

This is us: making CSAFE stronger each
week

CSAFE

2019-09-19

Contents

Chapter 1

Prerequisites

This is a *sample* book written in **Markdown**. You can use anything that Pandoc's Markdown supports, e.g., a math equation $a^2 + b^2 = c^2$.

The **bookdown** package can be installed from CRAN or Github:

```
install.packages("bookdown")
# or the development version
# devtools::install_github("rstudio/bookdown")
```

Remember each Rmd file contains one and only one chapter, and a chapter is defined by the first-level heading #.

To compile this example to PDF, you need XeLaTeX. You are recommended to install TinyTeX (which includes XeLaTeX): <https://yihui.name/tinytex/>.

Chapter 2

Introduction

This section will become the section for the administrative updates/organization once we have figured out how to use all of the bookdown features for our purposes.

You can label chapter and section titles using `{#label}` after them, e.g., we can reference Chapter `??`. If you do not manually label them, there will be automatic labels anyway, e.g., Chapter `??`.

Figures and tables with captions will be placed in `figure` and `table` environments, respectively.

```
par(mar = c(4, 4, .1, .1))
plot(pressure, type = 'b', pch = 19)
```

Reference a figure by its code chunk label with the `fig:` prefix, e.g., see Figure `??`. Similarly, you can reference tables generated from `knitr::kable()`, e.g., see Table `??`.

```
knitr::kable(
  head(iris, 20), caption = 'Here is a nice table!',
  booktabs = TRUE
)
```

You can write citations, too. For example, we are using the `bookdown` package (?) in this sample book, which was built on top of R Markdown and `knitr` (?).

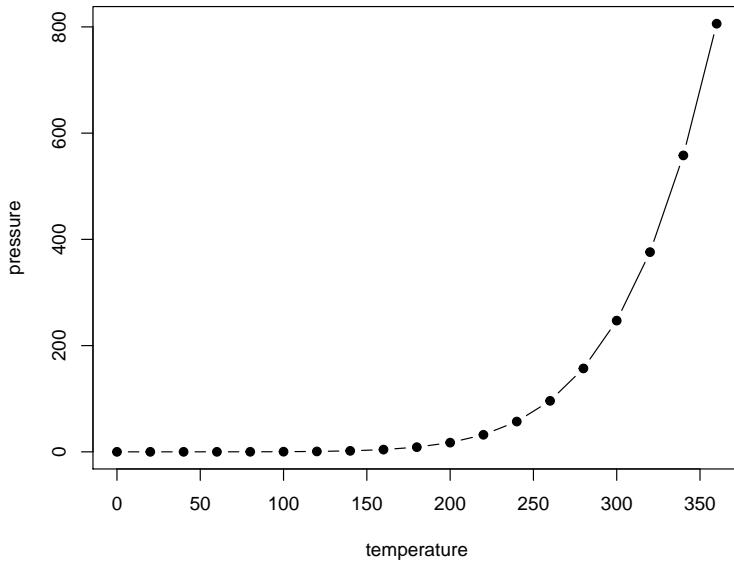


Figure 2.1: Here is a nice figure!

Table 2.1: Here is a nice table!

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5.0	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa
5.4	3.7	1.5	0.2	setosa
4.8	3.4	1.6	0.2	setosa
4.8	3.0	1.4	0.1	setosa
4.3	3.0	1.1	0.1	setosa
5.8	4.0	1.2	0.2	setosa
5.7	4.4	1.5	0.4	setosa
5.4	3.9	1.3	0.4	setosa
5.1	3.5	1.4	0.3	setosa
5.7	3.8	1.7	0.3	setosa
5.1	3.8	1.5	0.3	setosa

Chapter 3

Project CC: Bullets and Cartridge Cases

For both bullets and cartridge cases we are dealing with several inter-related aspects, that we want to address independently.

Those are:

1. data collection
2. computational tools
3. similarity scores
 1. for bullet lands:
 - a. crosscut identification
 - b. groove location
 - c. curvature removal
 - d. alignment of signatures
 - e. feature extraction
 - f. matching with trained Random Forest
 2. for breech faces
4. analysis of results
5. communication of results and methods

3.1 Data Collection

3.1.1 LAPD

All bullets are collected by Srinivasan Rathinam, LAPD.

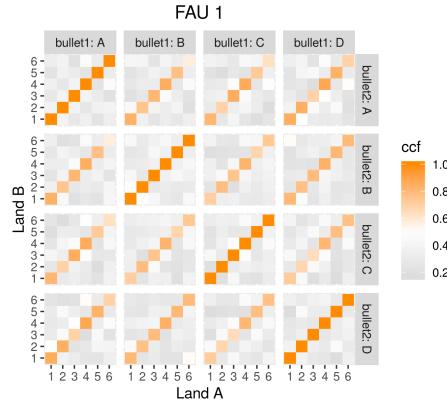


Figure 3.1: Results from assessing scans of barrel FAU 1 similarity.

3.1.1.1 Main study

4 bullets per barrel for 626 Beretta 92 F/FS firearms , ammunition used are 9 mm Luger Winchester 115 grain with a Copper surface.

scans are on Raven.

evaluation: Yawei is going to work through all 626 barrels of knowns to assess similarity scores

Why some of the cases failed? ($181/626 = 30\%$)

`x3p_crosscut_optimize()` failed to find the positions to get cross cut for some lands.

Manual identification of grooves now...

3.1.1.2 follow-up study

4 bullets per barrel for 96 of the original 626 Beretta firearms using different ammunition

bullets are being scanned

3.1.2 Hamby Sets

Scans for Hamby Sets 10, 36, 44, and 224

Scans for 3 replicates of clones for Hamby 224

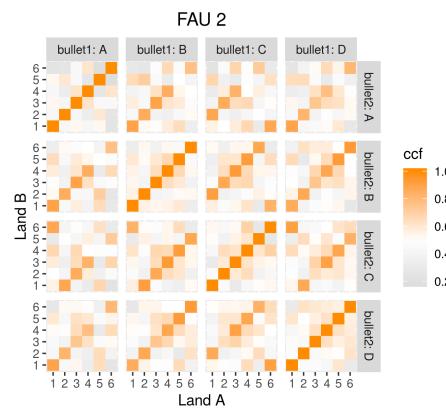


Figure 3.2: Results from assessing scans of barrel FAU 2 similarity.



Figure 3.3: Land scan for barrel FAU 3 bullet A land 6.



Figure 3.4: Land scan for barrel FAU 4 bullet C land 5.



Figure 3.5: Land scan for barrel FAU 5 bullet B land 5.

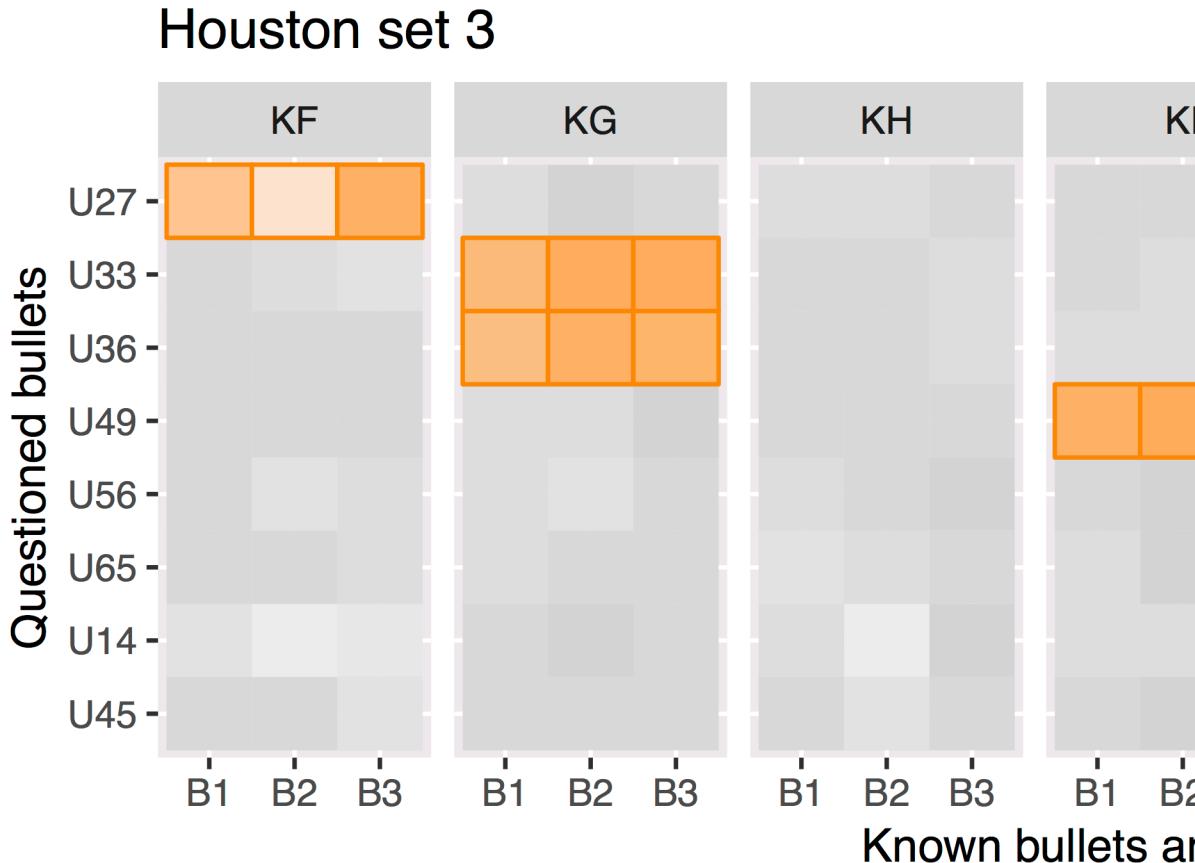


Figure 3.6: Bullet-to-bullet similarity scores for questioned bullets (y-axis) compared to all other bullets of the test set (x-axis).

3.1.3 Houston Tests

contact: Melissa Nally, Houston FSI

3.1.3.1 Pre-study

3 kits with 23 bullets each

evaluation included in submission to JFI

3.1.3.2 Study

4 kits with 20 bullets each

scans done, evaluation finished, some scans of doubtful quality

3.1.4 Houston Persistence

contact: Melissa Nally, Houston FSI

8 barrels with 40 fired bullets each

3.1.5 St Louis persistence

contact: Steve Kramer, St Louis PD

2 barrels with 192 fired bullets each (2 bullets collected every 25 shots)

3.1.6 DFSC Cartridge cases

Breech face data for knowns are scanned and available on a private github repository

evaluation

3.2 Computational Tools

3.2.1 x3ptools

`x3ptools` is an R package for working with files in x3p format. x3p is an ISO standard for describing 3d topographic surface measurements. `x3ptools` is available on CRAN, i.e. can be installed with the command `install.packages("x3ptools")`. The development version is available from github. Installation instructions and basic usage can be found at <https://heike.github.io/x3ptools/>

3.2.2 bulletxtrctr

`bulletxtrctr` is a developmental R package available from github (see <https://heike.github.io/bulletxtrctr/>) that allows an assessment of similarity scores using the data extraction pipeline described in ?.

3.2.3 grooveFinder

`grooveFinder` is a developmental R package providing different methods for identifying the location of grooves in scans of bullets. Installation instructions and some basic usage can be found at <https://heike.github.io/grooveFinder/>

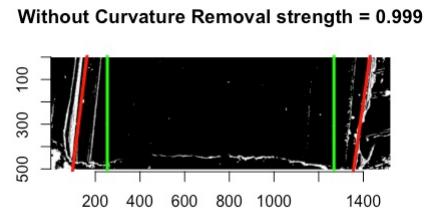


Figure 3.7: 2-dimensional visualization of example bullet br411 with .999 strength threshold



Figure 3.8: 3-dimensional visualization of example bullet br411 with .999 strength threshold

3.3 Similarity Scores

3.3.1 Bullet Lands

3.3.1.1 Approaches to identify groove locations

3.3.1.1.1 Hough Transform Method for Identifying Grooves

Charlotte 9/5/19 Update:

Will fill in with more detail later

Current Goals: - Iron-out issues with consistency of units with `get_hough_grooves`. I believe there are some issues translating from the 2-d visualization to the 3-d visualization that might have to do with inconsistent unit inputs? For Example

So either somethin is wrong with `get_mask_hough` or something is funky with the units.

- Also need to think of including a sort of rounding component where lines with slopes that are practically infinite can be viewed as a vertical line
- Compare Hough results with manual identification using score calculations from Kiegan.
- Write up results in Hough Groove Paper (It's coming I promise)
 - Create graphical images to explain line selection method



Figure 3.9: Phoenix Gun1 A-9 B1 Land 4 generated at strength threshold of 0.99, initially did not generate estimates at the 0.999 or 0.995 level

- Include 2-d and 3-d visualizations of Hough groove area identifications
- Include crosscut visualization and comparison in results

Charlotte update 09/12/19: This week I have been working on obtaining some results for the Phoenix set on Sunny. As a minor update the unit issues in `get_mask_hough()` are resolved (I think). Below is an example of a nice image that has been generated using masks.

However the mask is only as good as the Hough estimates that supports it as shown here (less nice).

Hough crosscut predictions for the Phoenix dataset are now uploaded to the bulletQuality Github in the “results” folder and contains Hough groove estimates at the following five strength levels: 0.999, 0.995, 0.99, 0.95, 0.9. The source and the crosscut estimate are also included in the dataset.



Figure 3.10: Phoenix Gun1 F-6 B2 Land 5 generated at strength threshold of 0.9, initially did not generate estimates at the 0.999 or 0.995, or 0.99 level

Here are some preliminary results of using Kiegan's area of misidentification method (thanks Kiegan!) on Hough groove estimates at the strength threshold of 0.999 in comparison to the BCP and Lasso method.

These scores are log transformed to show better separation but it's very clear that for the left groove both Lasso and BCP are out performing the Hough method in correctly identifying grooves. For the righthand side, scores tend to be more similar however once again, the Lasso method seems to be the best job since it has a larger density of low scores and minimizes high score misidentifications.

For improvement before next week, I will investigate why there are 47 missing Hough predictions resulting in a score of 0 in these results and change the parameters in the `get_grooves_hough()` function to try and generate estimates for some of those missing values.

Charlotte update 09/19/2019:

This week we are trying to think of a new way for selecting Hough lines for bullet estimates. The previous method for selecting Hough lines was to find lines with x-intercepts at the top and bottom of the lands closest to the lower and upper one sixth of the bullet lands. However this process was highly dependent on score thresholding from the Hough transform which is frustrating when running a large number of bullets since if the right score threshold was not achieved, no result would be produced. So right now I'm working on a way of selecting Hough lines from the normalized Hough scores.

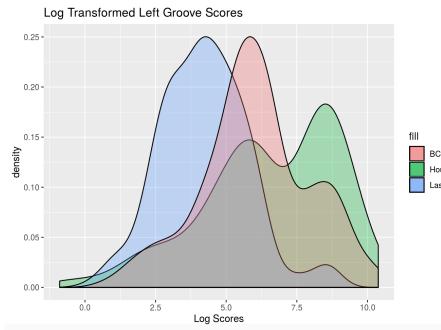


Figure 3.11: Left-hand groove area of misidentification log-transformed scores for BCP, Lasso, and Hough

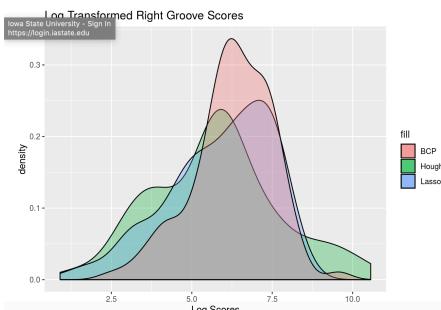


Figure 3.12: Right-hand groove area of misidentification log-transformed scores for BCP, Lasso, and Hough



Figure 3.13: Phoenix Gun 1-A9 Bullet 3 Land 1 visualized using current Hough process message



Figure 3.14: Phoenix Gun 1-A9 Bullet 3 Land 1 visualized using new Hough process message

To obtain a normalized Hough score I take the x-intercepts of each estimated Hough line generate and find the distance between the x-intercept at the top and the bottom of the land. This should give me the max possible score for each Hough line, rather than calculating based on theta. Then I take the Hough score and divide by this maximum to normalize scores between 0 and 1. Right now I am working on visualizing some of these results but my code is buggy because I'm getting negative values when I try to visualize the process using masks when I shouldn't. Here is an example of a bullet land using the old and new method. Really similar results although it would appear that the new result places the Hough transform lines further in to interior of the land than the old results. So that's promising?

3.3.1.1.2 LASSO Method

A paper is in preparation for submission to Forensic Science International describing this method (`get_grooves_lassofull` in `grooveFinder`), as well as the Bayesian changepoint method (`get_grooves_bcp`).

3.3.1.1.3 Robust LOESS Method

A paper submitted to the Journal of Forensic Science is waiting for peer review

response to the first round of revisions.

3.3.1.2 Bullet Land Comparisons Pipeline

Most data analysis processes can be thought of as a data analysis “pipeline”. This process can involve data collection, decisions about data cleaning, data transformation or reduction, and feature engineering. For example, consider the general process below:

In the case of the bullet project, we have a pipeline which starts with having two physical bullet LEAs and ends with a quantitative result, a random forest similarity score. Our pipeline could be described (roughly) as something like this:

To make this a little easier to see, we can look at how a 3D scan is processed into a 2D signature:

Now, something important to consider is whether each of these “data decisions” has an impact on the quantitative result (here, a similarity score between two LEA signatures). Consider a simple set of decisions we could make in our bullet pipeline:

If we have a pair of signatures, we could theoretically end up with 16 different similarity scores depending on the decisions we make at each point. That is also assuming that both signatures were processed in the same way at each point.

This year, I’ll be studying our bullet land “pipeline” here at CSAFE, as well as pipelines that are a little different than ours (e.g., ?). There are a few major goals I am working towards:

1. Quantifying the uncertainty of our RF similarity scores based on data decisions
2. Comparing reproducibility/robustness of differing bullet analysis approaches
 - ? vs. ?, for example
 - Crosscuts: method 1 vs. alternate? Crosscut parameter tuning?
 - Groove methods
 - Original RF vs. updated/retrained/re-engineering
3. Reproducibility/robustness of different approaches when we consider data COLLECTION.

Goal 3 is a major part of this pipeline process which I have been working on since the spring! We designed and collected a bullet scanning variability study of 9 bullets. I’m working on formally modeling the variability at the signature level, taking two major approaches:

1. Subsampling and assuming independence;
2. Directly modeling out the mean structure

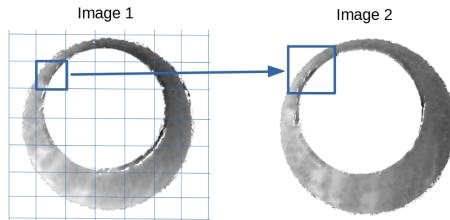


Figure 3.15: Comparing a cell in image 1 to a larger region in image 2. We wish to find the translations of the image 1 cell that yield the highest correlation within the image 2 region.

- Ignoring peak/valley dependence
- Using time series/spatial dependence modeling
- Using a Bayesian shrinkage prior (w/help from Amy!)

Results for Method 1, the subsampling, looks something like this:

I will be updating with more on the “direct” modeling in my next Spotlight!

3.3.2 Cartridge Cases

3.3.2.1 Congruent Matching Cells (CMC) algorithm for comparing cartridge case breech face impressions

Joe 9/5/19 Update: Dealing with missing values in the x3p scans continues to be an issue. The Fast Fourier Transform method for calculating cross-correlation can't handle missing data in an image, so we've attempted a few “fixes” that haven't necessarily turned out as well as expected. One idea we had was to replace the NA values in a cell with the average pixel value. However, this is artificially introducing a signal where before there was none. This can (and demonstrably has) led to inflated/incorrect correlations between cells that shouldn't have much at all in common. Unfortunately, this may be the only solution if we still wish to adhere to the CMC algorithm as described in Song et al. (2015). One improvement that I've implemented is to “crop out” the rows and columns of an image that only contain NAs. This at least means that we've weakened the strength of the artificial signal relative to the breechface's signal.

Below is a series of images that illustrate how we might compare a cell in one image to a region of another image.

For the sake of an example, let's focus on the blue outlined cell in image 1. Our goal is to use the image 1 cell to “search” a corresponding larger region in image 2 for the horizontal/vertical translations needed to produce the highest correlation. Below is a zoomed-in version of the blue outlined image 1 cell on the left and the larger image 2 region (approximately: I made the gridded image above by-hand outside of R while the images below are from R). The image 1

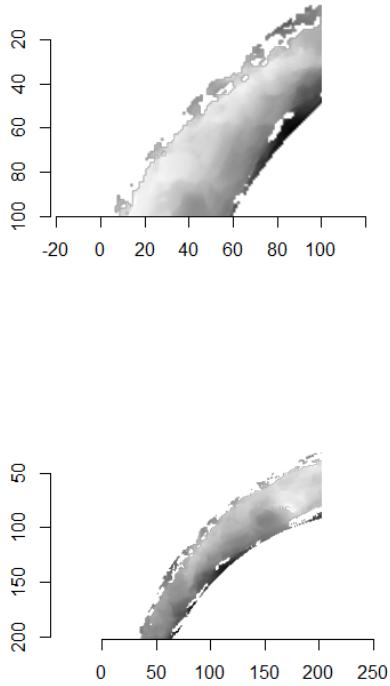


Figure 3.16: (Left) A cell from image 1. (Right) A region from image 2 centered in the same location as the image 1 cell, yet quadruple the area.

cell may look larger than the image 2 region, but we can see from the axes that the image 2 region is indeed larger. Any white pixels in the two images are NA values that need to be dealt with in some way before we can use FFTs to calculate the cross-correlation.

As already discussed above, one “solution” is to replace the NA values with the average pixel value of each image. However, to avoid creating a stronger artificial signal than necessary, we can crop-out the NA rows and columns from the two images above. Below is the cropped version of the two images. The cropping doesn’t produce significantly different images in this case, but you could imagine other examples in which a cell has captured only small amount of breechface in the corner. Such examples are fairly common and cropping significantly changes the resulting correlation values.

The last step before calculating correlation for these cells is to replace the re-

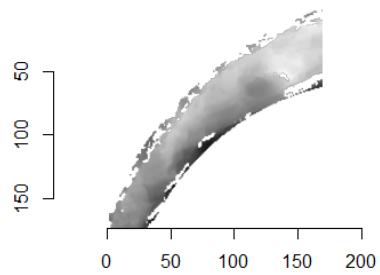
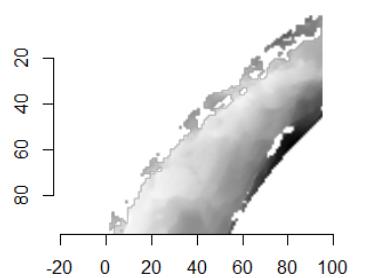


Figure 3.17: The same images as above after cropping NA rows/columns.

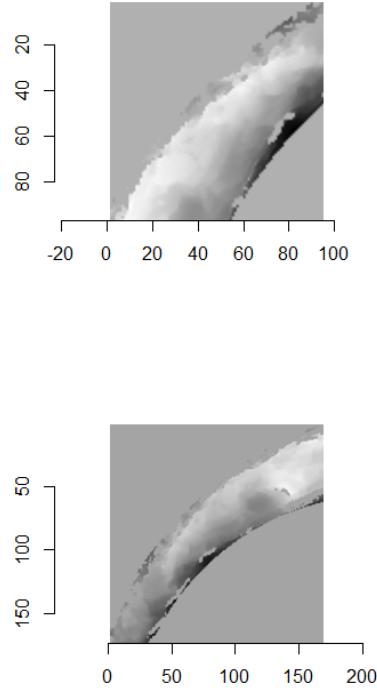


Figure 3.18: The NA-cropped images with remaining NAs replaced with the image’s average pixel values.

maining NAs with the average pixel value. This is shown below.

The cross-correlation is then calculated between these two images via a standard fast fourier transform process (see Cross-Correlation Theorem). The benefit of using such a process is that (as the name suggests) it’s faster than calculating the raw correlation between the two images. Also, the translations that produce the highest correlation between the image 1 cell and the image 2 region fall out of the calculation for free.

This pre-processing/cross-correlation calculation procedure is repeated for every cell in image 1 that contains breech face impression. Because it is not valid to assume that the two images are rotationally aligned by default, we perform the same procedure repeatedly while rotating image 2. Currently, we perform a “rough” grid search of $\theta \in [-177.5, 180]$ by increments of 2.5° . Theoretically, the final results tell us how we need to horizontally/vertically translate and rotate

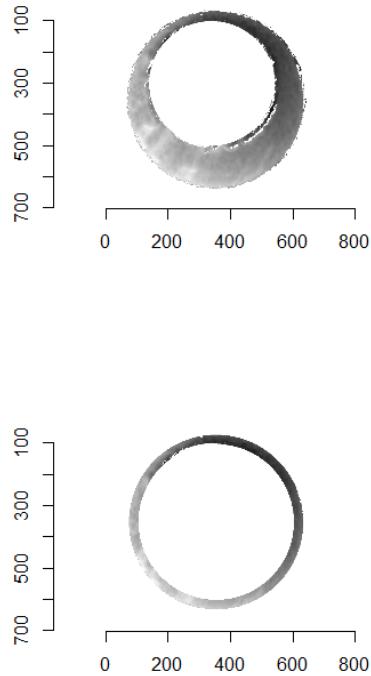


Figure 3.19: (Left) The original breech face impression scan image. (Right) A donut-shaped region cut out of the original image.

the two images to be correctly aligned.

3.3.2.2 Congruent Matching Tori: a promising solution to the missing value problem

As discussed above, dealing with missing values is proven to be a pain. The good news is that the currently-implemented CMC as described above yields results very similar to those published in Song et al. (2015) that originally describes that CMC algorithm. While our results seem to agree with currently published results, it would be nice if we could avoid needing to artificially replace missing values. We can do so if, rather than breaking up the circular breech face impression scans into disjoint squares, we break up the breech face impression into donut-shaped regions containing only breech face impression. Below is an example of such a toroidal region.

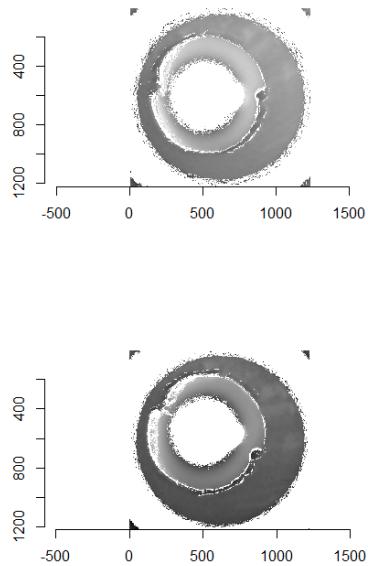


Figure 3.20: Two cartridge case scans before pre-processing.

By comparing such regions instead of the square cells, we would presumably only need to fill in a few missing value “holes” in the breech face impression scan rather than completely replacing a non-existent signal with an artificial one. In the near-future, I hope to finish up the pre-processing needed for this Congruent Matching Tori method by performing a polar transformation on these images to make them into strips that can easily be compared via an FFT.

Joe 9/12/19 Update: Before carving out toroidal regions from the two images we wish to compare, a fair amount of pre-processing needs to be completed. For example, the scans we work with begin with a considerable amount of auxiliary information, for example the firing pin impression, that we don’t want to use in our comparisons. This isn’t to say that firing pin impressions aren’t useful to determine a match between two cartridge cases. In fact there is quite a lot of published research on how to compare two firing pin impressions. Rather, it is common practice to compare breech face impressions and firing pin impressions separately since it is difficult to scan both simultaneously. Thus, there are regions of a breech face impression scan that we want to remove so that the breech face impressions are more easily comparable. Below is an example of two breech face impression scans before processing.

There are a variety of techniques to segment an image into various parts. In

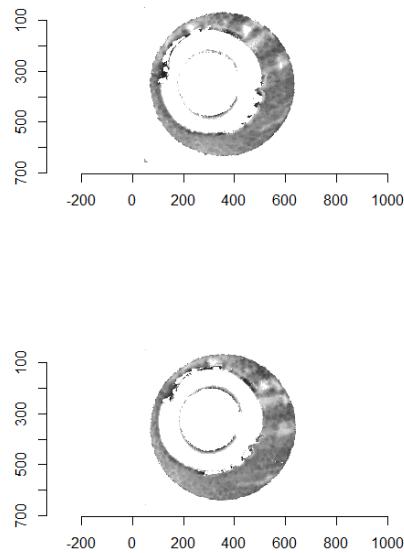


Figure 3.21: Residual values of a RANSAC plane fit to the two cartridge case scans shown above.

image processing, common techniques are the Canny edge detector, which identifies edges of shapes in an image using image gradient techniques, and the Hough Transform, which can detect a variety of geometrical shapes in an image. The Hough Transform is what is used to segment the cartridge case images used in the previous section. However, we've found that the use of a Hough Transform doesn't extract the "breech face signal" from an image as other techniques. Namely, the breech face can be effectively extracted using the RANSAC (Random sample consensus) method that iteratively fits a plane to a set of data until it settles upon a consensus-based "bulk" of the data. In the case of these cartridge case scans, the bulk of the data should predominantly be distributed around the mode height value. That is, the breech face impression. Once we've fit this plane to the breech face impression, we can extract the residuals of the fit to better accentuate the markings left in the cartridge case base by a firearm's breech face. Below is an example of the residuals left after fitting a RANSAC plane to two cartridge case scans above. In the example below, we grab any residuals less than 20 microns in magnitude.

Although these two images are of two different cartridge cases, you can hopefully see that one looks very much like a rotated version of the other. These two cartridge case scans are in fact fired from the same gun (known matches), so

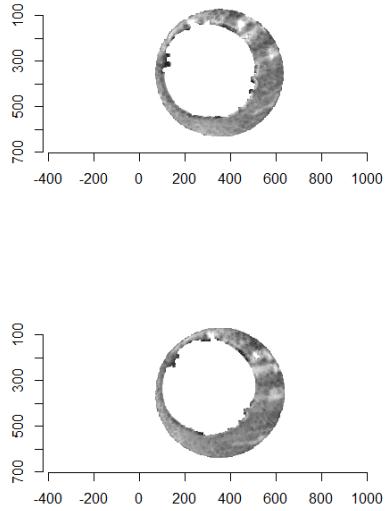


Figure 3.22: The selected breech face impressions based on dilation and erosion.

it's a good thing that they look so similar. We've now removed quite a bit of the unwanted regions of the original scans. However, there are still some areas of the image (e.g., the faint circular region of pixels in the center of the breech face scan) that just so happened to be close to the fitted plane and thus were brought along in the residual extraction. There are a few ways that we can clean up these last few areas. One is to use two Hough Transforms to detect the inner and outer circles of the breech face impression and filter out any pixels outside of the region between these two circles. The biggest issue with using a Hough Transform is that it must be given the radius of the circle that it is to search for in the image as an argument. That is, we need to know the radius of the breech face impression that we haven't yet identified in order to identify the breech face impression. Instead, we can dilate/erode (or vice-versa) the pixels in the image to remove the remaining "speckle" in the image. Below is an example of the breech face impressions cleaned via a dilation/erosion procedure.

The final step in the pre-processing is to align the two images in some consistent fashion. Luckily, the firing pin impression ring that's left after performing the above dilation/erosion provides us with some idea of how to align the breech face impressions. The location of the firing ring impression in the breech face impression provides us with an indicator of where the cartridge case was located relative to the firing pin when it was sitting in the barrel. So aligning two cartridge cases so that their firing pin impression rings align will ensure

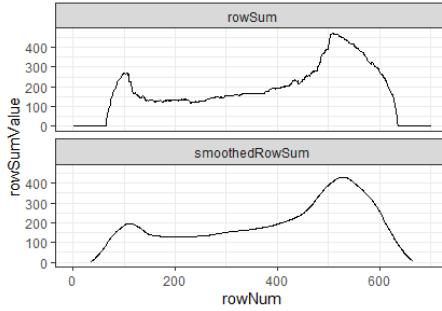


Figure 3.23: Non-NA pixel row counts and moving average-smoothed row count values plotted against row index.

that, at the very least, the breech face impression left on the cartridge case is horizontally/vertically aligned if not rotationally aligned.

Joe 9/18/19 Update: To automatically detect the radius of a given breech face impression, we can count the number of non-NA pixels in each row. If we were to imagine scanning down an image and counting the number of non-NA pixels in each row, then this count would obviously start to increase the moment we hit the top of the breech face impression. Because the breech face impressions are circular, the count would continue to increase the further down the image we scan. That is, until we hit the firing pin impression circle. At this point, because the firing pin impression circle consists of NAs, we would expect the non-NA pixel count to dip. This increasing followed by decreasing behavior in the non-NA pixel count constitutes a local maximum. We can use this local maximum of the non-NA pixel count to identify the beginning of the firing pin impression circle. Similarly, we would expect the non-NA pixel count to reach another local maximum once we hit the end of the firing pin impression circle. It's then a simple subtraction of the two row indices containing these local maxima to determine an estimate for the diameter of the firing pin impression circle.

We can see below an example of the non-NA pixel row sums plotted against the row indices (starting from the top of the image and moving down). You can hopefully see that the raw row sums are rather “noisy”. As such, we can pass a moving average smoother over the row sum values so that the local maxima are easier to identify. This may not be the most robust way to determine the local maxima. I hope to investigate the use of b-splines fit over the row sum values to see if these would be more effective at finding local maxima

However, because firing pin impression circles have somewhat perforated edges, performing one pass through the image may not yield a particularly accurate estimate. As such, we can repeat the process of finding the distance between local maxima for both the row and column non-NA pixel counts. We can also rotate the image by a few degrees and perform the same process. I am currently

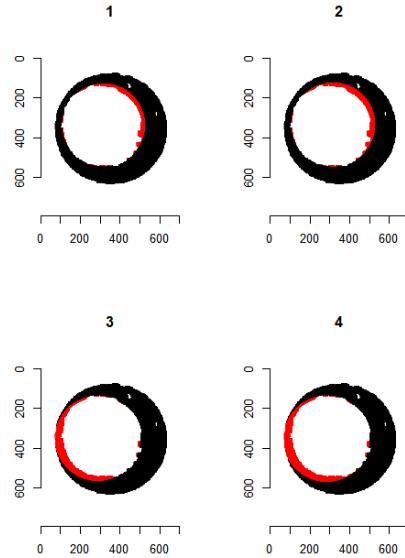


Figure 3.24: Hough Transform selected circles (red) of radius (1) 210, (2) 213, (3) 216, and (4) 219.

rotating the image 0, 15, 30, 45, 60, and 75 degrees and calculating row and column diameter estimates per rotation. Obviously we can apply whatever aggregation function we desire to these estimates to determine a final estimate. Below we see what the Hough Transform selects as the breech face for 4 different radii values. In particular, for circles of radius 210, 213, 216, and 219.

3.3.3 Modified Chumbley non-random

3.3.3.1 Land-to-land scores

3.3.3.2 Bullet-to-bullet scores

Ganesh: In this method we extend the modified chumbley non-random method from land-to-land scoring to bullet-to-bullet scoring.

3.4 Analysis of Results

3.5 Communication of Results and Methods

The results are communicated through an interactive user interface. The first part of this interface lets you add all the bullets, barrels and lands for which the random forest and other scores are to be computed. A preliminary diagnostic

of the orientations and dimensions of the lands tell us, if we can proceed safely to extraction of markings and then to cross-comparisons.

After this step, we can apply any sampling or interpolation needed on the land images, all these operations can be batched to the entire set of comparisons under consideration. Then we can make transformations like rotation, transpose etc on a sample image, visualize the results, and since we are dealing with conforming orientation and dimensions of lands present in the entire set, we can batch the transformations.

We extract markings, locate grooves, align signatures, and generate cross-comparison results. Each step is notified in UI and all steps are logged.

The scores and results are then communicated through an interactive visualization. We first interact at the top most level where we have bullet-to-bullet scores for all the cross-comparisons presented in a grid. We can select one comparison at a time which would generate a second level of grid visualization that shows the land-to-land scores for all 36 comparisons within a bullet. Interacting with this visualization, we can now pull up score tables, profiles, location of grooves, aligned signatures and raw images.

The framework of interactions, allows for validation of classification recommended by the RF model as well as gives an opportunity to critically assess, identify the cause and diagnose any problems encountered in the bullet matching pipeline.

3.5.1 Conference Presentations

3.5.1.1 American Academy of Forensic Sciences

- “Validation Study on Automated Groove Detection Methods in 3D Bullet Land Scans”
 - February 2019
 - Authors: Kiegan Rice, Ulrike Genschel, Heike Hofmann
 - Presentation given by Kiegan Rice

3.5.1.2 Association of Firearms and Toolmark Examiners Annual Training Seminar

- Heike’s talk
- “Reproducibility of Automated Bullet Matching Scores Using High-Resolution 3D LEA Scans”
 - May 2019
 - Authors: Kiegan Rice, Ulrike Genschel, Heike Hofmann
 - Presentation given by Kiegan Rice

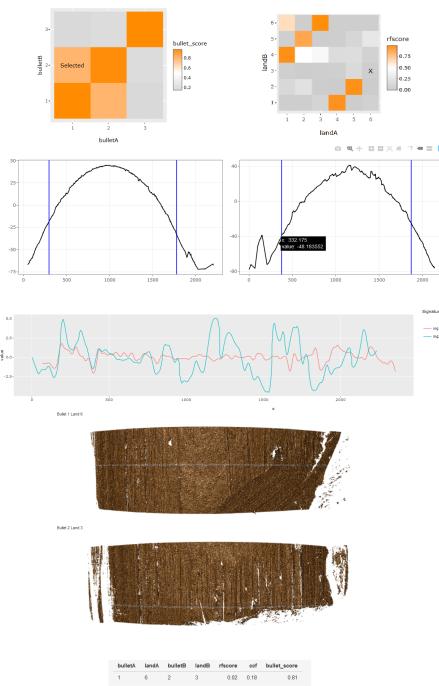


Figure 3.25: An instance of the interactive visualizations for communicating results

3.5.1.3 Joint Statistical Meetings

- “A non-parametric test for matching bullet striations: extending the chumbley score for bullet-to-bullet matching”
 - July 2019
 - Authors: Ganesh Krishnan, Heike Hofmann
 - Talk given by Ganesh Krishnan
- “Repeatability and reproducibility of automated bullet comparisons using high-resolution 3D scans”
 - July 2019
 - Authors: Kiegan Rice, Ulrike Genschel, Heike Hofmann
 - Poster presented by Kiegan Rice

3.5.1.4 Miscellaneous

- 10th International Workshop on Statistics and Simulation in Salzburg, Austria, September 2019
 - “Reproducibility of High-Resolution 3D Bullet Scans and Automated Bullet Matching Scores”
 - * Authors: Kiegan Rice, Ulrike Genschel, Heike Hofmann
 - * Poster presented by Kiegan Rice, won 2nd Springer Poster Award
 - “Case Study Validations of Automatic Bullet Matching”
 - * Authors: Heike Hofmann, Susan VanderPlas
 - * Presentation given by Alicia Carriquiry

3.6 People involved

3.6.1 Faculty

- Heike Hofmann
- Susan VanderPlas

3.6.2 Graduate Students

- Ganesh Krishnan
- Kiegan Rice
- Nate Garton
- Charlotte Roiger
- Joe Zemmels
- Yawei Ge

3.6.3 Undergraduates

- Talen Fisher (fix3p)
- Andrew Maloney

- Mya Fisher, Allison Mark, Connor Hergenreter, Carley McConnell, Anyasha Ray (scanner)

Chapter 4

Project G: Handwriting (& Signatures)

The handwriting project has four major focuses:

1. data collection
2. computational tools
3. statistical analysis
 - a. glyph clustering
 - b. closed set modeling for writer identification
4. communication of results

4.1 Data Collection

We are conducting a large data collection study to gather handwriting samples from a variety of participants across the world (most in the Midwest). Each participant provides handwriting samples at three sessions. Session packets are prepared, mailed to participants, completed, and mailed back. Once received, we scan all surveys and writing samples. Scans are loaded, cropped, and saved using a Shiny app. The app also facilitates survey data entry, saving that participant data to lines in an excel spreadsheet.

Data collection is underway with the most recent update (9/1) at 106 participants enrolled:

- 44 complete through session #3
- 52 complete through session #2
- 10 complete through session #1

As of September 2019, Marc and Anyesha are the primary contacts for the study.

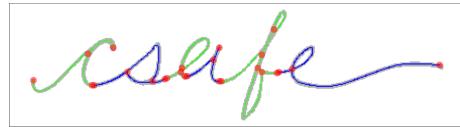


Figure 4.1: Connected text processed by ‘handwriter’. The grey background is the original pen stroke. Colored lines represent the single pixel skeleton with color changes marking glyph decomposition. Red dots mark endpoints and intersections of each glyph.

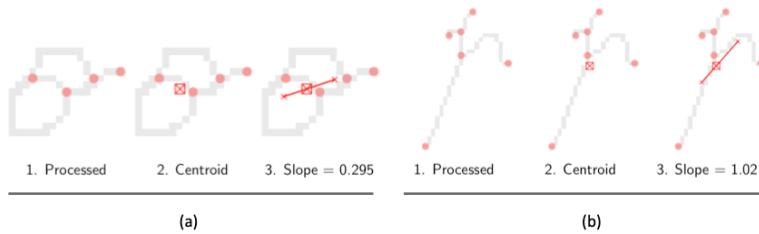


Figure 4.2: A visual of the “slope” calculation for two glyphs.

4.2 Computational Tools

`handwriter` is a developmental R package hosted at <https://github.com/CSAFE-ISU/handwriter>. It is our major computational tool for the project. The package takes in scanned handwritten documents and the following are performed.

1. Binarize. Turn the image to pure black and white.
2. Skeletonize. Reduce writing to a 1 pixel wide skeleton.
3. Break. Connected writing is decomposed into small manageable pieces called **glyphs**. Glyphs are graphical structures with nodes and edges that often, but not always, correspond to Roman letters, and are the smallest unit of observation we consider for statistical modelling.
4. Measure. A variety of measurements are taken on each glyph.

For an input document, functions in the package give back a list of glyphs with path and node location information, adjacency grouping assignment, slope (pictured below), and centroid locations, among other things.

We are currently working to incorporate the cluster grouping assignments into the package. This will be complete pending creation of a template.

Submitted to the Annals of Applied Statistics

A CLUSTERING METHOD FOR GRAPHICAL HANDWRITING COMPONENTS AND STATISTICAL WRITERSHIP ANALYSIS[†]

BY NICHOLAS S. BERRY^{‡§}, AMY M. CRAWFORD [‡]

Iowa State University[‡] and Berry Consultants, LLC[§]

4.3 Statistical Analysis

4.3.1 Clustering

Background to be added here.

Paper submitted!

...

4.3.2 Closed set modelling

The following will be the jumping off point for purposes of this book with respect to modelling and discussion of results.

4.4 Communication of Results

Presenting/corresponding author is in bold.

4.4.1 Papers

- “A Clustering Method for Graphical Handwriting Components and Statistical Writership Analysis”
 - Authors: Nick Berry, Amy Crawford
 - Submitted to The Annals of Applied Statistics in September 2019.
- “Handwriting 2”
 - Authors: Amy Crawford, Alicia Carriquiry, and Danica Ommen
 - In preparation for submission to PNAS

4.4.2 Talks

- “TITLE”
 - August 2019
 - Authors: Amy Crawford, Alicia Carriquiry, Danica Ommen
 - American Society of Questioned Document Examiners (ASQDE) Annual Meeting in Cary, NC.
 - Talk, 80 minutes.
- “TITLE”
 - July 2019



A Bayesian Hierarchical Model for Forensic Writer Identification

Amy Crawford, Alicia Carriquiry, PhD, and Danica Ommen, PhD
Iowa State University

1. Objective & Data	2. Glyph Grouping	3. Hierarchical Model	4. Writer Identification Results										
Objective Use automated and statistical methods to make probabilistic statements about a scanned handwritten document of unknown source with respect to unknowns from writers in a known, closed set.	Glyphs Glyphs are sorted into buckets via two methods. <ul style="list-style-type: none"> The rate at which a writer emits glyphs to each bucket can be used to characterize them as a writer. Method #1 – Graph Adjacency, Deterministic Each unique adjacency matrix is a bucket. Very rare buckets are considered together as one.	Let $\mathbf{Y}_{w D} = (Y_{w,1}, Y_{w,2}, \dots, Y_{w,n_w})$ be the number of glyphs assigned to each group, for document within writer, each document. $\mathbf{Y}_{w D} \sim Multinomial(\boldsymbol{\pi}_w)$ Method #2 – Clustering, Flexible & Robust A K-means type algorithm to create 40 glyph clusters that are more robust to small incidental strokes.	Log Loss measures model performance for each method. Method #2 Results: 										
Data Computer Vision Lab (CVL) Database ^[1] <ul style="list-style-type: none"> 27 writers 5 training documents, 1 holdout for each Process with the <i>handwriter R</i> package^[2] Decompose a document into glyphs <p>Figure: Processed writing. Red dots mark notes. Colors mark glyphs.</p>	<p>Figure: Three glyphs that are all the letter 'f', corresponding adjacency matrix, and label.</p>	<p>Figure: Three glyphs that are all the letter 'f', grouped together under the clustering method. The third is in a different group.</p>	<p>Figure: Posterior predictive results Under Method #2. Each row is a questioned document, containing the true writer and the posterior probability of authorship. Each row stands alone and sums to 1.</p>										
5. Writing Style Analysis	6. Work in Progress: Writer Variability Index												
The mixing parameter for each writer, ρ_w , gives insight into writing styles. <ul style="list-style-type: none"> Writers with low (blue) average values of ρ_w tend to have more connected, cursive writing. High (red) average values tend to indicate simple print writers. Writers show separation in their glyph emission rates (plot below for 3 of 40 groups). We see patterns for similar ρ_w values. <p>Figure (Top): Average Mixing Parameter (ρ_w) vs Writer ID. Points are color-coded by ρ_w value, ranging from blue (low) to red (high).</p> <p>Figure (Bottom): Average Mixing Parameter (ρ_w) vs Writer ID for writers 12, 16, 25, 26, and 29. Points are color-coded by ρ_w value, showing separation between writers.</p>	With count data models, it is important to investigate the presence of over- and under-dispersion. This can also be used as an index of writer variability. <ul style="list-style-type: none"> Multivariate Relative Dispersion Index (RDI) of Kokonendji and Puig, PI $RDI = \frac{\text{writer data GDI}}{\text{model simulated data GDI}}$, GDI is Generalized Dispersion Index. <table border="1"> <thead> <tr> <th>Writer</th> <th>12</th> <th>16</th> <th>25</th> <th>26</th> <th>29</th> </tr> </thead> <tbody> <tr> <td>RDI</td> <td>0.732</td> <td>0.793</td> <td>1.346</td> <td>1.963</td> <td>2.177</td> </tr> </tbody> </table> <p>Table: RDI for a few writers. Larger values indicate over-dispersion and that the writer has more sample to sample variability.</p>	Writer	12	16	25	26	29	RDI	0.732	0.793	1.346	1.963	2.177
Writer	12	16	25	26	29								
RDI	0.732	0.793	1.346	1.963	2.177								
7. References													
<p>[1] Alvaro, J., Puri, B., Tann, M. and Saito, H. (2010). CVL Database: An Offline Database for Writer Retrieval. <i>Workshop on Handwriting Recognition, On the Track of a Conference on Document Analysis and Recognition (ICDAR) 2010</i>. 560-564.</p> <p>[2] Puig, P. and Carriquiry, A. (2018). Handwriter: An R package for handwriting analysis.</p> <p>[3] Kokonendji, C. C. and Puig, P. (2016). Multivariate Count Distributions. <i>Journal of Multivariate Analysis</i>, 147, 140-160.</p>													

This work was partially funded by the Center for Statistics and Applications in Forensic Evidence (CSAFE) through Cooperative Agreement #70NANB15H176 between NIST and Iowa State University, which involve activities carried out at Carnegie Mellon University, University of California Irvine, and University of Virginia.

Figure 4.3: Poster given at the 10th International Workshop on Simulation and Statistics

- Authors: Amy Crawford, Nick Berry, Alicia Carriquiry, Danica Ommen
- Joint Statistical Meetings (JSM) in Denver, CO.
- Talk, 15 minutes.
- “TITLE”
 - July 2019
 - Authors: Alicia Carriquiry, Amy Crawford, Nick Berry, Danica Ommen
 - Lima, Peru.
 - Talk
- “TITLE”
 - February 2019
 - Authors: Amy Crawford, Nick Berry, Alicia Carriquiry, Danica Ommen
 - American Academy of Forensic Sciences (AAFS) Annual Meeting in Baltimore, MD.
 - Talk, 20 minutes
- “TITLE”
 - August 2018
 - Authors: Amy Crawford, Nick Berry, Alicia Carriquiry, Danica Ommen
 - American Society of Questioned Document Examiners (ASQDE) Annual Meeting in Park City, UT.
 - Talk, 20 minutes.
- “TITLE”
 - July 2018
 - Authors: Amy Crawford, Nick Berry, Alicia Carriquiry, Danica Ommen
 - Joint Statistical Meetings (JSM) in Vancouver, BC, Canada.
 - Talk, 15 minutes.
- “TITLE”
 - May 2018
 - Authors: Amy Crawford, Nick Berry, Alicia Carriquiry
 - American Bar Association, 9th Annual Prescription for Criminal Justice Forensics Program in New York, NY.
 - Talk, 15 minutes.

4.4.3 Posters

- “TITLE”
 - August 2019
 - Authors: Amy Crawford, Alicia Carriquiry, Danica Ommen
 - 10th International Workshop on Statistics and Simulation in Salzburg, Austria
 - 1st Springer Poster Award
- “TITLE”

- February 2018
- Authors: Amy Crawford, Nick Berry, Alicia Carriquiry
- American Academy of Forensic Sciences in Seattle, WA
- YFSF Best Poster Award
- “TITLE”
 - May 2018 and 2019
 - Authors: Amy Crawford, Nick Berry, Alicia Carriquiry, Danica Ommen
 - CSAFE Annual All-Hands Meetin in Ames, IA

4.5 People involved

4.5.1 Faculty

- Alicia Carriquiry
- Danica Ommen
- Hal Stern (UCI, Project G PI)

4.5.2 Graduate Students

- Amy Crawford

4.5.3 Undergraduates

- Anyesha Rey (data collection)

Chapter 5

Glass

Chapter 6

Shoes

6.1 Longitudinal Shoe Study

Github repository

6.1.1 Paper describing the database

Paper subdirectory of Github repository

Goal:

- Describe experiment
- Describe database function
- Publicize data for analysis by others in the community

Methods and Data Description

Methods and data description handed off to Alicia for editing

Data Analysis Tools

- Working with the EBImage package - very fast processing of images

6.1.1.0.1 Film and Powder Images

Analysis Summary: Create a mask via thresholding, clean it up, fill in mask holes, creating a shoe “region” mask. Apply this mask to the image, replacing any pixels outside the mask with the median background pixel. Additional thresholding and normalization can be applied if a binary image is more desirable.

1. Create threshold mask

- a. Blur image (circular/gaussian blur, diameter 15)
- b. Invert the image
- c. Threshold image (adaptive threshold, 10 x 10 region, keep anything with an average higher than 0.025 from the mean)
- d. Create mask

Default parameters selected by visually screening several shoes: (default parameters rad1 = 5, rad2 = 91, proportion = 1.5*area of rad2 in px/area of image in px)

 1. erode mask image (circle, diameter rad1)
 2. dilate mask image (circle, diameter rad2)
 3. label disjoint regions of the image
 4. prune small image regions (area < proportion parameter)
- e. Fill in mask holes
- f. Expand mask to capture entire shoe region
 1. set background color
 2. create dataframe of useful (non-background) pixels
 3. fill in holes and concave regions in mask, then expand by expand_rad vertically and horizontally (similar to “convex hull”, but with additional expansion radius)
- 2. Mask image to remove extra variability unrelated to the shoe
- 3. Threshold masked image?
Con: Lose grey information; Pro: fully remove background
- 4. Compromise: Keep grey pixels from thresholded, masked image (e.g. use
 3. as a mask), then renormalize

I've added the functions from last week to the `ShoeScrubR` package, which will hopefully contain methods for handling all of the different 2D shoe data from the longitudinal study.

Using that package, I tried the method out on a sequence of shoes over time to see what methods might best show wear. Each column shows a single left shoe over four timepoints. The shoes are the first 9 shoeIDs (e.g. 1 - 9).

Original

Cleaned

Cleaned and Thresholded

Even with the cleaning methods... there is a lot of extra noise.

Next step: templating!

Basic framework:

Create a template for each size and model combination (using GIMP - if I could automate this, I wouldn't need the template)

Intelligently brute force angle and position of template

Goal: Maximize the number of black pixels in the image within the template region

1. Start with an image and a template mask
2. Blur, normalize, invert, and threshold the image
3. Naively align the "centers" of the two images (avg of white pixel row/cols). To make this calculation comparable, do some very crude dilation/erosion (that may or may not generalize that well) to fill in the image a bit.

Then make the aligned center the actual center of the image via padding. (This is the 1st time we have modified the actual image beyond thresholding and color changes).

4. ~~Create a new mask to sample the image (and the mask) radially. ~~ This doesn't work when the object isn't a solid entity :(

New Option: Use image pyramids and brute-force alignment, starting off with an estimated rotation angle of θ from principal components

5. Brute force full-size image to get finer alignment.
6. Remove anything not in the mask region.

6.1.1.0.2 Wear Characterization

Ideas:

- average intensity of cleaned image
- length of border/edges detected

6.2 Passive Shoe Recognition

6.2.1 NIJ Grant

Grant scope: Build the shoe scanner, develop an automatic recognition algorithm for geometric design elements, test the scanner in locations around Ames.

Status: Funded! Next challenge: Figuring out how to transfer it to UNL.

6.2.2 CoNNOR: Convolutional Neural Network for Out-sole Recognition

Project Overview

- Label images of shoes according to geometric classification scheme
- Use convolutional base of pretrained CNN VGG16 and train a new classifier on labeled features
- Eventually, acquire real data passively and use CoNNOR to assess feature similarities and frequencies

[Link to submitted Creative Component on CoNNOR](#)

[Github repository for paper submitted to Forensic Science International](#)

Exploring new directions:

- Truncate convolutional base and train random forest on features
 - Could replace fully connected layers of neural net as classifier
 - Importance score can filter/reduce the number of features
 - *Block 4 random forest training terminated after one week :(*
 - *Block 5 currently training for two different random forest packages (randomForest and ranger)*
 - *If new models take more than 1-2 weeks, will look into subsampling techniques.*
- *Spatial integration*
 - *Model is currently set up to take in 256x256 pixels*
 - *Try taking in full shoe using a sliding window of size 256x256*
 - *View class predictions spatially*
- Fully convolutional networks (FCNs)
 - Unsupervised segmentation to assess current classification scheme
 - Handle whole shoe image of any size (instead of only 256x256 pixel images)

References for CNNs and FCNs

[Stack Exchange post explaining patchwise training](#)

[“Learning Hierarchical Features for Scene Labeling”](#): describes an application of multi-scale CNNs and image pyramids

[“Pyramid methods in image processing”](#): classic paper from 1984 explaining pyramid methods

[“Fully Convolutional Networks for Semantic Segmentation”](#)

[“W-Net: A Deep Model for Fully Unsupervised Image Segmentation”](#)