect $s_{\theta^*}, s_{m^*}, s_{n^*}$ to be small for truly matching cartridge case pairs relative to non-matching pairs.

From the full-scan and cell-based registration procedures, we obtain six features summarized in Table 4.1.

cor_{full}	Full-scan pairwise-complete correlation
\overline{cor}_{cell}	Average cell-based pairwise-complete correlation
s_{cor}	Standard deviation of the cell-based pairwise-complete correlations
s_{m^*}	Standard deviation of the cell-based vertical translations (in microns)
s_{n^*}	Standard deviation of the cell-based horizontal translations (in microns)
s_{θ^*}	Standard deviation of the cell-based rotations (in degrees)

Table 4.1: Six similarity features based on registering full scans and cells.

4.3.2.4 Density-Based Features

We wish to identify when multiple cells agree on, or cluster around, a particular registration value. However, pursuant with the notion that only certain regions of matching

cartridge cases contain distinctive markings, it is unreasonable to assume and empirically rare that **all** cells agree on a single registration. In fact, it is common for many cells to disagree on a registration. For example, the left scatterplot in Figure 4.5 shows the per-cell estimated translations $[m_{A,t,\theta}^*, n_{A,t,\theta}^*]$ when scan A is used as source and B^* as target rotated by $\theta = 3^\circ$. The right scatterplot shows the per-cell estimated translations with the roles of A and B^* reversed for $\theta = -3^\circ$. We see distinctive clusters, the black points, in both plots among many noisy, gray points. The task is to isolate the clusters amongst such noise.

We use the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm proposed by Ester et al. (1996) to identify clusters. Compared to other clustering algorithms such as k-means (MacQueen, 1967), DBSCAN does not require a pre-defined number of expected clusters. Instead, the algorithm forms clusters if the number of points within an $\epsilon > 0$ distance of a point exceeds some pre-defined threshold, $minPts > 1$. If a point does not belong to a cluster, then DBSCAN labels that point as “noise.” In Figure 4.5, we use DBSCAN with $\epsilon = 5$ and $minPts = 5$ to identify clusters of size 14 and 13, respectively, visualized as black points. These cluster sizes suggest that the scans match. Additionally, the mean cluster centers are approximately opposites of each other: $(\hat{m}_A, \hat{n}_A, \hat{\theta}_A) \approx (16.9, -16.7, 3^\circ)$ when A is used as source compared to $(\hat{m}_B, \hat{n}_B, \hat{\theta}_B) \approx (-16.2, 16.8, -3^\circ)$ when B^* is used as source. This provides further evidence of a match.

To calculate the density-based features, we first use a 2D kernel density estimator (Venables and Ripley, 2002) to identify the rotation $\hat{\theta}_d$ at which the per-cell translations achieve the highest density. Next, we compute clusters using the DBSCAN algorithm amongst the estimated translations $\{(m_{d,t,\hat{\theta}_d}^*, n_{d,t,\hat{\theta}_d}^*): t = 1, \dots, T_d\}$ like those shown in Figure 4.5.³ Let \mathbf{C}_d denote the set of cells in the DBSCAN cluster. We treat the mean cluster centers as the estimated translations $[\hat{m}_d, \hat{n}_d]$.

³If more than one cluster is identified, we binarize the points based on whether they were assigned to any cluster or if they are a noise point and proceed as if there is only one cluster. We assume that two or more clusters form only because of the coarse rotation grid considered. Were a finer grid used, the points would coalesce into a single cluster around the true translation value. This assumption has empirical support through our experimentation.

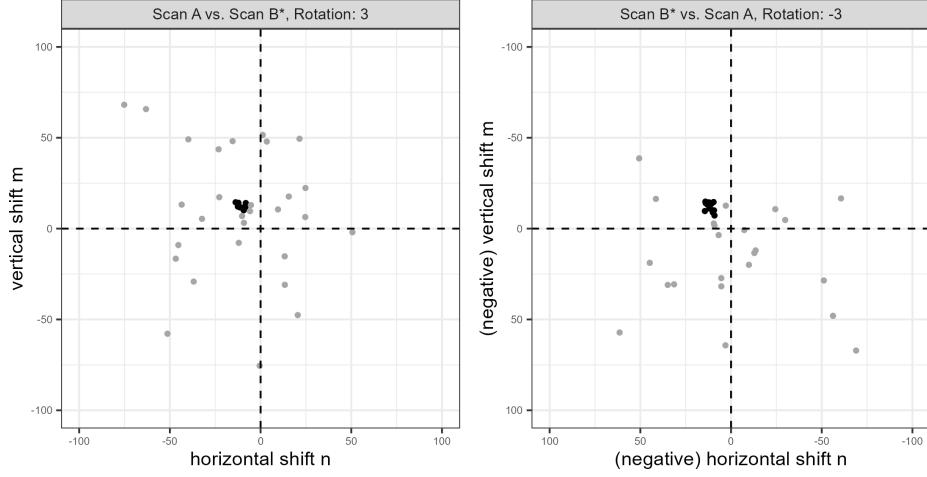


Figure 4.5: Cluster assignments based on the Density Based Spatial Clustering with Applications to Noise (DBSCAN) algorithm for estimated translations in two comparison directions. Using scan A as source results in a cluster of size 14 (left) compared to 13 when scan B^* is used as source (right). Noting the reversed axes in the right plot, we see that the clusters are located approximately opposite of each other. Points are jittered for visibility.

We calculate four features from the density-based clustering procedure: **average DBSCAN cluster size C** , the **DBSCAN cluster indicator C_0** , and the **root sum of squares of the density-estimated registrations $(\Delta_\theta, \Delta_{\text{trans}})$** defined as:

$$C = \frac{1}{2} (|\mathbf{C}_A| + |\mathbf{C}_B|)$$

$$C_0 = I(|\mathbf{C}_A| > 0 \text{ and } |\mathbf{C}_B| > 0)$$

$$\Delta_\theta = |\hat{\theta}_A + \hat{\theta}_B|$$

$$\Delta_{\text{trans}} = \sqrt{(\hat{m}_A + \hat{m}_B)^2 + (\hat{n}_A + \hat{n}_B)^2}$$

where $|\mathbf{C}_d|$ denotes the cardinality of \mathbf{C}_d and $I(\cdot)$ is the identity function equal to 1 if the predicate argument “.” evaluates to TRUE and 0 otherwise. We use both C and C_0 because of potential missingness in the values of C if no cluster is identified. Missing C values are imputed using the median non-missing value when fitting classifiers, so the missingness information is retained in C_0 .

For truly matching cartridge case pairs, we expect C to be large, C_0 to be 1, and $\Delta_\theta, \Delta_{\text{trans}}$ to be small relative to non-matching pairs. We obtain four density-based features summarized in Table 4.2.

C	Average DBSCAN cluster size
C_0	DBSCAN cluster indicator
Δ_θ	Absolute sum of the density-estimated rotations (in degrees)
Δ_{trans}	Root sum of squares of the cluster-estimated translations (in microns)

Table 4.2: Four similarity features based on the density-based clustering procedure.

4.3.2.5 Visual Diagnostic Features

The final set of features we calculate are based on visual diagnostic tools described in Chapter 3 [CITE]. These numerical features quantify the qualitative observations one can make from the diagnostics.

To create the visual diagnostics, we perform element-wise matrix operations. For a matrix $X \in \mathbb{R}^{k \times k}$ and Boolean-valued condition matrix $cond : \mathbb{R}^{k \times k} \rightarrow \{\text{TRUE}, \text{FALSE}\}^{k \times k}$, we define an element-wise filter operation $\mathcal{F} : \mathbb{R}^{k \times k} \rightarrow \mathbb{R}^{k \times k}$ as:

$$\mathcal{F}_{cond}(X) = (f_{ij})_{1 \leq i,j \leq k} = \begin{cases} x_{ij} & \text{if } cond \text{ is TRUE for element } i, j \\ NA & \text{otherwise} \end{cases}$$

Of particular interest in our application is the (absolute) difference between surface matrices. For example, $\mathcal{F}_{|A-B^*|>\tau}(A)$ contains elements of matrix A where the pair of scans A and B^* deviate by at least $\tau > 0$. Surface values in A and B^* that are “close,” meaning within τ distance, to each other are replaced with NA in this filtered matrix.

First, we calculate the correlation $cor_{d,\text{full,diff}}$ between the filtered matrices $\mathcal{F}_{|A-B^*|>\tau}(A)$ and $\mathcal{F}_{|A-B^*|>\tau}(B^*)$ for $d = A$ and $\mathcal{F}_{|A^*-B|>\tau}(A^*)$ and $\mathcal{F}_{|A^*-B|>\tau}(B)$ for $d = B$. We use the average **full-scan differences correlation** as a feature:

$$cor_{\text{full,diff}} = \frac{1}{2} (cor_{A,\text{full,diff}} + cor_{B,\text{full,diff}}).$$

We assume that $cor_{\text{full,diff}}$ will be large for matching cartridge case pairs relative to non-matching pairs. Said another way, we assume that regions of matching cartridge cases that are different will still follow similar trends. This can occur due to variability in the amount of contact between a cartridge case and breech face across multiple fires of a single firearm. We calculate the correlation by vectorizing the two filtered surface matrices and treating missing values by case-wise deletion.

As before, we extend our notation to accommodate cell comparisons $t = 1, \dots, T_d$ for $d = A, B$ using subscripts: $cor_{d,t,\text{diff}}$. For example, $cor_{A,t,\text{diff}}$ is the correlation between cell filtered surface matrices $\mathcal{F}_{|A_t-B_{t,\theta_t^*}|>\tau}(A_t)$ and $\mathcal{F}_{|A_t-B_{t,\theta_t^*}|>\tau}(B_{t,\theta_t^*})$ where B_{t,θ_t^*} is the matrix extracted from B^* that maximizes the CCF with A_t . We calculate the **average cell-based differences correlation** across all cells and both directions:

$$\overline{cor}_{\text{cell,diff}} = \frac{1}{T_A + T_B} \sum_{d \in \{A,B\}} \sum_{t=1}^{T_d} cor_{d,t,\text{diff}}$$

Next, we consider features based on the elements of the Boolean $cond$ matrix. Consider Figure 4.6 that shows the filtered element-wise average $\mathcal{F}_{|A-B^*|<\tau}(\frac{1}{2}(A+B^*))$ on the left and the associated $cond$ matrix $|A-B^*| > \tau$ visualized in black-and-white in the middle with filtered elements, whose $cond$ value is *TRUE*, shown in white.

We first calculate the ratio between such a $cond$ matrix and its complement. For $d = A$, we consider the $cond$ matrices $|A-B^*| \leq \tau$ and $|A-B^*| > \tau$. The ratio is given by

$$r_d = \frac{\mathbf{1}^T I(|A-B^*| \leq \tau) \mathbf{1}}{\mathbf{1}^T I(|A-B^*| > \tau) \mathbf{1}}$$

where $\mathbf{1} \in \mathbb{R}^k$ is a column vector of ones and $I(\cdot)$ is the element-wise, matrix-valued indicator function. We consider the average **full-scan similarities vs. differences ratio** across the two comparison directions:

$$r_{\text{full}} = \frac{1}{2}(r_A + r_B).$$

We expect r_{full} to be large for matching pairs compared to non-matching pairs. That is, truly matching pairs will have more similarities than differences.

We also calculate features based on the ratio for cell comparisons $t = 1, \dots, T_d$, $d = A, B$. Let $r_{d,t}$ denote the ratio for cell comparison t in direction d . We consider the **average and standard deviation of the cell-based similarities vs. differences ratio**:

$$\bar{r}_{\text{cell}} = \frac{1}{T_A + T_B} \sum_{d \in \{A, B\}} \sum_{t=1}^{T_d} r_{d,t}$$

$$s_{\text{cell},r} = \sqrt{\frac{1}{T_A + T_B - 1} \sum_{d \in \{A, B\}} \sum_{t=1}^{T_d} (r_{d,t} - \bar{r}_{\text{cell}})^2}.$$

We expect \bar{r}_{cell} and $s_{\text{cell},r}$ to be large for matching cartridge case pairs relative to non-match pairs.

Another aspect of the *cond* matrix we consider is the size of the individual filtered regions. For two matching cartridge cases, we expect that there are few differences compared to similarities *and* that the different regions are relatively small. We use a connected components labeling algorithm detailed in Hesselink et al. (2001) to identify individual “neighborhoods” of filtered elements (Barthelme, 2019). More precisely, the algorithm returns a set of sets $\mathbf{S}_d = \{S_{d,1}, S_{d,2}, \dots, S_{d,L_d}\}$ where each $S_{d,l}$ is a set of indices of the *cond* matrix that have a value of *TRUE* and are connected by a chained-together sequence of 4 (Rook’s) neighborhoods. The right side of Figure 4.6 shows each $S_{d,l}$ distinguished by different fill colors, $l = 1, \dots, L_d$.

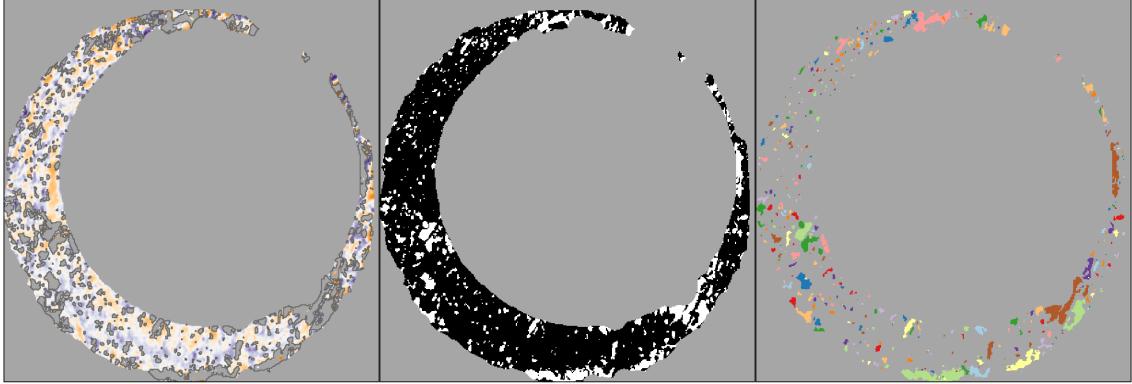


Figure 4.6: (Left) After aligning two scans, we filter regions that are "different" from each other, meaning the absolute difference between surface values is larger than some threshold. (Middle) We binarize the scan into "filtered" or "non-filtered" regions - shown in white and black, respectively. (Right) Using a connected components labeling algorithm, we identify connected "neighborhoods" of filtered elements. We assume that these neighborhoods will be small, on average, if comparing truly matching cartridge cases.

We calculate the following features using the full-scan labeled neighborhoods:

$$\overline{|S|}_{\text{full}} = \frac{1}{L_A + L_B} \sum_{d \in \{A,B\}} \sum_{l=1}^{L_d} |S_{d,l}|$$

$$s_{\text{full},|S|} = \sqrt{\frac{1}{L_A + L_B - 1} \sum_{d \in \{A,B\}} \sum_{l=1}^{L_d} (|S_{d,l}| - \overline{|S|}_{\text{full}})^2}$$

where $|S_{d,l}|$ is the cardinality of $S_{d,l}$. We assume that the **average** and **standard deviation of the full-scan neighborhood sizes** will be small for matching cartridge case pairs relative to non-matching pairs. That is to say, we assume that the the regions of A and B that are different will all be small, on average, and vary little in size. This assumption is appropriate assuming that the breech face leaves consistent markings on fired cartridge cases.

Again, we extend our notation to accommodate individual cells. Let $\mathcal{S}_{d,t} = \{S_{d,t,1}, \dots, S_{d,t,L_{d,t}}\}$ denote the set of labeled neighborhoods for a cell $t = 1, \dots, T_d$, $d = A, B$.

We calculate the per-cell average and standard deviation of the labeled neighborhood cell size:

$$\begin{aligned}\overline{|S|}_{d,t} &= \frac{1}{L_{d,t}} \sum_{l=1}^L |S_{d,t,l}| \\ s_{d,t,|S|} &= \sqrt{\frac{1}{L_{d,t}-1} \sum_{l=1}^{L_{d,t}} (|S_{d,t,l}| - \overline{|S|}_{\text{cell},d,t})^2}.\end{aligned}$$

We assume that the cell-based $\overline{|S|}_{d,t}$ and $s_{d,t,|S|}$ will be small, on average, for truly matching cartridge cases. Consequently, we use the sample average of these as features:

$$\begin{aligned}\overline{|S|}_{\text{cell}} &= \frac{1}{T_A + T_B} \sum_{d \in \{A,B\}} \sum_{t=1}^{T_d} \overline{|S|}_{d,t} \\ \bar{s}_{\text{cell},|S|} &= \frac{1}{T_A + T_B} \sum_{d \in \{A,B\}} \sum_{t=1}^{T_d} s_{d,t,|S|}\end{aligned}$$

We assume that the **average cell-wise neighborhood size** and the **average standard deviation of the cell-wise neighborhood sizes** will be small for matching cartridge case pairs relative to non-match pairs.

Table 4.3 summarizes the nine features based on visual diagnostics. This concludes our explanation of the ACES feature set. Next, we use the 19 ACES features to train and test classifier models.

4.3.3 Scoring

We use a data set of 510 cartridge cases fired from 25 firearms. We randomly split the data into 10 firearms for training and 15 firearms for testing. This resulted in a training data set of 210 cartridge cases, $\binom{210}{2} = 21,945$ pairwise comparisons, and a testing set of 300 cartridge cases, $\binom{300}{2} = 44,850$ pairwise comparisons. Because we consider every

$cor_{full,diff}$	Full-scan differences correlation
$\overline{cor}_{cell,diff}$	Average cell-wise differences correlation
r_{full}	Full-scan similarities vs. differences ratio
\bar{r}_{cell}	Average cell-based similarities vs. differences ratio
$s_{cell,r}$	Standard deviation of the cell-based similarities vs. differences ratio
$\overline{ S }_{full}$	Average full-scan neighborhood size
$s_{full, S }$	Standard deviation of the full-scan neighborhood sizes
$\overline{ S }_{cell}$	Average cell-wise neighborhood sizes
$\bar{s}_{cell, S }$	Average standard deviation of the cell-wise neighborhood sizes

Table 4.3: Nine similarity features calculated based on visual diagnostics.

pairwise comparison between these scans, there is a relatively large class imbalance between matches and non-matches in these data sets. Specifically, non-matching comparisons make up 19,756 of the 21,945 (90.0%) training comparisons and 41,769 of the 44,850 (93.1%) testing comparisons.

We use 10-fold cross-validation repeated thrice (Kuhn, 2022) to train three binary classifiers based on a logistic regression, a classification tree (Therneau and Atkinson, 2022), and a random forest [Breiman (2001);randomForest]. Each of these models predicts the probability that a pair of cartridge cases match. Then, the model classifies the pair as a match or non-match depending on whether the match probability exceeds a set threshold. On top of the tunable parameters of each model, we treat this threshold as a parameter to be optimized.

We consider different optimization criteria for training each classifier. One criterion prescribes selecting the model that maximizes the overall classification accuracy, or the pro-

portion of correct classifications, on the training data. For this criterion, we select the match probability threshold that maximizes the overall classification accuracy. However, models that are trained to maximize accuracy on imbalanced data often exhibit a “preference” for classifying new observations as the majority class (Fernández et al., 2018), which in our case are non-matches. We will see in the next section that models trained to maximize accuracy on our training data set indeed exhibit this behavior.

An alternative optimization criterion commonly used for imbalanced data is to select the model that maximizes the area under the Receiver Operating Characteristic (ROC) curve, which measures the performance of a model under different threshold values (James et al., 2013). The model that maximizes this area, commonly abbreviated AUC, is one that performs best under a variety of threshold values relative to the other models - this consistency is a desired trait. Using the ROC curve, we choose the match probability threshold that balances the true positive and true negative rates on the training data.

Once we have a trained model, we use it to predict the match probability and classify a new cartridge case pair. We do so for the pairwise comparisons in the test data as a means of comparing the generalizability of the various models. The following section details the results of this cross-validation training/testing procedure.

4.4 Results

Figure 4.7 summarizes the training and testing accuracy, true negative and true positive rates for 13 binary classifiers. We distinguish between the training and testing results using gray and black points/line segments, respectively, which allows us to assess the generalizability of the various models. The conclusions drawn from Figure 4.7 are intended to primarily be qualitative and comparative across models. Tables 4.6 and 4.7 in the Appendix provide a numerical summary of these results.

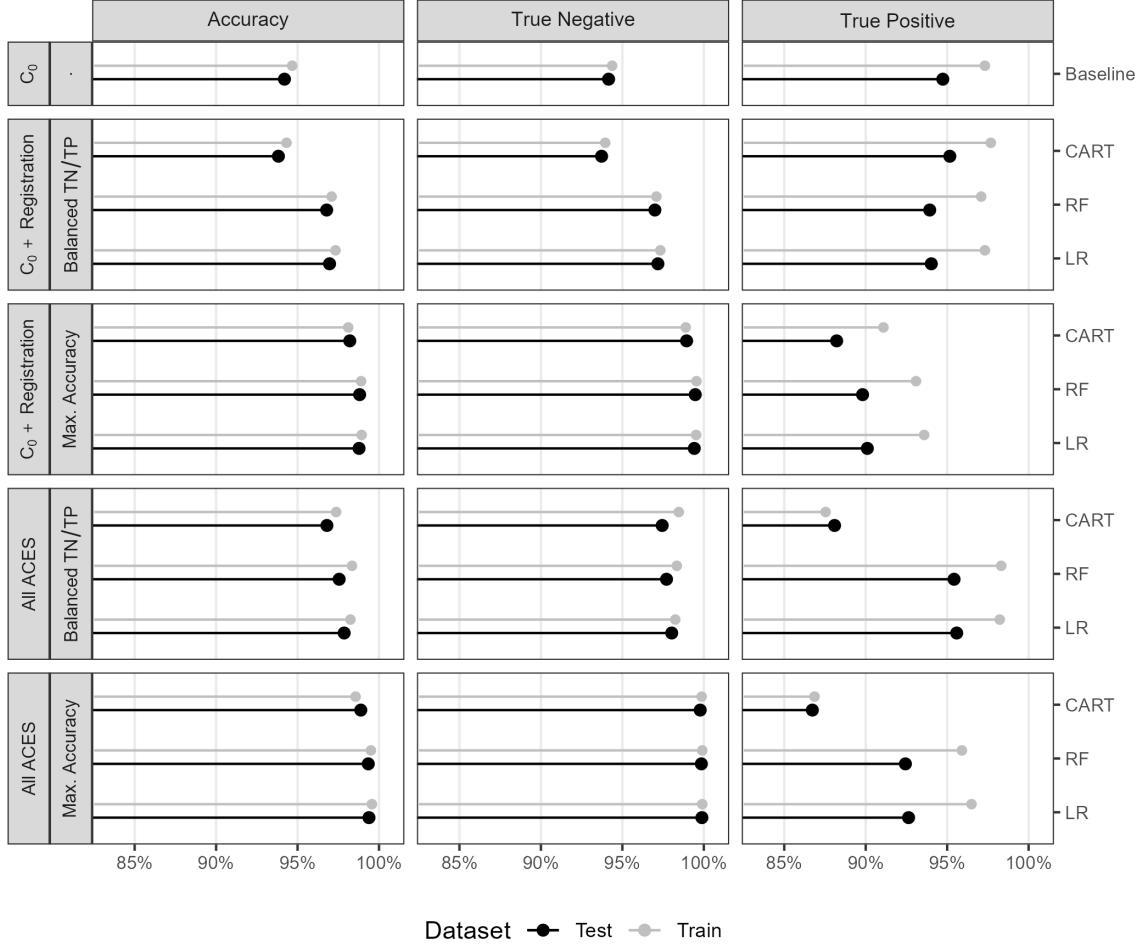


Figure 4.7: We summarize classification accuracy, true negative, and true positive rates for both the training and testing results, represented as gray and black points/lines respectively, for 13 binary classifier models. Our primary interest is the test data results, but visualizing the training data results allows us to assess the generalizability of the models after training. In the first row, we consider a classifier based on a single feature, the Cluster Indicator feature C_0 , as a baseline. The remaining rows show results from training/testing classifiers based on a decision tree (CART), random forest (RF), and logistic regression (LR) under various feature sets and optimization critieria. The second and third rows show results based on a subset of seven features from the ACES feature set while the fourth and fifth rows show results using all 19 ACES features. We also compare results from optimizing either the overall Accuracy (third and fifth rows) or the area under the Receiver Operating Characteristic curve (AUC, second and fourth rows) during training.

We first compare the training and testing results across the 13 models and three columns in Figure 4.7. In general, the true negative rates based on the test data are comparable, if slightly lower, to those of the training data. This indicates that the model's ability to distinguish between non-matching comparisons generalizes well to the testing data. In contrast, the true positive rates tend to be lower for the test data compared to the training data across the various models, which indicates a potential difference between the training and testing data. As we discuss below, there is a single firearm among the 15 test firearms that contributes the majority of false negative (misclassified match) test classifications. Despite lower true positive rates, the overall accuracy between the training and testing sets are comparable due to the large class imbalance between matching and non-matching comparisons in both.

In the first row, we consider a baseline classifier based solely on the Cluster Indicator feature C_0 . Namely, if the DBSCAN algorithm finds clusters in the cell-based translations from both directions of a cartridge case comparison, then that pair is classified as a match. This is analogous to the classification rule used in Zhang et al. (2020). The testing classification accuracy associated with this classifier is 94.20% with approximately balanced true negative and true positive rates of 94.73% and 94.16%, respectively.

The second and third rows of Figure 4.7 summarize results from training the three classifier models on a subset of the full ACES feature set consisting of the Cluster Indicator feature C_0 and the six registration-based features summarized in Table 4.1. As discussed in the next section, we consider this subset of features to represent the features used in Congruent Matching Cells methods [Song (2013);Zhang2020]. We distinguish the second and third rows by whether the models were selected to balance the overall true negative and true positive (TN/TP) rates or to maximize the overall accuracy. In general, we see that the logistic regression (LR) and random forest (RF) models perform comparable to each other in accuracy, true negative, and true positive rates. The decision tree (CART) model performs worse than the other two models, particularly for the balanced TN/TP

models in the second row - although it actually has the largest true positive rate in this case. Despite the fact that the models in the second row were selected based on balancing the training true negative and true positive rates, we note that these rates for the test data are not as well-balanced; namely, the true negative rates still tend to be larger than the true positive rates. This discrepancy is most notable in the third row, which illustrates how models selected to maximize overall accuracy on imbalanced data err on the side of the majority class. Below, we analyze the contribution of various test firearms towards the true positive rates to explore this discrepancy.

The fourth and fifth rows of Figure 4.7 summarize the classification results based on using all 19 ACES features. The conclusions made from comparing these two rows are qualitatively the same as those from comparing the second and third rows. Namely, the “maximum accuracy” models have a higher overall test accuracy compared to the “balanced TN/TP” models, albeit at the expense of worse test true positive rates. If we compare the “ $C_0 + \text{Registration}$ ”-trained models in the second and third vs. the “All ACES”-trained models in the fourth and fifth rows, we see that the addition of the other ACES features leads to improved test true negative and true positive rates (and consequently overall accuracy) with the largest gains observed in the true positive rates.

Across all 13 models, the All ACES-trained, maximum accuracy logistic regression model has the largest overall test accuracy of 99.56% and largest true negative rate of 99.90%. The All ACES-trained, balanced TN/TP logistic regression model has the largest test true positive rate of 98.32%. For both of these logistic regression models, the random forest model trained under the same conditions is a close second (see Table 4.7 for more details).

While it’s useful to consider the accuracy, true negative, and true positive rates to compare various models, forensic examiners would likely not use the binary classification returned by a model in casework. Instead, they would consider the match probability predicted by the model as a similarity score and incorporate it into their decision-making

process. As such, we also consider the distribution of the predicted match probabilities for matching and non-matching comparisons. Figure 4.8 shows a dot plot of the predicted match probabilities for the 41,769 non-match and 3,181 match comparisons in the test set. Specifically, these probabilities are predicted by the logistic regression model selected to maximize the overall classification accuracy based on the full ACES feature set. As we expect, few non-match comparisons have large match probabilities, which justifies the low false positive rate observed in Figure 4.7. However, there are an unexpected number of matching comparisons that also have a low match probability. Further, the density of the match probabilities is concentrated towards the interval's extremes, which may be a characteristic of the logistic regression model or of the underlying test data.

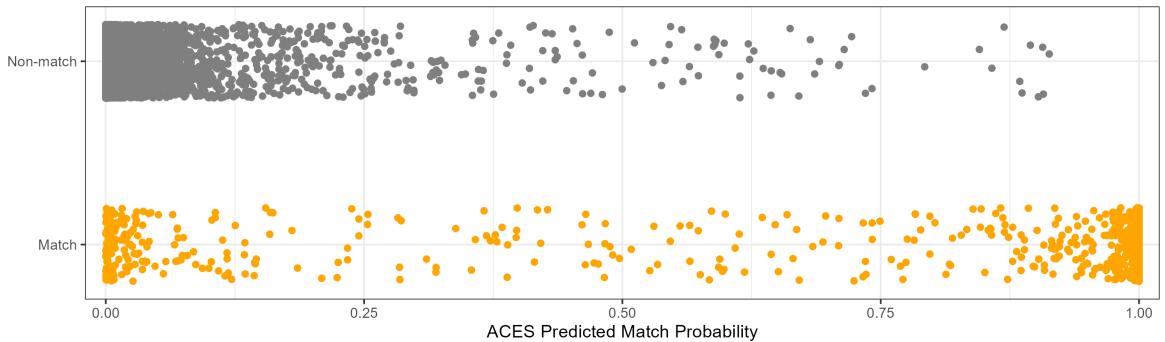


Figure 4.8: A dot plot of the predicted match probabilities for the non-match and match comparisons in the test set based on a logistic regression model. As we expect, the non-match comparisons tend to have a low match probability. However, we see that there are many matching comparisons that also have a low match probability.

The left side of Figure 4.9 shows the predicted match probabilities for the same matching test comparisons shown in Figure 4.8, but now separated by the 15 firearms comprising the test set. We see that the firearm T has far more matching comparisons with low match probabilities compared to the other 14 test firearms. This is further underscored by the right side of the Figure 4.9, which shows the ratio of misclassifications to total comparisons for every pair of test firearms based on the same logistic regression model used in Figure 4.8. The main diagonal shows the false negative misclassifications while the off-diagonal shows the false positives. For clarity, we use blank tiles for comparisons where 0 misclassifications

occurred. We see that the false negative rate for firearm T of 45.9% far outweighs the error rates from other firearm pairs. Further, the 161 false negative firearm T comparisons comprise about 71% of all of the 226 false negative test comparisons and 5% of the 3,181 matching test comparisons. In sum, the model performs distinctly worse at identifying matching comparisons from firearm T compared to the other firearms, which partially explains the lower test true positive rates noted in Figure 4.7. Upon visual inspection of the scans from firearm T, we noted a lack of consistent markings on their surfaces that is not shared with scans from the other test firearms.

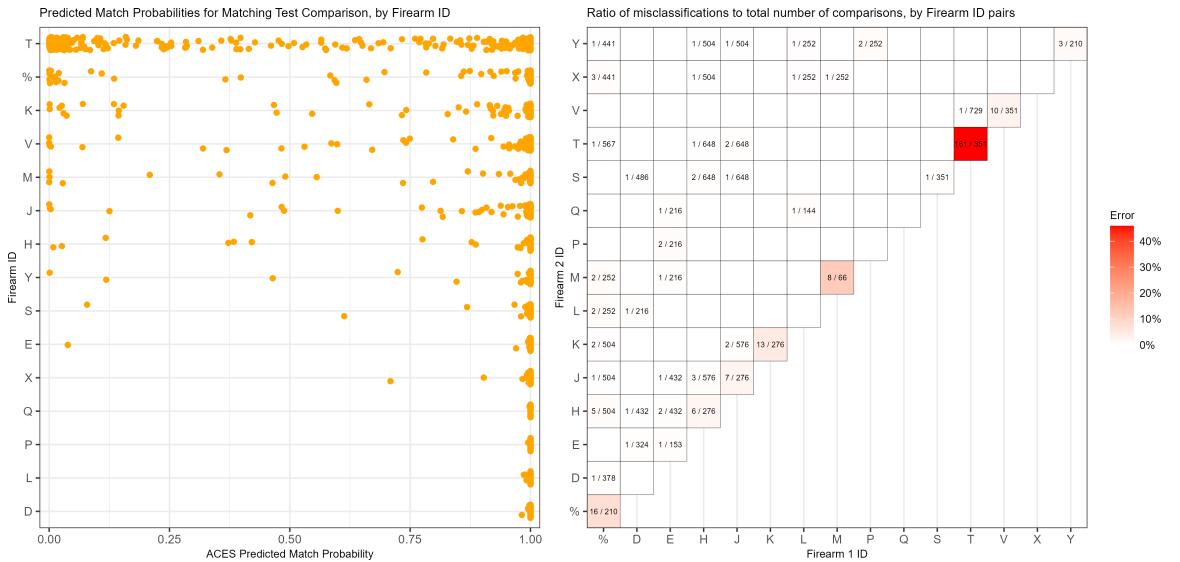


Figure 4.9: (Left) A dot plot of the predicted match probabilities for the match comparisons in the test set based on a logistic regression model, separated by firearm. We see that firearm T has more matching comparisons with low match probabilities than the other test firearms. (Right) Misclassifications divided by total number of pairwise comparisons for each pair of test firearms based on the same logistic regression model. For clarity, we do not show comparisons with 0 misclassifications. We note that the proportion of misclassified matching comparisons from firearm T of 45.9% is much higher than that of other comparisons.

Finally, we consider the relative importance of the 19 ACES features by fitting 10 replicate random forests using the full ACES feature set with fixed random seeds. For each replicate, we measure a variable's importance using the Gini Index, which measures the probability of making a misclassification for a given model (Hastie et al., 2001). A larger

decrease in the Gini Index corresponds with higher importance. Figure 4.10 shows the distribution of the mean Gini Index decrease for the 19 ACES features. Noting the log scale on which these points are plotted, we see that the cell-based similarities vs. differences ratio \bar{r}_{cell} is the most important variable followed by the cluster size C and the cell-based correlation \overline{cor}_{cell} . Interestingly, we note that these top three features consist of one feature from each of the three feature groups defined in the Methods section.

4.5 Discussion

We use a C_0 -based classifier as a baseline because it is analogous to the classification rule proposed in Zhang et al. (2020). Similarly, the cell-based registration features are based on the same cell-based comparison procedure used in Song (2013) and summarized in ???. Together, C_0 and the registration-based features can viewed as a fusion of previously-proposed cartridge case comparison algorithms. This is why we consider separate classifiers based on these features for the training and testing results shown in Figure 4.9. Table 4.4 summarizes the similarities between the ACES algorithm and the algorithms proposed in Zhang et al. (2020) and Song (2013). Another key difference between ACES and both of the previous algorithms is the training/testing procedure used to optimize and validate model parameters.

Table 4.5 shows the test classification error rates of the Congruent Matching Cells (CMC) algorithm proposed in Song (2013), the C_0 -based classifier like the one proposed in Zhang et al. (2020), and the two ACES logistic regression models selected to balance the true negative and true positive rates and maximize the classification accuracy. We obtained the CMC results by applying the implementation available in the cmcR R package (Zemmels et al., 2022a) on the same test data set used in the Results section. We used the optimization procedure described in Zemmels et al. (2023) to select CMC parameters. The C_0 -based error rates are the same as those shown in the first row of Figure 4.7. The ACES



Figure 4.10: Variable importance measures from fitting a random forest to the training data set, repeated 10 times under various random seeds. The top three features consist of one feature from each of the three feature groups defined in the Methods section. Points are plotted on a log scale and vertically jittered for visibility.

Original Paper	Similarities to ACES	Original Use	ACES Use
Song (2013)	Use		
	refig:cellComparison to estimate cell-based registrations	All cells Congruent Matching Cells if their registrations are close to a reference value. Classify a cartridge case pair as a match if the CMC count is at least 6.	Compute six summative features based on full-scan and cell registrations. Use features in a classifier model.
Zhang et al. (2020)	Use DBSCAN algorithm to identify cells that reach a consensus registration	Classify a cartridge case pair as a match if a DBSCAN cluster is identified.	Compute four numerical features based on DBSCAN clusters across both comparison directions. Use features in a classifier model.

Table 4.4: Comparison of the ACES algorithm to previous work. Although ACES shares similarities to previously-proposed algorithms, it includes additional nuance by computing features across both comparison directions and using these features in a classifier model.

logistic regression models perform better than the other classifiers on this test data set, most notably when compared to the CMC method in identifying matching cartridge case pairs. Interestingly, the C_0 -based classifier has a lower false negative error rate compared to the All ACES-trained, maximum accuracy logistic regression model, although it has a much higher false positive rate.

Classification Method	Error (%)	False Negative (%)	False Positive (%)
CMC method	2.6	33.8	0.3
Only C_0 feature	5.8	5.3	5.8
ACES LR, Balanced TN/TP	2.1	4.4	2.0
ACES LR, Max. Accuracy	0.6	7.4	0.1

Table 4.5: Testing classification error, false positive, and false negative rates for four types of classifier models. The CMC method results are derived from the implementation available in Zemmels et al. (2022a). The "Only C_0 feature" classifier is analogous to the classification rule used in Zhang et al. (2020). The last two rows show results from the Logistic Regression classifier trained on the all 19 ACES features under two optimization criteria - balancing the true negative and true positive rates or maximizing the overall classification accuracy on the training data set.

Both the registration and density-based features aim to measure similarities between two cartridge case surfaces. These features embody the notion that matching cartridge cases should have similar markings, so their cell-based correlations should be large and estimated registrations should agree. However, Figure 4.4 demonstrates that even non-matching cartridge case pairs may share similar markings. We are bound to find similarities if that is all we look for, so it is important to also consider dissimilarities. The visual diagnostic features accomplish this by partitioning scans into similar and different regions. The similarities vs. differences ratio and labeled neighborhood size features measure how extreme the differences are between two scans while the differences correlation features determine whether there are similarities among the different regions. This direct comparison

of the surface values aligns with the Theory of Identification which says that an examination should involve the comparison of the “relative height or depth, width, curvature and spatial relationship” of cartridge case impressions (AFTE Criteria for Identification Committee, 1992). Comparison algorithms like ACES will inevitably be used to augment the opinion of a forensic examiner, who may need to present algorithmic results to judges or juries as part of their expert testimony. As such, it is important that forensic examiners are able to interpret and explain the results of a comparison algorithm. The visual diagnostic features are useful for explaining the behavior of the algorithm in a manner that aligns with more traditional identification theory.

Our intention in fitting three classification models using different feature sets and optimization criteria was to explore each model’s strengths and weaknesses. A critical step in putting the ACES algorithm into practice will be to settle on a single model. Pragmatically, it seems reasonable to choose the model with the highest estimated accuracy on available test data. This turns out to also be the model with the smallest false positive rate, the All ACES-trained, maximum accuracy logistic regression model. Ethically, we might favor this model since misclassifying a truly non-matching cartridge case pair may incriminate an innocent individual. However, we see that using other optimization criteria, such as balancing the true negative and true positive rates, leads to different outcomes. Perhaps there is a happier medium between these two optimization criteria that leads to a model with a relatively small false positive error rate, yet We believe that this is ultimately a decision to be made with the input of others in forensic and legal communities. In our case, the ACES logistic regression model yields both the highest accuracy and lowest false positive rate on the test set, but this may not generalize to other cartridge cases. There is precedent for random forest models to be robust classifiers for forensic evidence (Hare et al., 2017; Park and Tyner, 2019). We need to apply the ACES algorithm to other types of evidence, such as using different ammunition or firearm makes/models, to better understand the algorithm’s strengths and weaknesses.

Another aspect to consider when choosing a model is interpretability and explainability. If an algorithm is applied in forensic casework, then evidentiary conclusions derived from the algorithm’s output will inevitably be presented to a non-expert judge or jury. More interpretable models are easier to understand, and therefore should be preferred. The classification behavior of the logistic regression and classification tree models are arguably easier to explain than the random forest model. For example, the logistic regression model parameters can be understood in terms of the estimated increase in odds of a match. Paired with its comparable performance to the random forest, we suggest using the logistic regression model with all 19 ACES features as the new standard to which other proposed similarity scoring algorithms are compared. However, we emphasize that the ultimate decision of which algorithm will be put into practice needs to be made with the input of the wider forensic and legal communities.

4.6 Conclusion

In this paper, we introduced the Automatic Cartridge Evidence Scoring (ACES) algorithm to measure the similarity between two fired cartridge cases based on their breech face impressions. In particular, we defined a set of 19 similarity features and used these features to train and test classifier models. We validated our algorithm on a set of 510 cartridge cases - the largest validation study of a cartridge case similarity scoring algorithm to-date. Compared to predominant algorithms like the CMC algorithm described in Song (2013), the ACES algorithm reduces the classification error rate by over 75% on our test data set. We propose a logistic regression classifier trained on the ACES feature set as a new benchmark to which future proposed methods are compared. We believe that the ACES algorithm, which is not designed to return “inconclusive” classifications, complements the results from the

Before the ACES algorithm can be put into practice, we must devise new stress-tests, using new ammunition and firearm combinations, to assess its robustness. A variety of factors, such as make/model and wear, may affect the discriminative power of the 19 features defined in this paper. As such, we view the current version of the ACES algorithm as more a foundation for future improvements than a final answer. We expect the ACES feature set to evolve over time; for discriminatory features to replace less informative features. Given the gravity of the application, we stress interpretability as a guiding principle for future feature engineering and model selection. A misunderstood feature or result may lead a lay judge or juror to an incorrect conclusion. Additionally, we urge future researchers to use a train/test procedure similar to the one outlined in this paper to validate proposed methods.

We developed the [scored](#) R package as an open-source companion to this paper. The code and data used in this paper are available at [link].

Computational Details

If necessary or useful, information about certain computational details such as version numbers, operating systems, or compilers could be included in an unnumbered section. Also, auxiliary packages (say, for visualizations, maps, tables, ...) that are not cited in the main text can be credited here.

The results in this paper were obtained using R 4.2.2. (R Core Team, 2017) R itself and all packages used are available from the Comprehensive R Archive Network (CRAN) at <https://CRAN.R-project.org/>.

We greatly appreciate the constructive feedback from the two anonymous reviewers. Special thanks also to all the developers and open-source contributors of R, knitr (Xie, 2015, 2014b), rticles (Allaire et al., 2021), and the tidyverse (Wickham et al., 2019), without whom this project would not have been possible.

Acknowledgments

This work was partially funded by the Center for Statistics and Applications in Forensic Evidence (CSAFE) through Cooperative Agreement 70NANB20H019 between NIST and Iowa State University, which includes activities carried out at Carnegie Mellon University, Duke University, University of California Irvine, University of Virginia, West Virginia University, University of Pennsylvania, Swarthmore College and University of Nebraska, Lincoln.

Appendix

A Registration Procedure Details

In our application, a registration is composed of a discrete translation by $(m, n) \in \mathbb{Z}^2$ and rotation by $\theta \in [-180^\circ, 180^\circ]$. Under this transformation, the index i, j maps to a new index i^*, j^* by:

$$\begin{pmatrix} j^* \\ i^* \end{pmatrix} = \begin{pmatrix} n \\ m \end{pmatrix} + \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} j \\ i \end{pmatrix}.$$

The value b_{ij} now occupies the index i^*, j^* . In practice, we use *nearest-neighbor interpolation* meaning i^* and j^* are rounded to the nearest integer.

To determine the optimal registration, we calculate the *cross-correlation function* (CCF) between A and B , which measures the similarity between A and B for every possible translation of B . Denoted $(A \star B)$, the CCF between A and B is a 2D array of dimension $2k-1 \times 2k-1$ with the m, n -th element given by:

$$(a \star b)_{mn} = \sum_{i=1}^k \sum_{j=1}^k a_{mn} \cdot b_{i+m, j+n}$$

where $1 \leq m, n \leq 2k - 1$. The value $(a \star b)_{mn}$ quantifies the similarity between A and B after B is translated m elements horizontally and n elements vertically. The CCF is often normalized between -1 and 1 for interpretability.

The above definition of the CCF is computationally taxing, particularly for large matrices. The Cross-Correlation Theorem provides an equivalent expression for the CCF:

$$(A \star B) = \mathcal{F}^{-1} \left(\overline{\mathcal{F}(A)} \odot \mathcal{F}(B) \right)$$

where \mathcal{F} and \mathcal{F}^{-1} are the discrete Fourier and inverse discrete Fourier transforms, respectively, $\overline{\mathcal{F}(A)}$ is the complex conjugate, and \odot is an element-wise (Hadamard) product (Brigham, 1988). We trade the moving sum computation from the previous CCF expression for two forward Fourier transforms, an element-wise product, and an inverse Fourier transform. The Fast Fourier Transform (FFT) algorithm reduces the computational load considerably [cite Tukey].

We estimate the registration by calculating the maximum CCF value across a range of rotations of matrix B . Let B_θ denote B rotated by an angle $\theta \in [-180^\circ, 180^\circ]$ and $b_{\theta mn}$ the m, n -th element of B_θ . Then the estimated registration (m^*, n^*, θ^*) is:

$$(m^*, n^*, \theta^*) = \arg \max_{m, n, \theta} (a \star b_\theta)_{mn}.$$

In practice we consider a discrete grid of rotations $\Theta \subset [-180^\circ, 180^\circ]$. The registration procedure is outlined in Image Registration Algorithm. We refer to the matrix that is rotated as the “target.” The result is the estimated registration of the target matrix to the “source” matrix.

The Fast Fourier Transform algorithm used in Image Registration Algorithm does not permit missing values in A or B . It is common for cartridge case scans to contain many missing values - the gray regions in Figure 4.3 represent structural values in the

scan. Thus, when calculating the CCF we impute these missing values with the average non-missing value in the scan. To measure the similarity between A and B while accounting for missingness, we calculate the correlation between the non-missing intersection of the aligned scans.

A.1 Cell-Based Registration Details

Following the full-scan registration, we next perform a cell-based registration procedure. Song (2013) points out that breech face impressions rarely appear uniformly on a cartridge case surface. Rather, distinguishing markings appear in specific, usually small, regions of a scan (the author refers to these as *valid correlation regions*). Calculating a correlation between two whole scans does not necessarily capture the similarity between these regions. Song (2013) proposes partitioning a scan into a rectangular grid of “cells” to isolate the valid correlation regions. Figure 4.4 shows an example of two non-match cartridge cases where the source matrix (left) is partitioned into an 8×8 grid of cells.

The cell-based comparison procedure begins with selecting one of the matrices, say A , as the “source” matrix to be partitioned into a grid of cells. Each of these source cells will be compared to the “target” matrix, in this case B^* . Because A and B^* are already partially aligned based on the course rotation grid Θ , we compare each source cell to B^* using a new rotation grid of $\Theta'_A = \{\theta_A^* - 2^\circ, \theta_A^* - 1^\circ, \theta_A^*, \theta_A^* + 1^\circ, \theta_A^* + 2^\circ\}$.

If two cartridge cases are truly matching, then we assume that multiple cells will “agree” on a particular translation value at the true rotation. This agreement phenomenon is illustrated in Figure 4.11 where each point represents the translation that maximizes the CCF for a particular cell and rotation. The points appear randomly distributed for most of the rotation values except around $\theta = 3$ where a tight cluster of points forms around translation $[m, n] \approx [17, -16]$. This is evidence to suggest that a true registration exists for

these two cartridge cases, implying that they match. The task is to determine when cells reach a registration consensus.

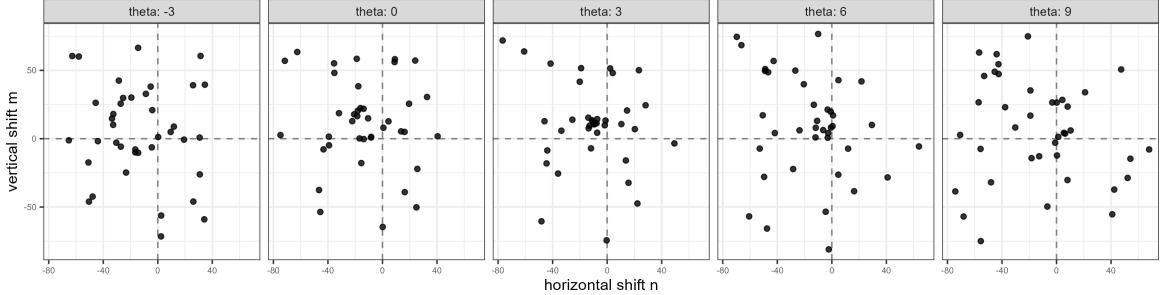


Figure 4.11: A scatterplot where points represent the cell-wise estimated translations faceted by rotation for a matching pair of cartridge cases. As evidenced by the tight cluster in the middle facet, it appears that multiple cells agree on a translation of $[m, n] \approx [17, -16]$ after rotating by 3° . Points are jittered for visibility.

A.2 Registration-Based Feature Distributions

Figure 4.12 shows density plots of the registration-based features for 21,945 cartridge case pairs. The first two rows show densities for the sample mean and standard deviation of the cell-based registrations, respectively. The third row shows densities for the pairwise-complete correlation features. The standard deviation of the cell-based registrations discriminate more between match vs. non-match pairs than the sample means, which justifies their inclusion in the final feature set. [More to say here?]

B DBSCAN Algorithm Details

DBSCAN has two parameters: a neighborhood radius ϵ and a minimum point threshold $Minpts$. First, the algorithm identifies cluster “core” points that contain at least $Minpts$ points within an ϵ distance.⁴ These points form the beginning of a cluster. All points within an ϵ -neighborhood of a core point are included in the associated cluster. Clusters whose core points are within each other’s ϵ -neighborhood are combined into a single cluster.

⁴Euclidean distance, in our application

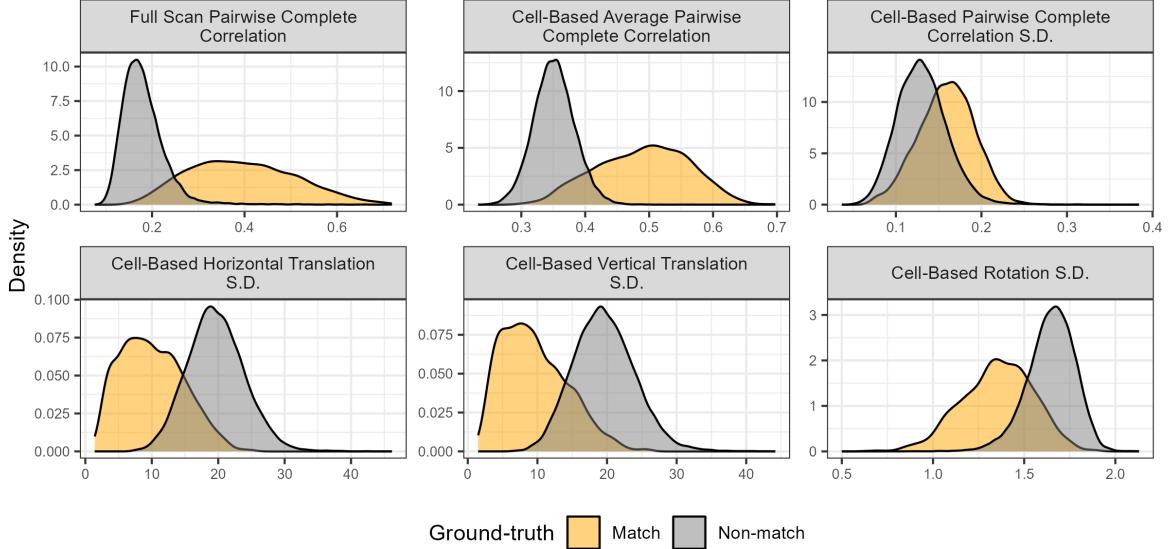


Figure 4.12: Density plots of the Registration-Based features for 21,945 cartridge case pairs. The standard deviation of the cell-based registrations distinguish between match and non-match pairs better than the mean values.

Any point outside of the ϵ -neighborhood of a core point are called “noise points.” Unlike other clustering algorithms, the DBSCAN algorithm does not require a specified number of expected clusters as a parameter; any points not belonging to a cluster are “noise.”

[Add image to illustrate DBSCAN?]

B.1 Density-Based Feature Distributions

Figure 4.13 shows the distributions of the density-based features C , Δ_θ , and Δ_{trans} . The stacked bar chart in the top-left shows the proportion of comparisons where no DBSCAN cluster is identified by outcome (match or non-match). We see that the vast majority of comparisons for which no DBSCAN cluster is identified are non-match comparisons, indicating that C_0 is a good indicator of ground-truth. In fact, there is only one non-match comparison that resulted in a DBSCAN cluster. It’s difficult to see in the plots, but the C value for this non-match pair is 5 and the Δ_{trans} value is 23.9. As expected, C tends to be relatively large for matching comparisons while Δ_θ and Δ_{trans} tends to be small.

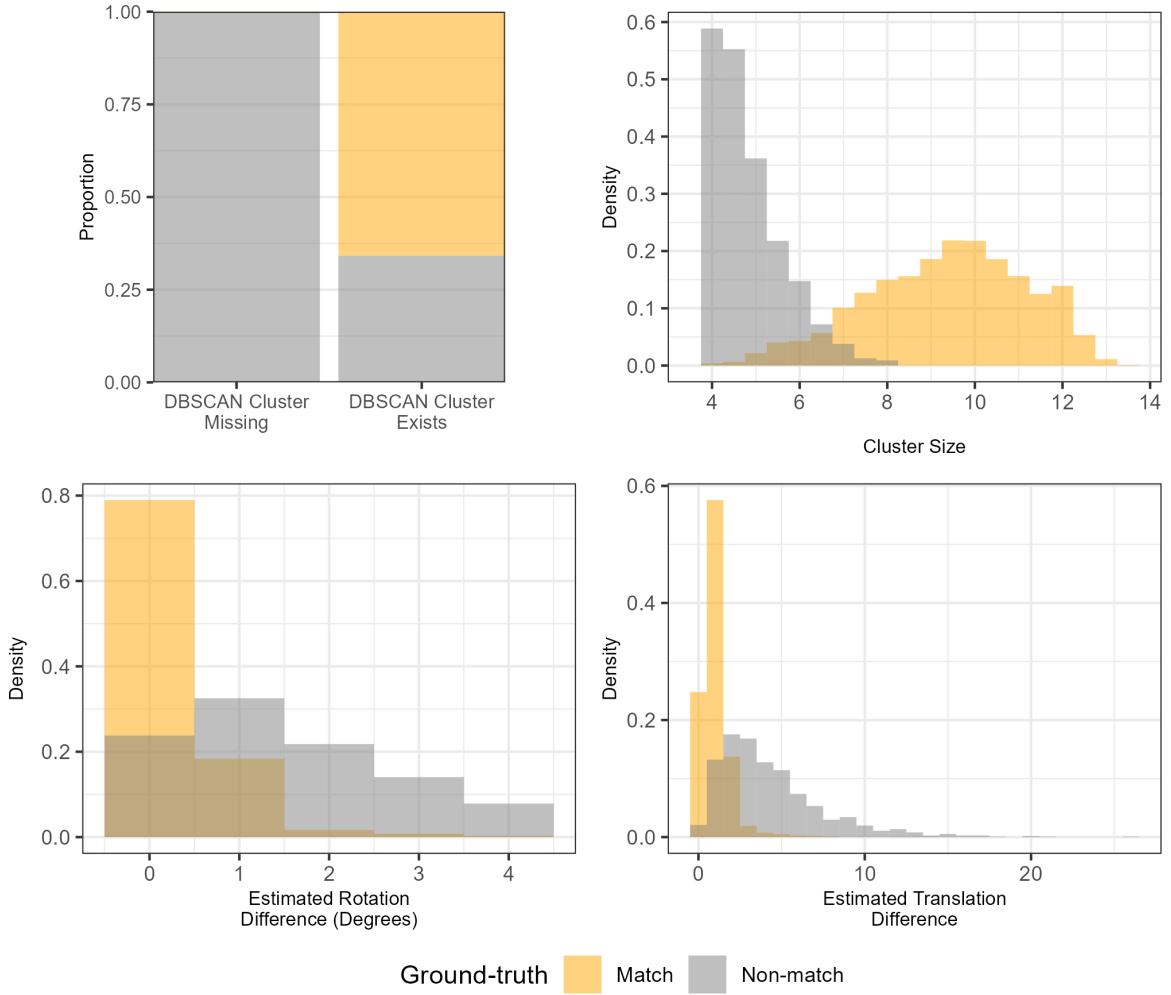


Figure 4.13: Distributions of the density-based features for 21,945 cartridge case pairs. The Cluster Size and Estimated Translation Difference features may be missing (NA) if no DBSCAN cluster is identified, which commonly occurs for non-matching cartridge case pairs as evidenced by the stacked bar chart in the top left. This explains the near absence of non-matching comparisons from Cluster Size and Estimated Translation Difference plots. Whether a cluster is identified for a particular comparison strongly predicts whether it is a match or a non-match, which justifies the inclusion of the cluster indicator feature C_0 .

C Visual Diagnostic Details

The Complementary Comparison Plot visualizes the similarities and differences between two scans. Figure 4.14 shows a Complementary Comparison plot between scan A and B^* defined previously. The left column shows Scans A and B^* . The middle column shows a filtered element-wise average between A and B^* ; namely $\mathcal{F}_{|A-B^*|<\tau}\left(\frac{1}{2}(A+B^*)\right)$. This filtered element-wise average emphasizes similarities between A and B^* . The right column shows $\mathcal{F}_{|A-B^*|>\tau}(A)$ and $\mathcal{F}_{|A-B^*|>\tau}(B^*)$ on top and bottom, respectively. These plots emphasize the differences between the two scans. The complementary comparison plot is a powerful tool for assessing the estimated alignment and identifying similarities and differences between two surface matrices. We repeat this in the other comparison direction ($d = B$) to obtain filtered matrices $\mathcal{F}_{|A^*-B|<\tau}\left(\frac{1}{2}(A^*+B)\right)$, $\mathcal{F}_{|A^*-B|>\tau}(A^*)$ and $\mathcal{F}_{|A^*-B|>\tau}(B)$.⁵

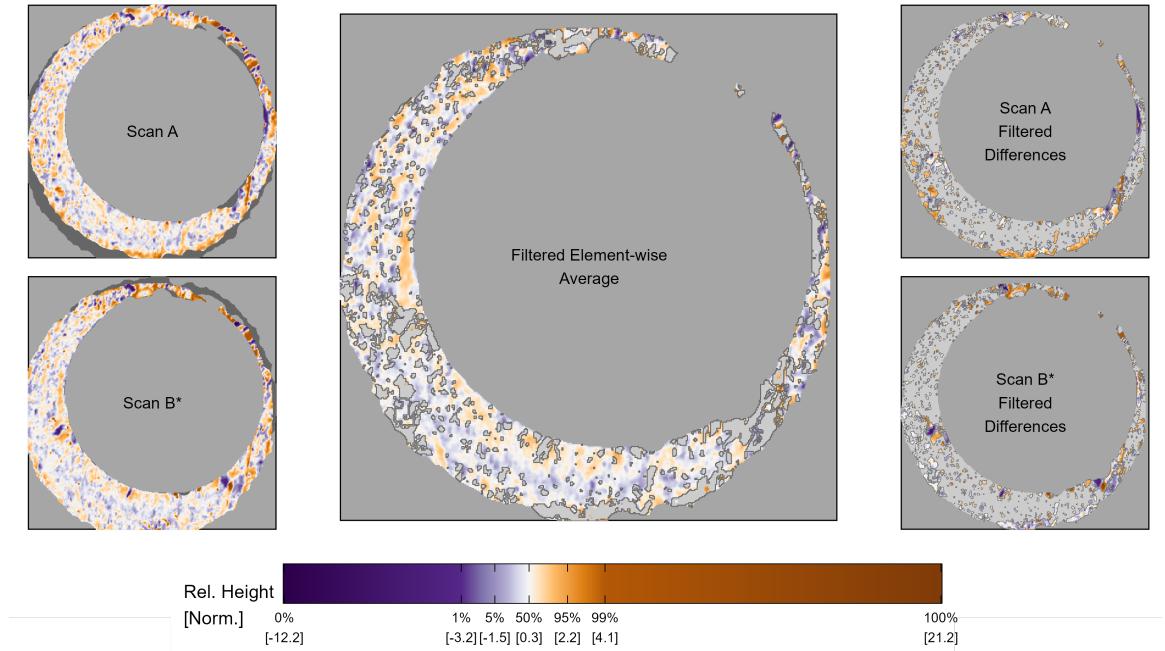


Figure 4.14: Full scan comparison plot.

⁵As with the registration-based features, in reality these matrices should be equivalent across the two comparison directions. However, there are slight differences due to the discretely-indexed nature of the surface matrices.

We make a series of qualitative assumptions related to how a Complementary Comparison Plot will look for matching and non-matching cartridge case pairs. We develop a set of features that measure the degree to which these assumptions are met by a particular cartridge case pair.

C.1 Visual Diagnostic Feature Distributions

Figure 4.15 shows the distribution of the six visual diagnostic-based features. As expected, matching comparisons at the full-scan and cell-based levels tend to have smaller neighborhood sizes and higher correlation values on average.

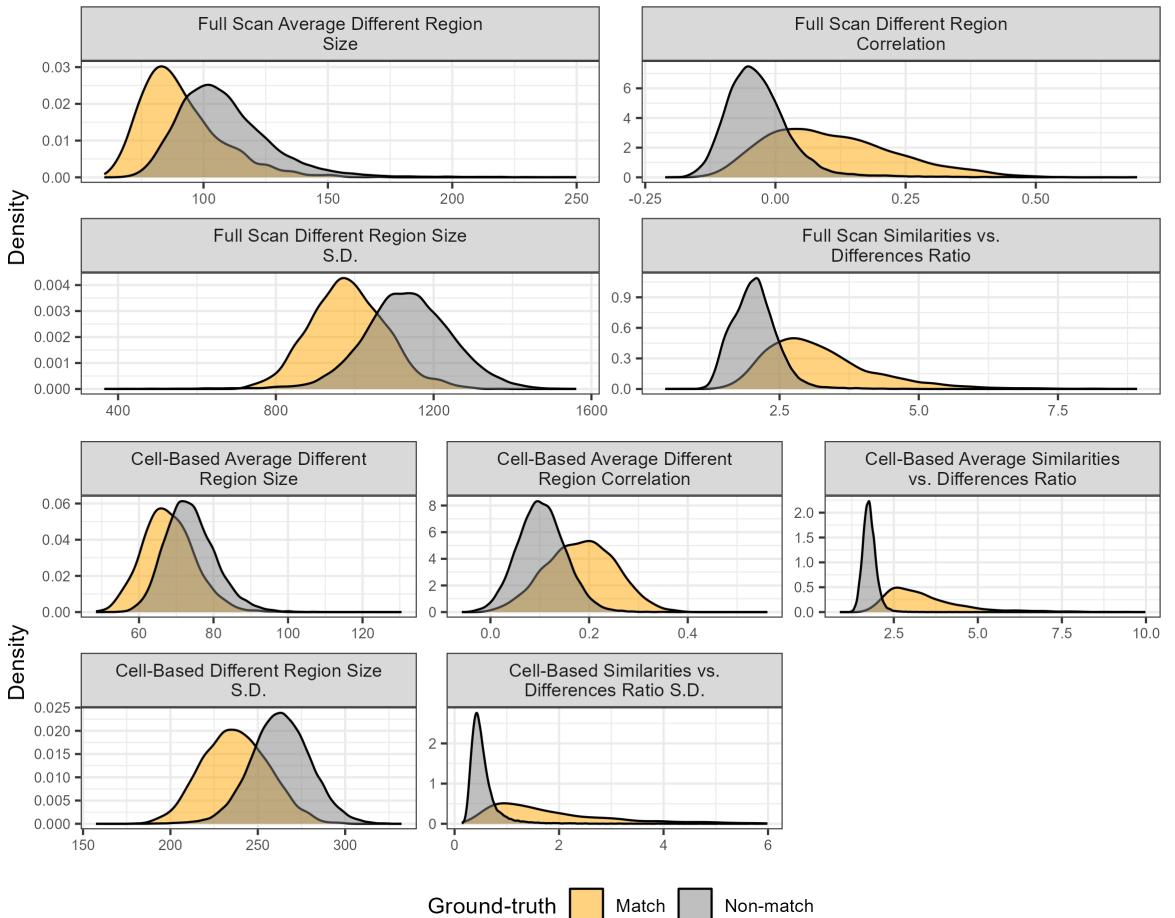


Figure 4.15: Distributions of the visual diagnostic-based features for 21,945 cartridge case pairs. Matching comparisons tend to have smaller neighborhood sizes on average and higher correlation values than non-matches indicating their utility in a classifier.

D Model-Specific Results

Table 4.6 summarizes the accuracy, true positive, and true negative rates based on the training data for the 13 binary classifier models. We see that the Logistic Regression (LR) and Random Forest (RF) models perform comparably, particularly having the exact same True Negative rate in the last two rows of the table. The CART model performs consistently worse compared to the other two models.

Table 4.7 summarizes the accuracy, true positive, and true negative rates based on the test data for the 13 binary classifier models. We see that the Logistic Regression (LR) model performs slightly better than the Random Forest (RF) model in most cases while the CART model consistently lags behind the other two. The true positive rates for the test data are noticeably lower to those for the training data summarized in Table 4.6, although the true negative rates are similar.

[Write full logistic regression model here]

Figure Figure ?? shows the trained CART model. For each node, The first line is the predicted class if it were treated as a terminal node. The second line is the estimated probability that an observation is a non-match given that it falls into the node. The third line is the percentage of all observations that fall into the node. We see that splits are made on the three variables that are ranked as most important in Figure Figure 4.10; namely, C_0 , $\overline{cor}_{\text{cell}}$, and s_{m^*} .

[Figure of fitted CART model here.]

Table 4.6: Accuracy, True Positive, and True Negative rates based on the training data for the 13 binary classifier models. This table shows a numeric summary of the results shown in [Figure]. We bold the largest values in each column for emphasis.

Feature Set	Criterion	Model	Accuracy	True Negative	True Positive
C_0		Baseline	94.67	94.37	97.32
$C_0 + \text{Registration}$	Max. AUC	CART	94.32	93.95	97.68
$C_0 + \text{Registration}$	Max. AUC	RF	97.09	97.09	97.09
$C_0 + \text{Registration}$	Max. AUC	LR	97.33	97.33	97.32
$C_0 + \text{Registration}$	Max. Accuracy	CART	98.11	98.89	91.09
$C_0 + \text{Registration}$	Max. Accuracy	RF	98.90	99.55	93.09
$C_0 + \text{Registration}$	Max. Accuracy	LR	98.93	99.53	93.59
All ACES	Max. AUC	CART	97.37	98.47	87.54
All ACES	Max. AUC	RF	98.34	98.34	98.32
All ACES	Max. AUC	LR	98.25	98.25	98.23
All ACES	Max. Accuracy	CART	98.56	99.87	86.86
All ACES	Max. Accuracy	RF	99.50	99.90	95.91
All ACES	Max. Accuracy	LR	99.56	99.90	96.50

Table 4.7: Accuracy, True Positive, and True Negative rates based on the test data for the 13 binary classifier models. This table shows a numeric summary of the results shown in [Figure]. We bold the largest values in each column for emphasis.

Feature Set	Criterion	Model	Accuracy	True Negative	True Positive
C_0		Baseline	94.20	94.16	94.73
$C_0 + \text{Registration}$	Max. AUC	CART	93.82	93.73	95.16
$C_0 + \text{Registration}$	Max. AUC	RF	96.79	97.00	93.93
$C_0 + \text{Registration}$	Max. AUC	LR	96.97	97.18	94.03
$C_0 + \text{Registration}$	Max. Accuracy	CART	98.21	98.94	88.22
$C_0 + \text{Registration}$	Max. Accuracy	RF	98.81	99.48	89.81
$C_0 + \text{Registration}$	Max. Accuracy	LR	98.78	99.42	90.10
All ACES	Max. AUC	CART	96.80	97.45	88.09
All ACES	Max. AUC	RF	97.55	97.71	95.42
All ACES	Max. AUC	LR	97.86	98.03	95.59
All ACES	Max. Accuracy	CART	98.88	99.78	86.73
All ACES	Max. Accuracy	RF	99.34	99.85	92.44
All ACES	Max. Accuracy	LR	99.38	99.88	92.63

REFERENCES

- 30 Magazine Clip (2017). Calibers explained. <https://www.30magazineclip.com/the-firearms-crash-course/calibers-explained/>.
- AFTE Criteria for Identification Committee (1992). Theory of identification, range striae comparison reports and modified glossary definitions. *AFTE Journal*, 24(3):336–340.
- Allaire, J., Xie, Y., R Foundation, Wickham, H., Journal of Statistical Software, Vaidyanathan, R., Association for Computing Machinery, Boettiger, C., Elsevier, Brozman, K., Mueller, K., Quast, B., Pruijm, R., Marwick, B., Wickham, C., Keyes, O., Yu, M., Emaasit, D., Onkelinx, T., Gasparini, A., Desautels, M.-A., Leutnant, D., MDPI, Taylor and Francis, Ögreden, O., Hance, D., Nüst, D., Uvesten, P., Campitelli, E., Muschelli, J., Hayes, A., Kamvar, Z. N., Ross, N., Cannoodt, R., Luguern, D., Kaplan, D. M., Kreutzer, S., Wang, S., Hesselberth, J., and Dervieux, C. (2021). *rticles: Article Formats for R Markdown*. R package version 0.18.
- American Academy of Forensic Sciences (2021). What is forensic science?
- Anscombe, F. J. (1973). Graphs in statistical analysis. *The American Statistician*, 27(1):17. <https://doi.org/10.2307/2682899>.
- Aurich, V. and Weule, J. (1995). Non-linear gaussian filters performing edge preserving diffusion. In *Informatik aktuell*, pages 538–545. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-79980-8_63.

- Bache, S. M. and Wickham, H. (2022). *magrittr: A Forward-Pipe Operator for R.* R package version 2.0.2.
- Baldwin, D. P., Bajic, S. J., Morris, M., and Zamzow, D. (2014). A study of false-positive and false-negative error rates in cartridge case comparisons. Technical report. <https://doi.org/10.21236/ada611807>.
- Barthelme, S. (2019). *imager: Image Processing Library Based on 'CImg'.* R package version 0.41.2.
- Beeley, C. and Sukhdev, S. R. (2018). *Web Application Development with R Using Shiny.* Packt Publishing, Birmingham, England, 3 edition.
- Belle, V. and Papantonis, I. (2021). Principles and practice of explainable machine learning. *Frontiers in Big Data*, 4.
- Berry, N., Taylor, J., and Baez-Santiago, F. (2021). *handwriter: Handwriting Analysis in R.* R package version 1.0.1.
- Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1):5–32.
- Brigham, E. O. (1988). *The Fast Fourier Transform and Its Applications.* Prentice-Hall, Inc., USA.
- Brinkman, S. and Bodschatwinna, H. (2003a). Advanced Gaussian filters. In Blunt, L. and Jiang, X., editors, *Advanced Techniques for Assessment Surface Topography: Development of a Basis for 3D Surface Texture Standards "SURFSTAND"*. Elsevier Inc., United States.
- Brinkman, S. and Bodschatwinna, H. (2003b). *Advanced Techniques for Assessment Surface Topography.* Elsevier. <https://doi.org/10.1016/b978-1-903996-11-9.x5000-2>.
- Brown, L. G. (1992). A survey of image registration techniques. *ACM Computing Surveys*, 24(4):325–376.

- Buja, A., Cook, D., Hofmann, H., Lawrence, M., Lee, E.-K., Swayne, D. F., and Wickham, H. (2009). Statistical inference for exploratory data analysis and model diagnostics. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 367(1906):4361–4383. <https://doi.org/10.1098/rsta.2009.0120>.
- Cadre Forensics (2019). Top match-3d high capacity: 3d imaging and analysis system for firearm forensics. <https://www.cadreforensics.com/pdf/TopMtopmatchatch-3D-HighCapacity.pdf>.
- Chang, A. C. and Li, P. (2022). Is economics research replicable? sixty published papers from thirteen journals say “often not”. *Critical Finance Review*, 11(1):185–206. <https://doi.org/10.1561/104.00000053>.
- Chang, W., Cheng, J., Allaire, J., Sievert, C., Schloerke, B., Xie, Y., Allen, J., McPherson, J., Dipert, A., and Borges, B. (2021). *shiny: Web Application Framework for R*. R package version 1.7.1.
- Chapnick, C., Weller, T. J., Duez, P., Meschke, E., Marshall, J., and Lilien, R. (2020). Results of the 3d virtual comparison microscopy error rate (VCMER) study for firearm forensics. *Journal of Forensic Sciences*, 66(2):557–570.
- Chen, Z., Song, J., Chu, W., Soons, J. A., and Zhao, X. (2017). A convergence algorithm for correlation of breech face images based on the congruent matching cells (CMC) method. *Forensic Science International*, 280:213–223.
- Chu, W., Tong, M., and Song, J. (2013). Validation Tests for the Congruent Matching Cells (CMC) Method Using Cartridge Cases Fired with Consecutively Manufactured Pistol Slides. *Journal of the Association of Firearms and Toolmarks Examiners*, 45(4):6.
- Cleveland, W. (1994). *The Elements of Graphing Data*. AT&T Bell Laboratories.
- Crawford, A. (2020). *Bayesian hierarchical modeling for the forensic evaluation of hand-written documents*. Ph.D thesis, Iowa State University.

- Crowder, M. and Hand, D. (1990). *Analysis of Repeated Measures*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis.
- Curran, J. M., Champod, T. N. H., and Buckleton, J. S., editors (2000a). *Forensic interpretation of glass evidence*. CRC Press, Boca Raton, FL.
- Curran, J. M., Champod, T. N. H., and Buckleton, J. S., editors (2000b). *Forensic interpretation of glass evidence*. CRC Press, Boca Raton, FL.
- DeFrance, C. and Arsdale, M. (2003). Validation study of electrochemical rifling. *Association of Firearms and Tool Marks Examiners Journal*, 35:35–37.
- Deng, H. (2018). Interpreting tree ensembles with inTrees. *International Journal of Data Science and Analytics*, 7(4):277–287. <https://doi.org/10.1007/s41060-018-0144-8>.
- Desai, D. R. and Kroll, J. A. (2017). Trust but Verify: A Guide to Algorithms and the Law. *Harvard Journal of Law & Technology (Harvard JOLT)*, 31(1):1–64.
- Duez, P., Weller, T., Brubaker, M., Hockensmith, R. E., and Lilien, R. (2017). Development and validation of a virtual examination tool for firearm forensics, ., *Journal of Forensic Sciences*, 63(4):1069–1084.
- Duvendack, M., Palmer-Jones, R. W., and Reed, W. (2015). Replications in economics: A progress report. *Econ Journal Watch*, 12(2). <https://EconPapers.repec.org/RePEc:ejw:journl:v:12:y:2015:i:2:p:164-191>.
- Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD’96, page 226–231. AAAI Press. doi.org/10.5555/3001460.3001507.

Fadul, T., Hernandez, G., Stoiloff, S., and Sneh, G. (2011a). An Empirical Study to Improve the Scientific Foundation of Forensic Firearm and Tool Mark Identification Utilizing 10 Consecutively Manufactured Slides.

Fadul, T., Hernandez, G., Stoiloff, S., and Sneh, G. (2011b). An Empirical Study to Improve the Scientific Foundation of Forensic Firearm and Tool Mark Identification Utilizing 10 Consecutively Manufactured Slides.

Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B., and Herrera, F. (2018). *Learning from Imbalanced Data Sets*. Springer International Publishing.

Garton, N., Ommen, D., Niemi, J., and Carriquiry, A. (2020). Score-based likelihood ratios to evaluate forensic pattern evidence. <https://doi.org/10.48550/ARXIV.2002.09470>.

Goldstein, E. and Brockmole, J. (2016). *Sensation and Perception*. CENGAGE Learning Custom Publishing, Mason, OH, 10 edition.

Goode, K. and Hofmann, H. (2021). Visual diagnostics of an explainer model: Tools for the assessment of LIME explanations. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 14(2):185–200. <https://doi.org/10.1002/sam.11500>.

Goodman, S. N., Fanelli, D., and Ioannidis, J. P. A. (2016). What does research reproducibility mean? *Science Translational Medicine*, 8(341):341ps12–341ps12.

Goor, R., Hoffman, D., and Riley, G. (2020). Novel method for accurately assessing pull-up artifacts in str analysis. *Forensic Science International: Genetics*, 51:102410.

Grüning, B., Chilton, J., Köster, J., Dale, R., Soranzo, N., van den Beek, M., Goecks, J., Backofen, R., Nekrutenko, A., and Taylor, J. (2018). Practical computational reproducibility in the life sciences. *Cell Systems*, 6(6):631–635. <https://doi.org/10.1016/j.cels.2018.03.014>.

- Gundersen, O. E., Gil, Y., and Aha, D. W. (2018). On reproducible AI: Towards reproducible research, open science, and digital scholarship in AI publications. *AI Magazine*, 39(3):56–68. <https://doi.org/10.1609/aimag.v39i3.2816>.
- Hadler, J. R. and Morris, M. D. (2017). An improved version of a tool mark comparison algorithm. *Journal of Forensic Sciences*, 63(3):849–855. <https://doi.org/10.1111/1556-4029.13640>.
- Hamby, J. E., Brundage, D. J., and Thorpe, J. W. (2009). The identification of bullets fired from 10 consecutively rifled 9mm ruger pistol barrels: A research project involving 507 participants from 20 countries. volume 41, pages 99–110.
- Hampton, D. (2016). Firearms identification. a discipline mainly concerned with determining whether a bullet or cartridge was fired by a particular weapon. - ppt download. <https://slideplayer.com/slide/9972083/>.
- Haralick, R. M., Sternberg, S. R., and Zhuang, X. (1987). Image analysis using mathematical morphology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(4):532–550. <https://doi.org/10.1109/tpami.1987.4767941>.
- Hare, E., Hofmann, H., and Carriquiry, A. (2017). Automatic Matching of Bullet Land Impressions. *The Annals of Applied Statistics*, 11(4):2332–2356.
- Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA.
- Hesselink, W. H., Meijster, A., and Bron, C. (2001). Concurrent determination of connected components. *Science of Computer Programming*, 41(2):173–194.
- Hofmann, H., Carriquiry, A., and Vanderplas, S. (2021). Treatment of inconclusives in the AFTE range of conclusions. *Law, Probability and Risk*, 19(3-4):317–364.

Hofmann, H., Vanderplas, S., Krishnan, G., and Hare, E. (2020). *x3ptools: Tools for Working with 3D Surface Measurements*. R package version 0.0.3.

Huber, W., Carey, V. J., Gentleman, R., Anders, S., Carlson, M., Carvalho, B. S., Bravo, H. C., Davis, S., Gatto, L., Girke, T., Gottardo, R., Hahne, F., Hansen, K. D., Irizarry, R. A., Lawrence, M., Love, M. I., MacDonald, J., Obenchain, V., Oleś, A. K., Pagès, H., Reyes, A., Shannon, P., Smyth, G. K., Tenenbaum, D., Waldron, L., and Morgan, M. (2015). Orchestrating high-throughput genomic analysis with bioconductor. *Nature Methods*, 12(2):115–121.

Indiana County Court of Common Pleas (2009). *Commonwealth of Pennsylvania vs. Kevin J. Foley*.

Iqbal, S. A., Wallach, J. D., Khoury, M. J., Schully, S. D., and Ioannidis, J. P. A. (2016). Reproducible research practices and transparency across the biomedical literature. *PLOS Biology*, 14(1):e1002333. <https://doi.org/10.1371/journal.pbio.1002333>.

ISO 16610-21 (2011). Geometrical product specifications (GPS) - Filtration - Part 61: Linear areal filters: Gaussian filters. Standard, International Organization for Standardization, Geneva, CH. <https://www.iso.org/standard/60159.html>.

ISO 16610-71(2014) (2014). Geometrical product specifications (GPS) - Filtration - Part 71: Robust areal filters: Gaussian regression filters. Standard, International Organization for Standardization, Geneva, CH.

ISO 25178-72(2017) (2017). Geometrical product specifications (GPS) — Surface texture: Areal — Part 72: XML file format x3p. Standard, International Organization for Standardization, Geneva, CH.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An Introduction to Statistical Learning: with Applications in R*. Springer.

- Knowles, L., Hockey, D., and Marshall, J. (2021). The validation of 3d virtual comparison microscopy (VCM) in the comparison of expended cartridge cases. *Journal of Forensic Sciences*, 67(2):516–523.
- Krishnan, G. and Hofmann, H. (2018). Adapting the chumbley score to match striae on land engraved areas (leas) of bullets,. *Journal of Forensic Sciences*, 64(3):728–740. <https://doi.org/10.1111/1556-4029.13950>.
- Kuhn, M. (2022). *caret: Classification and Regression Training*. R package version 6.0-91.
- Kwong, K. (2017). The Algorithm Says You Did It: The Use of Black Box Algorithms to Analyze Complex DNA Evidence Notes. *Harvard Journal of Law & Technology (Harvard JOLT)*, 31(1):275–302.
- Leek, J. T. and Jager, L. R. (2017). Is Most Published Research Really False? *Annual Review of Statistics and Its Application*, 4(1):109–122.
- MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In Cam, L. M. L. and Neyman, J., editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press.
- Mattijssen, E. J., Witteman, C. L., Berger, C. E., Brand, N. W., and Stoel, R. D. (2020). Validity and reliability of forensic firearm examiners. *Forensic Science International*, 307:110112. <https://doi.org/10.1016/j.forsciint.2019.110112>.
- Midway, S. R. (2020). Principles of effective data visualization. *Patterns*, 1(9):100141. <https://doi.org/10.1016/j.patter.2020.100141>.
- National Academy of Sciences, Engineering, and Medicine (2019). *Reproducibility and Replicability in Science*. National Academies Press.

National Research Council (2009). *Strengthening Forensic Science in the United States: A Path Forward*. The National Academies Press, Washington, DC.

Neuman, M., Hundl, C., Grimaldi, A., Eudaley, D., Stein, D., and Stout, P. (2022). Blind testing in firearms: Preliminary results from a blind quality control program. *Journal of Forensic Sciences*, 67(3):964–974. <https://doi.org/10.1111/1556-4029.15031>.

Ommen, D. M. and Saunders, C. P. (2018). Building a unified statistical framework for the forensic identification of source problems. *Law, Probability and Risk*, 17(2):179–197. <https://doi.org/10.1093/lpr/mgy008>.

OSAC Human Factors Committee (2020). Human factors in validation and performance testing of forensic science. Technical report. <https://doi.org/10.29325/osac.ts.0004>.

Ott, D., Thompson, R., and Song, J. (2017). Applying 3D measurements and computer matching algorithms to two firearm examination proficiency tests. *Forensic Science International*, 271:98–106.

Park, S. and Carriquiry, A. (2020). An algorithm to compare two-dimensional footwear outsole images using maximum cliques and speeded-up robust feature. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 13(2):188–199.

Park, S. and Tyner, S. (2019). Evaluation and comparison of methods for forensic glass source conclusions. *Forensic Science International*, 305:110003. <https://doi.org/10.1016/j.forsciint.2019.110003>.

Peng, R. D. (2009). Reproducible research and Biostatistics. *Biostatistics*, 10(3):405–408.

Peng, R. D. (2011). Reproducible Research in Computational Science. *Science*, 334(6060):1226–1227. Publisher: American Association for the Advancement of Science.

Piccolo, S. R. and Frampton, M. B. (2016). Tools and techniques for computational reproducibility. *GigaScience*, 5(1). <https://doi.org/10.1186/s13742-016-0135-4>.

President's Council of Advisors on Sci. & Tech. (2016). Forensic science in criminal courts: Ensuring scientific validity of feature-comparison methods.

Puiutta, E. and Veith, E. M. S. P. (2020). Explainable reinforcement learning: A survey. In *Lecture Notes in Computer Science*, pages 77–95. Springer International Publishing. https://doi.org/10.1007/978-3-030-57321-8_5.

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rattenbury, R. C. (2015). Semiautomatic pistol. <https://www.britannica.com/technology/semitomatic-pistol#/media/1/44886/66099>.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1135–1144, New York, NY, USA. Association for Computing Machinery. <https://doi.org/10.1145/2939672.2939778>.

Rice, K. E. (2020). *A Framework for Statistical and Computational Reproducibility in Large-Scale Data Analysis Projects with a Focus on Automated Forensic Bullet Evidence Comparison*. PhD thesis. Copyright - Database copyright ProQuest LLC; ProQuest does not claim copyright in the individual underlying works; Last updated - 2021-05-25.

Riva, F. and Champod, C. (2014). Automatic comparison and evaluation of impressions left by a firearm on fired cartridge cases. *Journal of Forensic Sciences*, 59(3):637–647. <https://doi.org/10.1111/1556-4029.12382>.

Riva, F., Hermsen, R., Mattijssen, E., Pieper, P., and Champod, C. (2016). Objective evaluation of subclass characteristics on breech face marks. *Journal of Forensic Sciences*, 62(2):417–422. <https://doi.org/10.1111/1556-4029.13274>.

- Riva, F., Mattijssen, E. J., Hermsen, R., Pieper, P., Kerkhoff, W., and Champod, C. (2020). Comparison and interpretation of impressed marks left by a firearm on cartridge cases – towards an operational implementation of a likelihood ratio based technique. *Forensic Science International*, 313:110363. <https://doi.org/10.1016/j.forsciint.2020.110363>.
- Roth, J., Cariveau, A., Liu, X., and Jain, A. K. (2015). Learning-based ballistic breech face impression image matching. In *2015 IEEE 7th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–8.
- Smith, T. P., Smith, G. A., and Snipes, J. B. (2016). A validation study of bullet and cartridge case comparisons using samples representative of actual casework. *Journal of Forensic Sciences*, 61(4):939–946.
- Song, J. (2013). Proposed “NIST Ballistics Identification System (NBIS)” Based on 3D Topography Measurements on Correlation Cells. *American Firearm and Tool Mark Examiners Journal*, 45(2):11.
- Song, J., Chu, W., Tong, M., and Soons, J. (2014). 3D topography measurements on correlation cells—a new approach to forensic ballistics identifications. *Measurement Science and Technology*, 25(6):064005.
- Song, J., Vorburger, T. V., Chu, W., Yen, J., Soons, J. A., Ott, D. B., and Zhang, N. F. (2018). Estimating error rates for firearm evidence identifications in forensic science. *Forensic Science International*, 284:15–32.
- Stodden, V., Guo, P., and Ma, Z. (2013). Toward Reproducible Computational Research: An Empirical Analysis of Data and Code Policy Adoption by Journals. *PLoS ONE*, 8(6):e67111.
- Stodden, V., Krafczyk, M. S., and Bhaskar, A. (2018a). Enabling the verification of computational results. In *Proceedings of the First International Workshop on Practical Reproducible Evaluation of Computer Systems*. ACM.

- Stodden, V., Seiler, J., and Ma, Z. (2018b). An empirical analysis of journal policy effectiveness for computational reproducibility. *Proceedings of the National Academy of Sciences*, 115(11):2584–2589. <https://www.pnas.org/doi/abs/10.1073/pnas.1708290115>.
- Stroman, A. (2014). Empirically determined frequency of error in cartridge case examinations using a declared double-blind format. *AFTE Journal*, 46:157–175.
- Swofford, H. and Champod, C. (2021). Implementation of algorithms in pattern & impression evidence: A responsible and practical roadmap. *Forensic Science International: Synergy*, 3:100142. <https://doi.org/10.1016/j.fsisyn.2021.100142>.
- Tai, X. H. (2019). *Matching Problems in Forensics*. PhD thesis. <https://doi.org/10.1184/R1/9963596.V1>.
- Tai, X. H. (2021). *cartridges3D: Algorithm to Compare Cartridge Case Images*. R package version 0.0.0.9000.
- Tai, X. H. and Eddy, W. F. (2018). A Fully Automatic Method for Comparing Cartridge Case Images,. *Journal of Forensic Sciences*, 63(2):440–448.
- Telea, A. C. (2014). *Data visualization: principles and practice*. CRC Press.
- The Linux Foundation (2017). Using open source software to speed up development and gain business advantage.
- Therneau, T. and Atkinson, B. (2022). *rpart: Recursive Partitioning and Regression Trees*. R package version 4.1.16.
- Thompson, R. (2017). *Firearm Identification in the Forensic Science Laboratory*. National District Attorneys Association.
- Tong, M., Song, J., and Chu, W. (2015). An Improved Algorithm of Congruent Matching Cells (CMC) Method for Firearm Evidence Identifications. *Journal of Research of the National Institute of Standards and Technology*, 120:102.

- Tong, M., Song, J., Chu, W., and Thompson, R. M. (2014). Fired Cartridge Case Identification Using Optical Images and the Congruent Matching Cells (CMC) Method. *Journal of Research of the National Institute of Standards and Technology*, 119:575.
- Tvedebrink, T., Andersen, M. M., and Curran, J. M. (2020). Dnatoools: Tools for analysing forensic genetic dna data. *Journal of Open Source Software*, 5(45):1981.
- Tyner, S., Soyoung Park, Krishnan, G., Pan, K., Hare, E., Luby, A., Tai, X. H., Hofmann, H., and Basulto-Elias, G. (2019). sctyner/openforscir: Create doi for open forensic science in r. <https://doi.org/10.5281/ZENODO.3418141>.
- Ulery, B. T., Hicklin, R. A., Buscaglia, J., and Roberts, M. A. (2011). Accuracy and reliability of forensic latent fingerprint decisions. *Proceedings of the National Academy of Sciences*, 108(19):7733–7738. <https://doi.org/10.1073/pnas.1018707108>.
- Ulery, B. T., Hicklin, R. A., Buscaglia, J., and Roberts, M. A. (2012). Repeatability and reproducibility of decisions by latent fingerprint examiners. *PLoS ONE*, 7(3):e32800. <https://doi.org/10.1371/journal.pone.0032800>.
- Ulery, B. T., Hicklin, R. A., Roberts, M. A., and Buscaglia, J. (2014). Measuring what latent fingerprint examiners consider sufficient information for individualization determinations. *PLoS ONE*, 9(11):e110179. <https://doi.org/10.1371/journal.pone.0110179>.
- Vanderplas, S., Nally, M., Klep, T., Cadevall, C., and Hofmann, H. (2020). Comparison of three similarity scores for bullet LEA matching. *Forensic Science International*.
- Venables, W. N. and Ripley, B. D. (2002). *Modern Applied Statistics with S*. Springer, New York, fourth edition. ISBN 0-387-95457-0.
- Vorburger, T. V., Song, J., and Petraco, N. (2015). Topography measurements and applications in ballistics and tool mark identifications. *Surface Topography: Metrology and Properties*, 4(1):013002.

- Vorburger, T. V., Yen, J. H., Bachrach, B., Renegar, T. B., Filliben, J. J., Ma, L., Rhee, H. G., Zheng, A., Song, J. F., Riley, M., Foreman, C. D., and Ballou, S. M. (2007). Surface topography analysis for a feasibility assessment of a national ballistics imaging database. Technical Report NIST IR 7362, National Institute of Standards and Technology, Gaithersburg, MD. Edition: 0.
- Weller, T., Brubaker, M., Duez, P., and Lilien, R. (2015). Introduction and initial evaluation of a novel three-dimensional imaging and analysis system for firearm forensics. *AFTE Journal*, 47:198.
- Weller, T. J., Zheng, A., Thompson, R., and Tulleners, F. (2012). Confocal microscopy analysis of breech face marks on fired cartridge cases from 10 consecutively manufactured pistol slides. 57(4). <https://doi.org/10.1111/j.1556-4029.2012.02072.x>.
- Werner, D., Berthod, R., Rhumorbarbe, D., and Gallusser, A. (2021). Manufacturing of firearms parts: Relevant sources of information and contribution in a forensic context. *WIREs Forensic Science*, 3(3):e1401. <https://doi.org/10.1002/wfs2.1401>.
- Wickham, H. (2009). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York.
- Wickham, H. (2014). Tidy data. *The Journal of Statistical Software*, 59.
- Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T. L., Miller, E., Bache, S. M., Müller, K., Ooms, J., Robinson, D., Seidel, D. P., Spinu, V., Takahashi, K., Vaughan, D., Wilke, C., Woo, K., and Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, 4(43):1686. <https://doi.org/10.21105/joss.01686>.
- Wilkinson, L. (2005). *The Grammar of Graphics*. Springer-Verlag, Berlin, Heidelberg.
- Xiao Hui Tai (2018). Comparing cartridge breechface marks: 2d versus 3d.

- Xie, Y. (2014a). knitr: A comprehensive tool for reproducible research in R. In Stodden, V., Leisch, F., and Peng, R. D., editors, *Implementing Reproducible Computational Research*. Chapman and Hall/CRC. ISBN 978-1466561595.
- Xie, Y. (2014b). knitr: A comprehensive tool for reproducible research in R. In Stodden, V., Leisch, F., and Peng, R. D., editors, *Implementing Reproducible Computational Research*. Chapman and Hall/CRC. ISBN 978-1466561595.
- Xie, Y. (2015). *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition. ISBN 978-1498716963.
- Zemmels, J., Hofmann, H., and VanderPlas, S. (2022a). *cmcR: An Implementation of the 'Congruent Matching Cells' Method*. R package version 0.1.9.
- Zemmels, J., Hofmann, H., and Vanderplas, S. (2022b). Zemmels et al. (2023) Cartridge Case Scans.
- Zemmels, J., VanderPlas, S., and Hofmann, H. (2023). A study in reproducibility: The congruent matching cells algorithm and cmcR package. *The R Journal*, 14(4):79–102.
- Zhang, H., Zhu, J., Hong, R., Wang, H., Sun, F., and Malik, A. (2020). Convergence-improved congruent matching cells (CMC) method for firing pin impression comparison. *Journal of Forensic Sciences*, 66(2):571–582. <https://doi.org/10.1111/1556-4029.14634>.
- Zheng, X., Soons, J., Thompson, R., Singh, S., and Constantin, C. (2020). NIST ballistics toolmark research database. *Journal of Research of the National Institute of Standards and Technology*, 125. <https://doi.org/10.6028/jres.125.004>.
- Zheng, X., Soons, J., Vorburger, T. V., Song, J., Renegar, T., and Thompson, R. (2014). Applications of surface metrology in firearm identification. *Surface Topography: Metrology and Properties*, 2(1):014012. <https://doi.org/10.1088/2051-672x/2/1/014012>.

Zheng, X. A., Soons, J. A., and Thompson, R. M. (2016). NIST Ballistics Toolmark Research Database.

Zimmerman, N., Wilson, G., Raniere Silva, Ritchie, S., Michonneau, F., Oliver, J., Dashnow, H., Boughton, A., Teucher, A., Mawdsley, D., MacDonald, A., Rice, T., Emonet, R., Daigle, R., Mills, B., Bolker, B., Penrose, S., Sloggett, C., Blischak, J., Moore, T. E., Mawdsley, D., Arnold, J., Bridges, D., Becker, E. A., Riva, G. V. D., Ing-Simmons, L., Research Bazaar, Bekolay, T., Piaskowski, J., Sze, M., Hadley, M. J., Hejazi, N., Mayer, F., Leinweber, K., Deer, L., Lesniak, N., Burge, O. R., Martinez, P. A., Conrado, A. C., Jankevics, A., Ashander, J., Duckles, J., Zappia, L., Burle, M.-H., Mitchell, N., Bouchet, P., Harris, R. M., Renaut, S., Sparks, A. H., Daniel, Attali, D., Tyre, D., Morrison, E., McDonald, G., Bar, I., Mickley, J., McDevitt-Irwin, J., Koziar, K., Samuk, K., Marwaha, K., Chatzidimitriou, K., Chang, L., Kardish, M., Potter, N., Boersch-Supan, P., Funkhouser, S. A., Magle, T., Waiteb5, Ahsan Ali Khoja, Lee, A., Berlanga-Taylor, A., Ashwin Srinath, Bippuspm, Beheim, B., Butterflyskip, Harris, D. J., Oliveira, D. R., Balamuta, J., Quan, J., Woo, K., Hertweck, K., Ottoboni, K., Weitemier, K., Nederbragt, L., Lonsdale, A., Johnston, L. W., Frassl, M., Dunning, M., Donovan, M., Clark, M., Jackson, M., Cadzow, M., Narayanan Raghupathy, Sélem, N., Bachant, P., Banaszkiewicz, P., Barnes, R., Bagchi, R., Brosda, S., Munro, S., Lavrentovich, S., Rossum, T. V., Kelly, T. C., Vicken Hillis, West, K. A., and Takemon, Y. (2019). *swcarpentry/r-novice-gapminder: Software carpentry: R for reproducible scientific analysis*, june 2019. <https://doi.org/10.5281/ZENODO.3265164>.