1 Introduction

I do not know how to start introductions!!!!!!

2 Methods

2.1 Brief Overview of Hough Transforms

In broad terms, the Hough Transform is a feature extraction technique for detecting shapes in an image. For any given point in the (x,y)-plane of our image we can calculate a corresponding line in the feature space. Points that lie on the same line will have lines that intersect in the feature space.

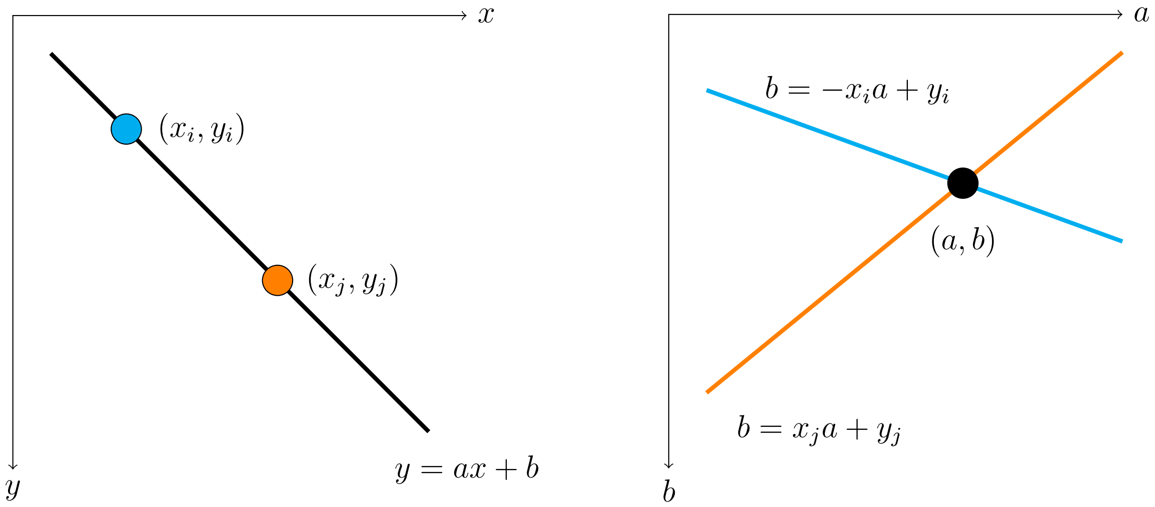
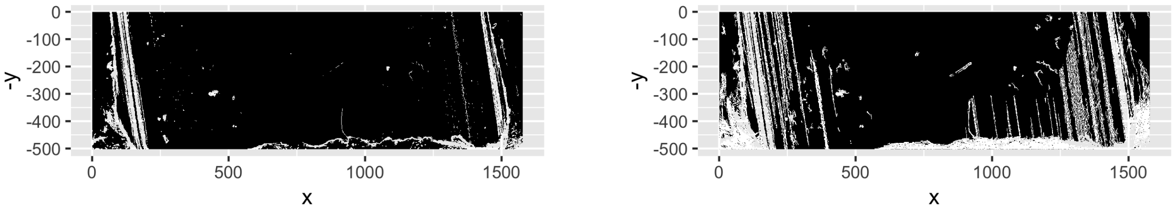


Figure 1: Diagram of feature space linea transformation oriented for image origin.

The point of intersection in the feature space, then corresponds to the parameters used to describe the edge in the (x,y)-plane that the two detected points lie upon. A two dimensional array called the ”accumulator array” is used to keep track of features detected. The accumulator array covers the entirety of the feature space separated into a user-specified number of bins. For each set of features detected the bin in the accumulator array associated with that set of features is incremented. So in theory, strong features should have higher values in their associated bin because there is a larger number of pixels detected that all have the same set of features in the feature space.

In order to best identify the GEAs we first want to diminish noise in the image. This can be achieved by converting each scan into an image gradient, which signifies where there are directional changes in the color of the image. This approach unforunately loses most of the detail of the three dimensional scans, however, it better highlights the differences between LEAs and GEAs. Once an image gradient is obtained we select only edges we consider to be ”strong”, meaning they have a magnitude above the 99th percentile. Our reason for not fully carrying out a Canny Edge detection algorithm before using a Hough transform is that the

Canny Edge algorithm increases processing time by about 35 seconds per image and actually highlights more striae and breakoff.



(a) Edges with magnitudes in the 99th percentile (b) Edges improved with Canny edge detection

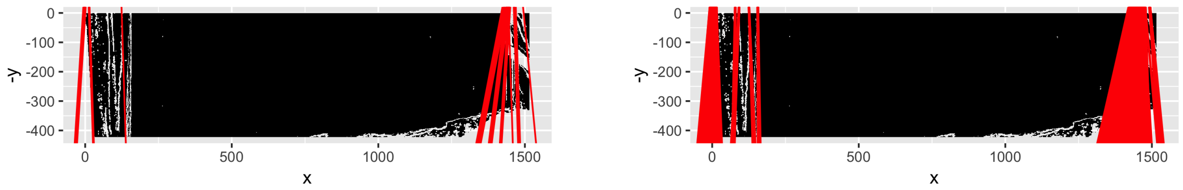
Figure 2: Side-by-side comparison of Hamby 252 Bullet using both magnitude thresholding and Canny edge detection

As shown in Figure 2 the striae in the LEA are much more pronunced than in the image with only strong edges. We wish to focus on only detecting GEAs, so detecting an increased number of striae through Canny edge detection is not useful for our algorithm. Once we obtain the image gradient with only the strong edges we can then utilize our Hough transformation to obtain generally reasonable estimates of image boundaries. For the Hough transformation we utilize the function “ `hough_lines`” from the imager package with the number of bins set to 100. While changing the number of bins does not seem to effect the processing time of the Hough transform, having a larger number of bins increases the number of extraneous detected lines in the image. For example in Figure 3a we can see that the Hough transforms with 100 bins does a perfectly aedequate job of picking up the suspected grooves of the bullet. Therefore the extra detected lines in 3b are not used in our analysis so we chose to limit our bin number as a result.

We make the assumption that scans are oriented properly and as such, most Hough lines that correlate to the GEAs will be roughly vertical with some deviations in angle based on scanning technique. It is worth noting that the original output of the “ `hough_lines`” functions have angles ranging from 0 to 2π. So to make selecting lines from our desired range, easier we chose to transform any angle greater than π, to instead be from 0 to $-\pi$ by subtracting 2π from the original theta angle. Therefore we select only the Hough lines that have theta angles from the positive x-axis less than $\frac{\pi}{16}$ and greater than $\frac{-\pi}{16}$. The parametrization produced by “ `hough_lines`” is of the form:

$$\rho = x\ cos(\theta)\ +\ y\ sin(\theta)$$

Where ρ represents the length of the normalized orthogonal vector between the detected line and the origin of the image, and θ is the angle between the positive x-axis and the normalized vector. We note that the format of bullet images is slightly different from standard x-y-plots. In two-dimensional representations of bullet images, the y-axis is inverted from what we



(a) Hough Transform with 100 bins

(b) Hough Transform with 900 bins

Figure 3: Side-by-side comparison of Houston Barrel F Land 1 Hough Transformation. Hough lines are filtered by having scores in the 99.9th percentile, and having central angles less than $\frac{\pi}{16}$.

would expect. A reason for this is that the images are processed in **C++** which stores images, known as "CImgs" in a vector format where the first pixel in the vector corresponds to the upper lefthand pixel of the image located at (0,0). Pixels are then ordered from left to right, extending from the origin to the positive x-direction and from top to bottom, from the x-axis towards the negative y-direction. In Figure 4 we see an example of how Hough transforms parametrize a line. In this figure, the orange line represents an example of a detected Hough line. The teal line connecting the origin to the orthogonal bisector of our detected Hough line represents the vector denoted ρ in our above equation. Similarly the teal arc below represents the angle, θ, which is the difference between the top of the image and the orthogonal bisector.

We note that when θ is equal to 0, the x-intercept is given by ρ however when θ is not 0 we utilize the above equation to find that the x-intercept is equivalent to $\frac{\rho}{\cos(\theta)}$. We utilize this calculation to find where the estimated Hough line intersects the top and bottom of the bullet land. The top is the x-intercept calculated previously, however to calculate the bottom intersection of the bullet land we utilize some geometric properties. We can draw a perpendicular line from the x-intercept at the top of the land to the bottom of the land creating a right triangle as shown below in 5. Since both the angle between the orthogonal bisector labeled ρ and the new triangle created from the xintercept to the bottom of the land are both right triangles, we know that the angle at the top of the newly formed triangle is equivalent to θ. Since the x-intercept is known, we can use an elementary geometry technique to calculate the distance labelled δ in the diagram below. Since $\tan(\theta)$ is equivalent to the proportion of the length of the "opposite" side over the "adjacent" side of the right triangle, we know that $\delta = \tan(\theta) * \text{height}$. So the index of the bottom intersection of the bullet land is equivalent to top-intersection - δ.

The utility for calculating the top and bottom intercept of our bullet land is that it allows us to calculate the slope of each Hough line with respect to the y-direction.

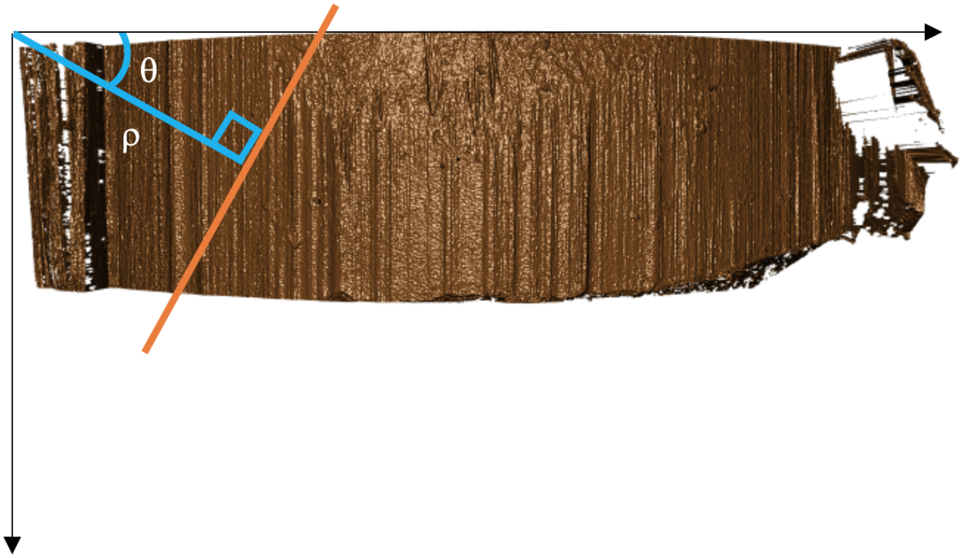


Figure 4: Example of Hessian Normal Form parametrization overlaid a bullet scan

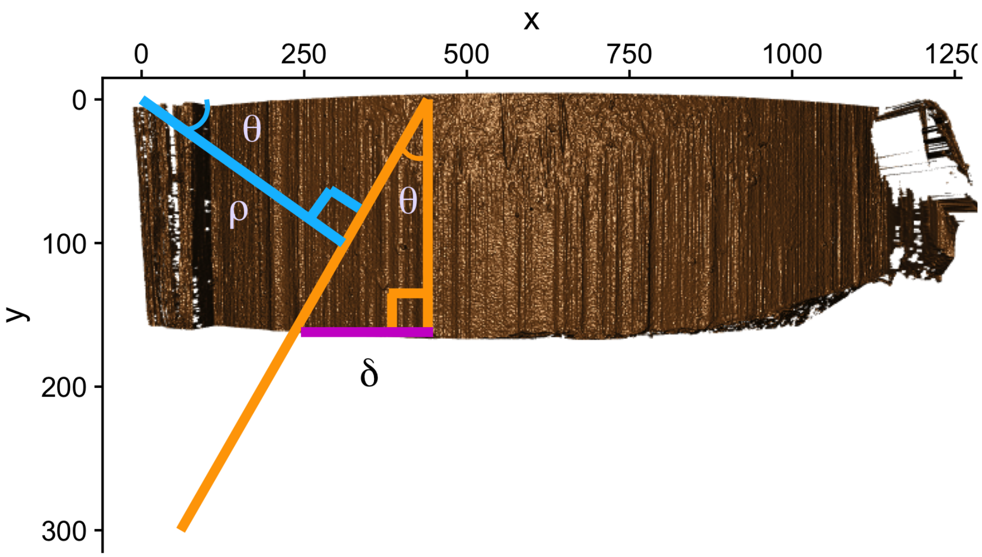


Figure 5: Demonstration of calculation of bottom intercept of a bullet land using SOH-CAH-TOA

$$\text{Slope with respect to y} = \frac{\text{(x-intercept of the top}-\text{x-intercept of the bottom)}}{\text{height of the bullet land}}$$

Traditionally slope with respect to the x-direction is chosen for describing the equation of a line, however, because we are primarily interested in vertical lines the slope with respect to the x-direction tends to infinity. This is undesireable for a number of reasons, and so we have decided to use the slope with respect to y because it is more robust for describing Hough lines.

As discussed before, the Hough transform outputs a sort of score that correpsonds to the number of pixels detected on a line that can give us an indication of the strength of the feature detected. Theoretically, we would expect the strongest lines in our image to be the grooves on either side of the bullet land. So the problem arises of how best to select strong edges. Rather than simply filter scores on some arbitrary threshold, we normalize the Hough scores by the largest possible score that could be achieved for each set of features detected. The reason for this is that longer lines will have a larger Hough score, simply by virtue of having a larger number of possible detectable pixels.

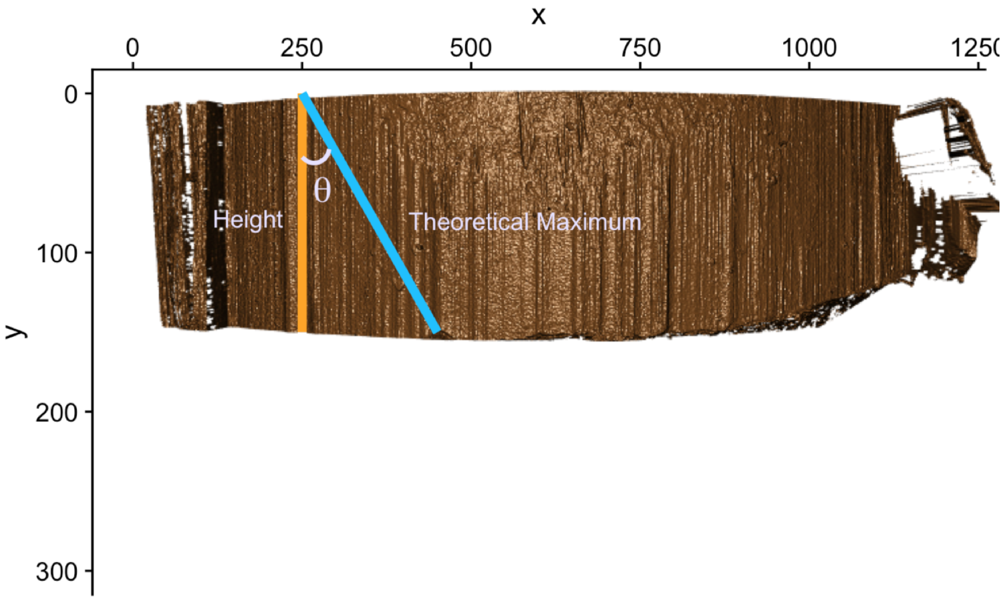


Figure 6: Demonstration of calculation of theoretical maximum Hough score using SOH-CAH-TOA

We calculate the largest possible theoretical score by dividing the height of the image by $\cos(\theta)$. This should in theory give us the total possible pixels that the Hough transform could have detected in our image by exploiting geometric properties of the right triangle shown in figure 6. Then for each unique set of features that describe a line that the Hough transform detects we divide the score associated with these features by the theoretical maximum score, yielding the normalized score.

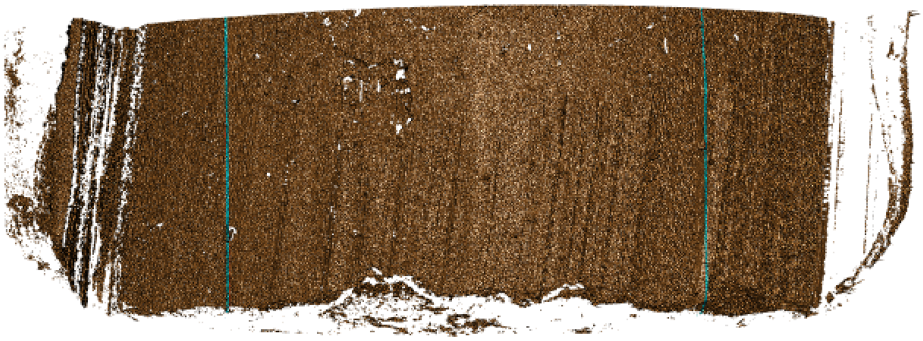


Figure 7: Middle fifty percent of the bullet land marked by two cyan coloured vertical lines imposed over the Phoenix set Gun 1-M2 Bullet 1 Land 1 scan

To further specify the best candidates for the bullet land grooves, we rely on the heuristic that most of the middle 50% of the bullet will be occupied by striae. Therefore, we can eliminate any strong lines detected within this region. An example of the middle fifty percent of a bullet land can be seen in figure 7 where the middle fifty percent of the bullet land is bordered by cyan coloured vertical lines. We note that the grooves are well away from this middle fifty percent region, making this a suitable heuristic. We then select the highest normalized score of the lines detected outside of the middle fifty percent of the bullet land. We claim then that this is our detected bullet groove. If on the off chance that the strongest feature detected is within the middle 50% area of the bullet land, then the grooves are set to be the borders of the middle 50% area.

Once features are chosen to describe both the left and right hand grooves of the bullet land the ‘get_grooves_hough‘ function translates these features into two equations for lines. It is worthwhile to note at this point that our calculations have been in terms of pixels detected in an image rather than the microns of the bullet scan. So we must transform our equations of lines from being in terms of pixels to microns. This is accomplished inside the ‘get_grooves_hough‘ function by the helper function ‘pix_to_micron‘ which subtracts one from the pixel location (since pixels start indexed at one and microns start indexed at zero), then multiplies that pixel location by the scale of the x3p-scan. Because we are interested in the slope with respect to the y-direction we calculate our linear equations as follows:

$$\text{Groove Estimate} = \text{Bottom x-intercept in microns} + \text{slope} * y_i - \text{adjust}$$