

This is us: making CSAFE stronger each
week

CSAFE

2019-09-12

Contents

1	Prerequisites	5
2	Introduction	7
3	Project CC: Bullets and Cartridge Cases	9
3.1	Data Collection	9
3.2	Computational Tools	13
3.3	Similarity Scores	14
3.4	Analysis of Results	24
3.5	Communication of Results and Methods	24
3.6	People involved	27
4	Handwriting	29
5	Glass	31
6	Shoes	33
6.1	Longitudinal Shoe Study	33
6.2	Passive Shoe Recognition	35
6.3	Maximum Clique Matching	36
6.4	Project Tread (formerly Cocoa Powder Citizen Science)	36
6.5	3d Shoe Recognition	37
7	Theoretical foundations	39
7.1	Nate's Updates 9/5/2019	39
7.2	Nate's Updates 9/12/2019	40
8	Outreach activities	41
8.1	Book on Forensic Science and Statistics	41

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 2. If you do not manually label them, there will be automatic labels anyway, e.g., Chapter 5.

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 2.1. Similarly, you can reference tables generated from `knitr::kable()`, e.g., see Table 2.1.

```
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 (Xie, 2019) in this sample book, which was built on top of R Markdown and **knitr** (Xie, 2015).



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.

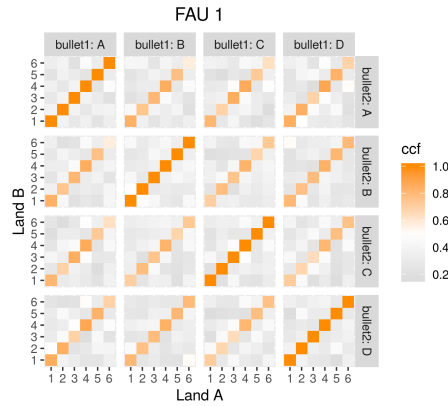


Figure 3.1: Results from assessing scans of barrel FAU 1 similarity.

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

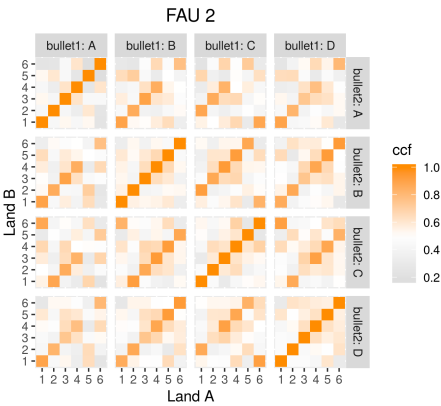


Figure 3.2: Results from assessing scans of barrel FAU 2 similarity.

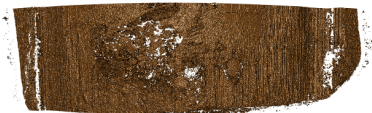


Figure 3.3: Land scan for barrel FAU 3 bullet A land 6.



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

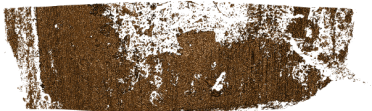


Figure 3.5: Land scan for barrel FAU 5 bullet B land 5.

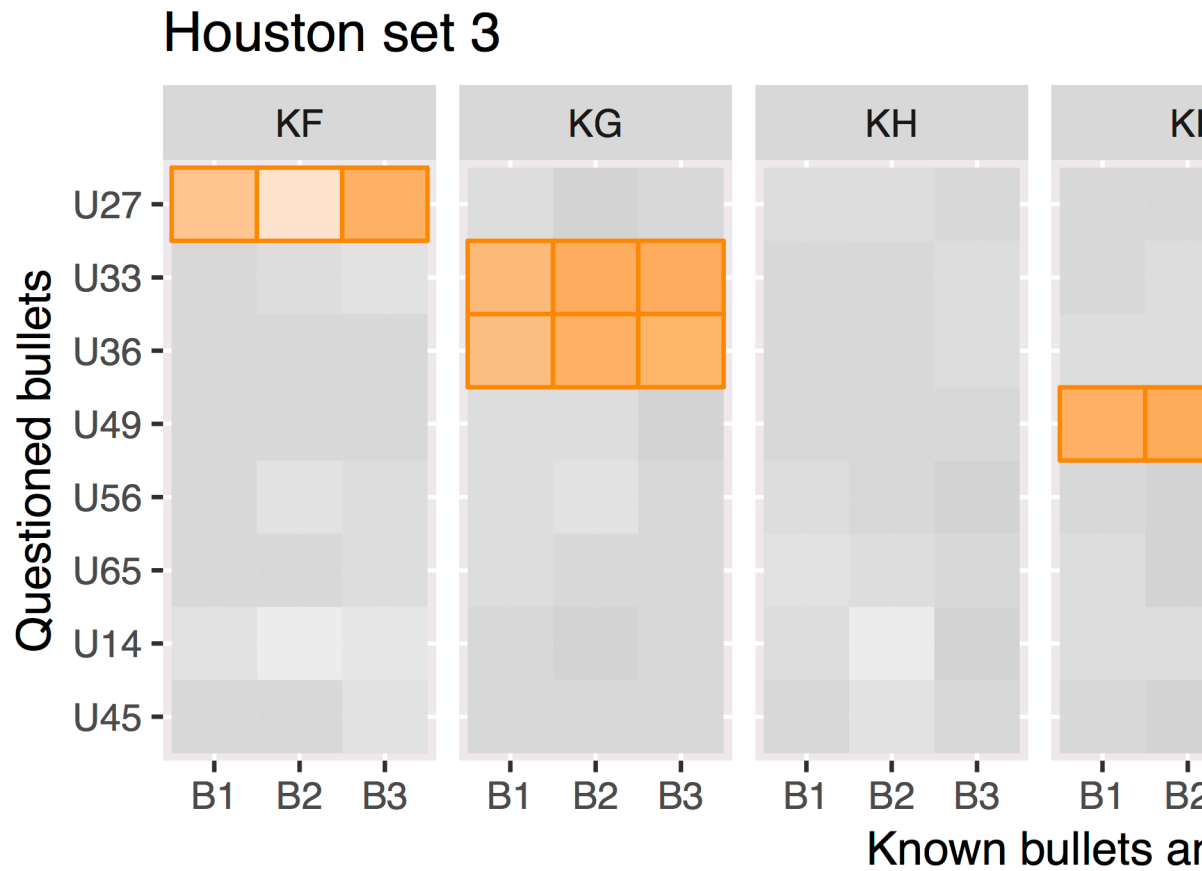


Figure 3.6: Bullet-to-bullet similarity scores for questioned bullets (y-axis) compared to all other bullets of the test set (x-axis).

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 Hare et al. (2016).

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/>

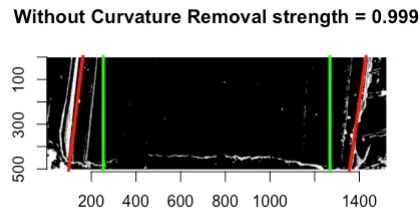


Figure 3.7: 2-dimensional visualization of example bullet br411 with .999 strength threshold

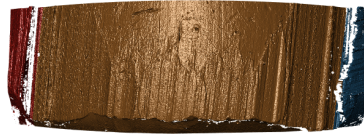


Figure 3.8: 3-dimensional visualization of example bullet br411 with .999 strength threshold

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

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

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.

3.3.2 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 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

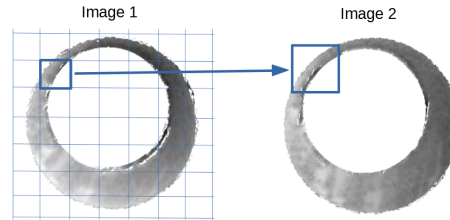


Figure 3.9: Comparing a cell in image 1 to a larger region in image 2. We wish to find the translations of the image 1 cell that yield the highest correlation within the image 2 region.

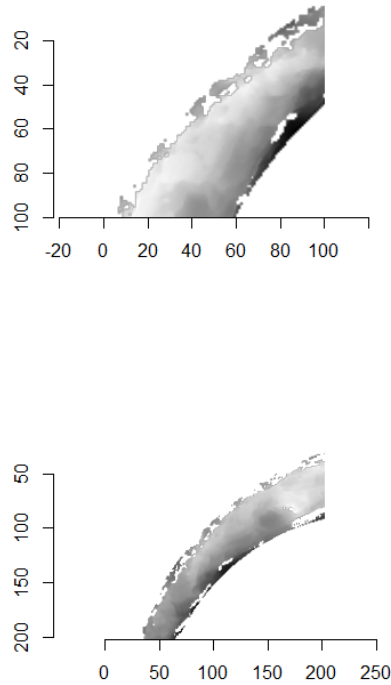


Figure 3.10: (Left) A cell from image 1. (Right) A region from image 2 centered in the same location as the image 1 cell, yet quadruple the area.

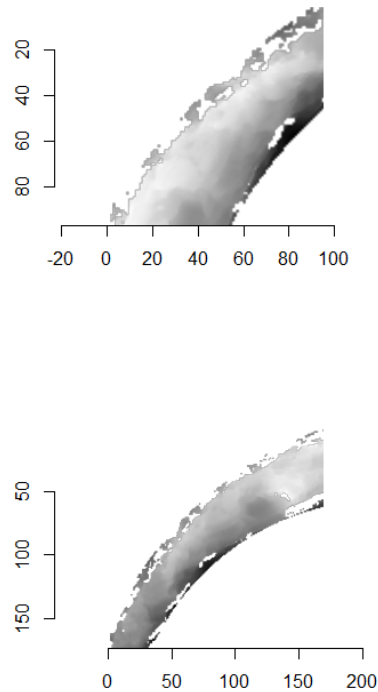


Figure 3.11: The same images as above after cropping NA rows/columns.

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 breachface 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 remaining 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

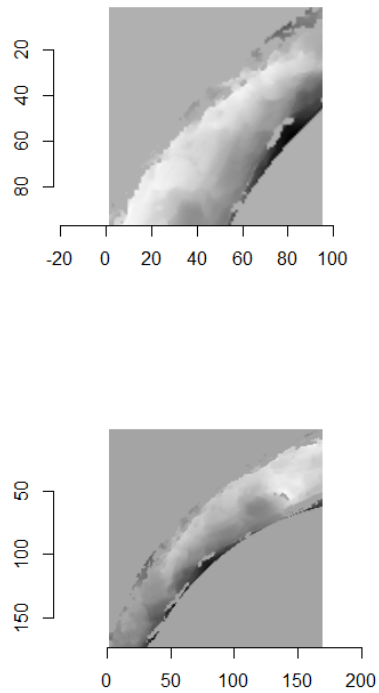


Figure 3.12: The NA-cropped images with remaining NAs replaced with the image's average pixel values.

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 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.

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 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

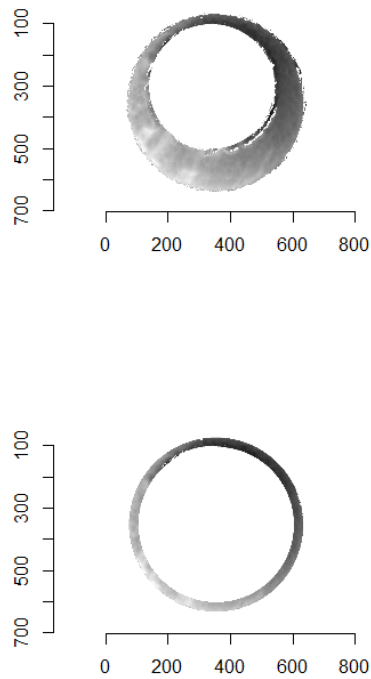


Figure 3.13: (Left) The original breech face impression scan image. (Right) A donut-shaped region cut out of the original image.

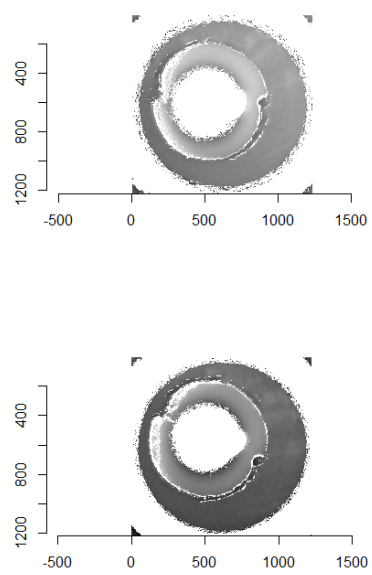


Figure 3.14: Two cartridge case scans before pre-processing.

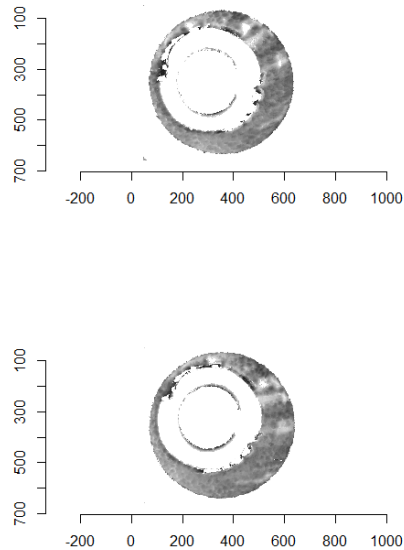


Figure 3.15: Residual values of a RANSAC plane fit to the two cartridge case scans shown above.

used in the previous section. However, we’ve found that the use of a Hough Transform doesn’t extract the “breach face signal” from an image as other techniques. Namely, the breach 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 breach face impression. Once we’ve fit this plane to the breach 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 breach 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 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 breach face scan) that just so happened to be close to the fitted plane and thus were

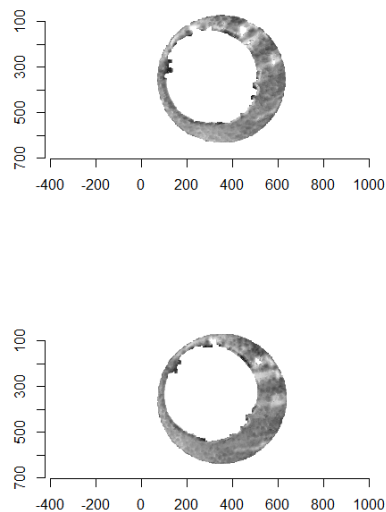


Figure 3.16: The selected breach face impressions based on dilation and erosion.

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 breach 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 breach face impression that we haven't yet identified in order to identify the breach 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 of the breach 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 breach face impressions. The location of the firing ring impression in the breach 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 that, at the very least, the breach face impression left on the cartridge case is horizontally/vertically aligned if not rotationally aligned.

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 asses, 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”

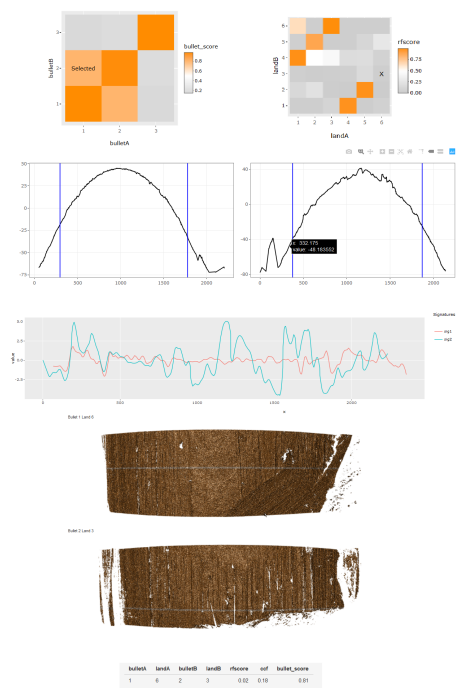


Figure 3.17: An instance of the interactive visualizations for communicating results

- 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

3.5.1.3 Joint Statistical Meetings

- “A non-parametric test for matching bullet striations: extending the chum-bley 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 Roigers
- Joe Zemmels
- Yawei Ge

3.6.3 Undergraduates

- Talen Fisher (fix3p)
- Andrew Maloney
- Mya Fisher, Allison Mark, Connor Hergenreter, Carley McConnell, Anyesha Ray (scanner)

Chapter 4

Handwriting

We describe our methods for going about the handwriting project here.

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.

Create multiple radial masks, each offset slightly; optimize over the intersection (# pixels or average pixel intensity) to get a coarse alignment angle.

5. Brute force for finer alignment.
6. Remove anything not in the mask region.

Modified from (Kim and de Araújo, 2007)

6.1.1.0.2 Wear Characterization

Ideas:

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

6.2 Passive Shoe Recognition

###CoNNOR: Convolutional Neural Network for Outsole 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”

6.3 Maximum Clique Matching

6.4 Project Tread (formerly Cocoa Powder Citizen Science)

Project Tread, modified from Leverhulme Institute’s Sole Searching, is a developing CSAFE project with the goals of engaging community participation in

forensic research and acquiring shoe print data that may be useful in future analyses.

In progress:

- Review procedures and IRB documents written by James
- Perhaps modify procedures, then bribe some friends into helping me test them :)
 - Test for length, clarity, ease, etc.
- Be involved in set up of data collection site (through CSSM)

6.5 3d Shoe Recognition

The set up

What we have tried

What we are doing now - Transforming the mesh objects to points aligned by the center of mass to overlay them detecting difference. - Problems - Isn't aligning properly as you can see - Next - angle transformations

Chapter 7

Theoretical foundations

7.1 Nate's Updates 9/5/2019

- Currently in Virginia
- RA for this semester (year?) is under Danica
- **Central Goals:**
 - continue work started by Danica and Peter Vergeer on the analysis of likelihood ratios
 - study the differences between specific source (SS) and common source (CS) likelihood ratios (LRs) in an information theoretic way
 - does the CS or SS LR have more “information”?
 - can be the CS or SS hypotheses (prosecution or defense) be formally compared in terms of being easier to “prove” or “disprove”?
- **Basic Setup**
 - H_p , H_d are CS prosecution and defense hypotheses upon which we will place priors
 - A and B are discrete r.v.'s representing two “sources” of evidence
 - * distributions for A and B defined conditionally based on the hypothesis
 - * SS hypothesis is represented by the conditional random variable $H_p|A$
 - X is data coming from A , Y is data coming from B
 - compare information contained in (X, Y) about H_p and $H_p|A$
 - * this is what I'm working on now
 - * important quantities:
 - Kullback-Leibler divergences, entropy, mutual information

7.2 Nate's Updates 9/12/2019

- Assume $p(X, Y, A, B, H) = p(X, Y|A, B)p(B|A, H)p(A|H)p(H)$
- Entropy: $E(X) = \int -p(x) \log p(x) dx$
 - “average amount of information gained through observation of $X = x$ ”
 - * low entropy \implies better able to predict outcome
 - * $\mathcal{E}(H|X, Y) \geq \mathcal{E}(H|X, Y, A)$
 - Common source (posterior) entropy is greater than specific source
 - Mutual Information: $\mathcal{I}(X; Y) = \mathcal{E}(X) - \mathcal{E}(X|Y)$
 - * $\mathcal{I}(X; Y) = KL(p(X, Y) || p(X), p(Y)) \geq 0$

Chapter 8

Outreach activities

CSAFE has several ongoing outreach projects.

8.1 Book on Forensic Science and Statistics

Project members:

- Kiegan Rice
- Alicia Carriquiry
- Hal Stern (UCI)

General book outline:

- Chapter 1: Introduction
- Chapter 2: Analysis of Forensic Evidence
 - Drafted. In the editing stage.
- Chapter 3: The Path to the Witness Stand
 - Drafted. In the editing stage.
- Chapter 4: Communicating Evidence in the Courtroom
 - Not drafted. In the writing stage.
- Chapter 5: Conclusions

Bibliography

- Hare, E., Hofmann, H., and Carriquiry, A. (2016). Automatic Matching of Bullet Lands. *ArXiv e-prints*.
- Kim, H. Y. and de Araújo, S. A. (2007). Grayscale Template-Matching Invariant to Rotation, Scale, Translation, Brightness and Contrast. In Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J. M., Mattern, F., Mitchell, J. C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., Vardi, M. Y., Weikum, G., Mery, D., and Rueda, L., editors, *Advances in Image and Video Technology*, volume 4872, pages 100–113. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Xie, Y. (2015). *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition. ISBN 978-1498716963.
- Xie, Y. (2019). *bookdown: Authoring Books and Technical Documents with R Markdown*. R package version 0.10.