Automated Target Analysis

By Casey Hartnett

Practicum for Master's Degree in Mathematical Statistics

Advisors: Dr. John Alford and Dr. Edward Swim

Spring 2014

Table of Contents

1. Motivation

Shooting sports have 15 events in the Summer Olympics, 11 events in the Winter Olympics, and thousands of professional tournaments with between 20 and 40 million annual participants in the United States. All of these events center around a key shooting characteristics; accuracy. Current methods of analyzing shooting targets are primitive and limited to the following;

- Tournaments use a point system instead of directly measuring accuracy. The closer to the center of the target the more points are rewarded. A simple fix, but this process loses the continuity that truly exists and replaces it with 11 discrete possibilities, as shown in Figure 1.

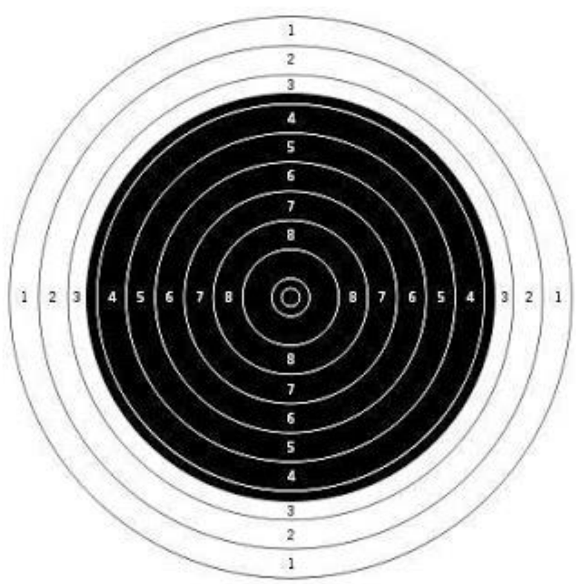

Figure 1. Olympics 10m Target

- The simplest is the grouping measurement, which is the distance between the two furthest apart holes. This is very easy to calculate and highly used. Yet, is not a good measure because it is high variability and dependent on the number of shots on target.

- OnTarget is a software that allows for two options for target analysis. A user may use any target and manually select each hole in the target. Alternatively, a user can use one of their provided targets which consists of 30 mini-targets each of which can only be shot once. The detected holes are then used to calculate a variety of measures. This software costs from \$12 to \$75, dependent on which features are purchased.

I believe that people can become better and enjoy shooting sports more with a simple to use software to analyze target accuracy. Shooters could perform experiments to determine which ammo, rifle or shooter may perform better. Tracking the measurements over time could provide positive feedback on improvement that would otherwise not be observed. Furthermore, professional shooters, military, and ammo manufacturers could apply the same techniques to improve or increase the efficiency of measuring shooting targets. All these industries still rely on non-automated methods for identifying the holes in targets for experiments.

2. Statistical Model

2.1 Model Description

To develop an understanding of the different measurement methods, a statistical model must be developed for the impact points on a target. Standard practice is to assume the impact point of a bullet follows a bivariate normal distribution.

Consider the bivariate normal distribution probability density function, given by

$$f(x,y) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} e^{-\frac{1}{2(1-\rho^2)}\left[\left(\frac{x-\mu_x}{\sigma_x}\right)^2 - 2\rho\left(\frac{x-\mu_x}{\sigma_x}\right)\left(\frac{y-\mu_y}{\sigma_y}\right) + \left(\frac{y-\mu_y}{\sigma_y}\right)^2\right]},$$

where x and y are the Cartesian coordinates of a bullet's impact with mean (μ_x, μ_y), the standard deviation of x is σ_x, the standard deviation of y is σ_y and correlation coefficient is ρ. The probability of a bullet impact over the region $a < x < b, c < y < d$ where $a, b, c, d \in \mathbb{R}$ is the integration of the bullet impact density function which given by

$$P(a < x < b, c < y < d) = \int_c^d \int_a^b f(x,y)dxdy.$$

We assume that $\rho = 0$ and $\sigma_x = \sigma_y$ in order to simplify the calculations. The bullet impact probability density function reduces to

$$f(x,y) = \frac{1}{2\pi{\sigma_x}^2} e^{-\frac{1}{2}\left[\left(\frac{x-\mu_x}{\sigma_x}\right)^2 + \left(\frac{y-\mu_y}{\sigma_x}\right)^2\right]}.$$

We apply the change of variables, $x^* = x - \mu_x$ and $y^* = y - \mu_y$. Therefore, the bullet impact probability density function can now be expressed as

$$f(x, y) = \frac{1}{2\pi{\sigma_x}^2} e^{-\frac{1}{2}\left[\left(\frac{x}{\sigma_x}\right)^2 + \left(\frac{y}{\sigma_x}\right)^2\right]}.$$

Polar coordinates further simplify the mathematics and interpretation. They use the polar transformation $r = \sqrt{x^2 + y^2}$ and $\theta = \tan^{-1}(y/x)$. This provides r the distance from the center to the point (x, y) and θ the associated angle. The bullet impact probability density function can now be expressed as

$$f(r, \theta) = \frac{r}{2\pi{\sigma_x}^2} e^{-\frac{1}{2}\left[\left(\frac{r}{\sigma_x}\right)^2\right]}.$$

The marginal probability allows us to understand a set of variables without association to another. The marginal probability density function of θ is found by integrating r over its domain as shown below. θ is non-essential in as shown by the probability, $h(\theta)$, being constant. This is a uniform probability function and provides no insight into shooting characteristics.

$$h(\theta) = \int_0^\infty f(r, \theta) dr = \frac{\pi}{2}$$

Similarly, the marginal distribution of r is found by integrating θ over its domain. The integral of $h(r)$ provides the probability of a bullet impact over the defined radius. $h(r)$ is the basis for future measures to be utilized.

$$h(r) = \int_0^{2\pi} f(r, \theta) d\theta = \frac{r}{{\sigma_x}^2} e^{-\frac{1}{2}\left[\left(\frac{r}{\sigma_x}\right)^2\right]}.$$

2.2 Derivation of μ and $\sigma_r{}^2$

With the bullet impact probability function in terms of the radius we can now investigate the mean bullet impact radius and the variance around the mean. In future sections, this will be utilized to create an estimate for shooting accuracy.

The mean r is found by taking the expected value of marginal probability distribution $h(r)$. The expected value of r is the integration of $r * h(r)$ over the domain of r. Let the mean of r to be denoted by μ.

$$\mu = E(r)$$

$$= \int_0^\infty r * h(r)dr$$

$$= \int_0^\infty r\left[\frac{r}{\sigma_x{}^2} e^{-\frac{1}{2}\left[\left(\frac{r}{\sigma_x}\right)^2\right]}\right]dr$$

$$= \int_0^\infty r^2\left[\frac{1}{\sigma_x{}^2} e^{-\left[\left(\frac{r}{\sqrt{2}\sigma_x}\right)^2\right]}\right]dr$$

The integral must now be set up for u substitution where $u = \frac{r}{\sqrt{2}\sigma_x}$ then $du = \frac{dr}{\sqrt{2}\sigma_x}$.

$$\mu = \int_0^\infty 2\sqrt{2}\sigma_x\left(\frac{r}{\sqrt{2}\sigma_x}\right)^2 e^{-\left[\left(\frac{r}{\sqrt{2}\sigma_x}\right)^2\right]}\left(\frac{dr}{\sqrt{2}\sigma_x}\right)$$

Now let $u = \frac{r}{\sqrt{2}\sigma_x}$ then $du = \frac{dr}{\sqrt{2}\sigma_x}$ to yield

$$\mu = 2\sqrt{2}\sigma_x\int_0^\infty u^2 e^{-u^2}du$$

$$= 2\sqrt{2}\sigma_x\left[\frac{\sqrt{\pi}}{4}\right].$$

Therefore, we find the mean of the bullet impact probability density function given by

$$\mu = \sqrt{\frac{\pi}{2}}\sigma_x \approx 1.25\sigma_x.$$

We found the mean radius of bullet impact. The variation around the mean is the variance. This informs us of about the consistency of the accuracy. Let the variance of r be denoted by ${\sigma_r}^2$. Then the formula for the variance is given by ${\sigma_r}^2 = E(r^2) - [E(r)]^2$. We have already found $E(r)$. Therefore we will now will find $E(r^2)$.

$$E(r^2) = \int_0^\infty r^2 * h(r)dr$$

$$= \int_0^\infty r^2 \left[\frac{r}{{\sigma_x}^2} e^{-\frac{1}{2}\left[\left(\frac{r}{\sigma_x}\right)^2\right]}\right] dr$$

$$= \int_0^\infty (4{\sigma_x}^2) \left(\frac{r}{\sqrt{2}\sigma_x}\right)^3 e^{-\left[\left(\frac{r}{\sqrt{2}\sigma_x}\right)^2\right]} \left(\frac{dr}{\sqrt{2}\sigma_x}\right)$$

The integral is now setup for a u substitution integration. Let $u = \frac{r}{\sqrt{2}\sigma_x}$ then $du = \frac{dr}{\sqrt{2}\sigma_x}$ thus

$$E(r^2) = 4{\sigma_x}^2 \int_0^\infty u^3 e^{-u^2} du$$

$$= 4{\sigma_x}^2 * \frac{1}{2}$$

$$= 2{\sigma_x}^2.$$

Therefore, the variance is

$${\sigma_r}^2 = E(r^2) - [E(r)]^2$$

$$= 2{\sigma_x}^2 - \left[\sqrt{\frac{\pi}{2}}\sigma_x\right]^2$$

$$= \left(2 - \frac{\pi}{2}\right){\sigma_x}^2 \approx .429{\sigma_x}^2 .$$

2.3 MLE Estimation

The mean and variance are theoretical values. Therefore, we need to find estimates for the theoretical quantities μ and σ_x. We found that the mean of the bullet impact radii, μ, is just a scalar of σ_x. Furthermore, the variance of the bullet impact radii, ${\sigma_r}^2$, is also composted of σ_x. Therefore, by the invariance properties of maximum likelihood method we can find the estimates for μ and σ_r by finding the maximum likelihood estimate of σ_x.

We start by considering n identically and independently drawn samples from the probability density function $h(r)$. Then, the log likelihood function is given by

$$l({\sigma_x}^2; r_1, r_2, \dots, r_n) = \sum_{i=1}^{n} ln[\,h(r_i)]$$

$$= \sum_{i=1}^{n} ln\left[\frac{r_i}{{\sigma_x}^2} e^{-\frac{{r_i}^2}{2{\sigma_x}^2}}\right]$$

$$= \sum_{i=1}^{n} \left[\ln(r_i) + \ln\left(\frac{1}{{\sigma_x}^2}\right) - \frac{{r_i}^2}{2{\sigma_x}^2}\right]$$

The next step is to maximize the log likelihood function to find the MLE estimate. We start by taking the first derivative with respect to ${\sigma_x}^2$ and setting to 0.

$$\frac{d}{d\widehat{{\sigma_x}^2}} l\left(\widehat{{\sigma_x}^2}; r_1, r_2, \dots, r_n\right) = 0$$

$$\Rightarrow \sum_{i=1}^{n} \left[-\frac{1}{\widehat{{\sigma_x}^2}} + \frac{{r_i}^2}{2\left(\widehat{{\sigma_x}^2}\right)^2}\right] = 0$$

We must rearrange the equation to isolate $\widehat{{\sigma_x}^2}$ the MLE estimate.

$$\Rightarrow \frac{n}{\widehat{{\sigma_x}^2}} = \frac{\sum_{i=1}^{n} {r_i}^2}{2\left(\widehat{{\sigma_x}^2}\right)^2}$$

$$\Rightarrow \widehat{{\sigma_x}^2} = \frac{\sum_{i=1}^{n} {r_i}^2}{2n}$$

Now, in terms of the original Cartesian coordinates we have $r_i = \sqrt{(x_i - \overline{x})^2 + (y_i - \overline{y})^2}$. Thus, the MLE estimate for σ_x is given by

$$\widehat{\sigma_x} = \sqrt{\frac{\sum_{i=1}^{n}[(x_i-\overline{x})^2+(y_i-\overline{y})^2]}{2n}}.$$

2.4 Grouping Measurement

The most common method of measurement in shooting is referred to as grouping. This is calculated by taking the measurement between the centers of the two furthest apart holes in a set of shots. This method is chosen because it is incredibly easy to measure by hand. This is because it only uses one measurement between two obvious points. Other methods will require every hole to be measured which would significantly more time and depending on the measurement method more difficult.

For a set of n shots with centers (x_i, y_i) for $i \in (1,2,\dots,n)$ the grouping measurement is defined to be

$$g(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_n) = \max\left\{\sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2}\right\}$$

for every i and $j \in (1,2,\dots,n)$.

Unfortunately, development of the theoretical statistical model for the grouping measurement cannot be completed because it does not follow any known distribution. Instead we will continue to understand the measurement using simulations. A simulation is devised to understand the distribution of grouping measurements over varying sample sizes.

We start by taking a sample from a multivariate normal distribution. So let $(x_i, y_i) \sim iid\ MVN\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}150 & 0\\0 & 150\end{bmatrix}\right)$ for $i \in (1,\dots,n)$.

The grouping measurement, g, is found for the sample by using the definition. This process is then repeated 20,000 times and for $n \in (2,3,\dots,100)$. Therefore, we have 20,000 simulations of the grouping measurement for the exact same parameter values. We then define the mean and variance of the grouping measurement to understand the underlying distribution.

$\overline{x}_g(n) = \overline{g(n)}$ is the sample mean of the grouping measurement for a sample size of n.

$S_g^2(n) = \frac{\sum(g(n)-\overline{g(n)})^2}{n-1}$ is the sample variance of the grouping measurement for a sample size of n.

Below is a plot of the mean grouping measurement for the simulation data, Figure. 2. It clearly illustrates a key problem of the grouping measurement. The measurement is dependent on sample size. This characteristic makes it a poor measure because values from one sample size cannot be directly compared to another sample size. For example, a shooter cannot compare a measurement with five shots to a measurement with 10 shots. Furthermore, the measurement changes more extreme where most shooters shoot, where n is small.

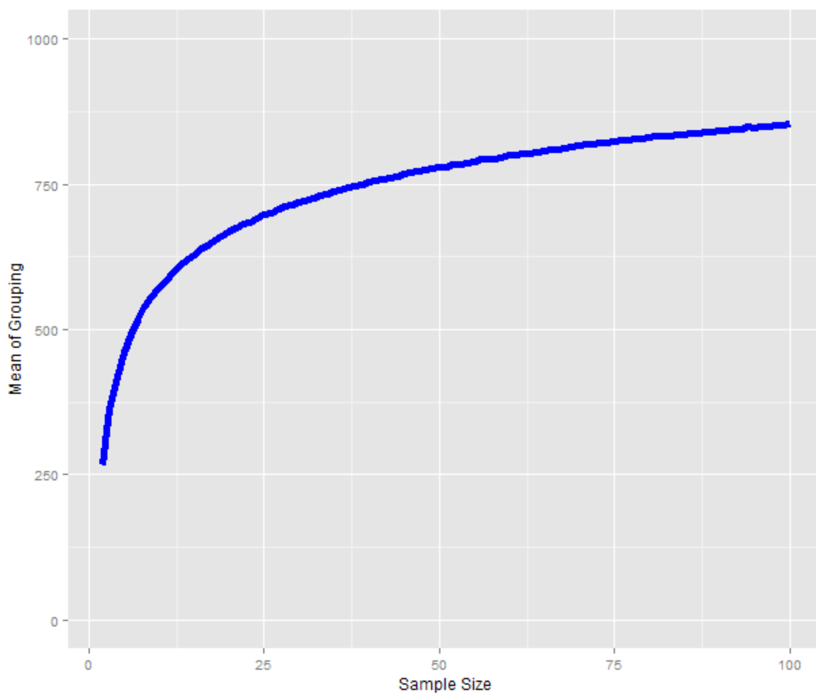

Figure 2. Mean grouping measurement plotted against sample size.

2.5 Circular Error Probable (CEP) Measurement

Circle of Equal Probability is the radius of a circle centered at the mean to capture 50% of the points. Modern mathematical and statistical texts do not care for this measurement method but it is used in the field of geography and military ballistics. The CEP measurement is a favorite of shooters because of the simple definition.

To derive, we must find the radius r^* such that a circle centered at the mean that captures 50% of the bullets impacts.

$$P(0 < r < r^*) = \frac{1}{2}$$

Therefore, the first thing that is required is to simplify the left side of the equation. From earlier we found the probability density function for a radius given by $h(r)$. Then the probability of a point lying within a radius r^* is given by

$$P(0 < r < r^*) = \int_0^{r^*} h(r)dr$$

$$= \int_0^{r^*/\sqrt{2}\sigma_x} ue^{-u^2}du$$

$$= -e^{-\left[r^*/\sqrt{2}\sigma_x\right]^2} + 1.$$

Thus, we now can solve $P(0 < r < r^*) = \frac{1}{2}$ for r^*.

$$1 - e^{-\left[r^*/\sqrt{2}\sigma_x\right]^2} = \frac{1}{2}$$

$$e^{-\left[r^*/\sqrt{2}\sigma_x\right]^2} = \frac{1}{2}$$

$$\left[r^*/\sqrt{2}\sigma_x\right]^2 = \ln(2)$$

$$r^* = \sqrt{2\ln(2)}\,\sigma_x \approx 1.1774\sigma_x$$

Therefore, we have a scaling difference only between the CEP and the standard deviation of the original multivariate normal distribution. r^* is a theoretical quantity therefore we want to find the estimate for r^*, the CEP measurement. Therefore, we use the estimate we found earlier of σ_x given by $\widehat{\sigma_x} = \sqrt{\frac{\sum_{i=1}^{n}[(x_i-\bar{x})^2+(y_i-\bar{y})^2]}{2n}}$ to yield

$$r^* = \sqrt{\ln(2)\,\frac{\sum_{i=1}^{n}[(x_i-\bar{x})^2+(y_i-\bar{y})^2]}{n}}.$$

According to Moranda this estimate exhibits a bias. Moranda introduces an unbiasing factor to make the estimate unbiased and is given below

$$\widehat{CEP_1} = \sqrt{-\ln(1/2)\,\frac{\sum_{i=1}^{n}[(x_i-\bar{x})^2+(y_i-\bar{y})^2]}{n}}\left(\sqrt{n}\,\frac{\Gamma^2(n)}{\Gamma({}^1/_2(2\mathrm{n}+1))}\right),$$

where Γ is the gamma function.

A simulation is the devised to understand this estimate better and compare with the grouping measurement. We start by taking a sample from a multivariate normal distribution. So let $(x_i, y_i) \sim iid\ MVN\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}150 & 0\\ 0 & 150\end{bmatrix}\right)$ for $i \in (1, \dots, n)$.

The grouping measurement and the Circular Error Probable measurement is found for the sample. This process is then repeated 20,000 times and for $n \in (2,3,\dots,100)$. Therefore, we have 20,000 simulations of the grouping and Circular Error Probable measurement for the exact same parameter values. We then define the mean and variance of the Circular Error Probable measurement to compare to the grouping measurement.

$\hat{\mu}_{cep} = \overline{cep}$ is the mean of the Circular Error Probable measurement.

$S_{cep}^2 = \frac{\sum(cep-\overline{cep})^2}{n-1}$ is the variance of the Circular Error Probable measurement.

Below is a plot of the mean CEP measurement for the simulation data. Unlike the grouping measurement, the CEP measurement is not effected by sample size. This will allow direct comparison of measurements with different sample sizes.

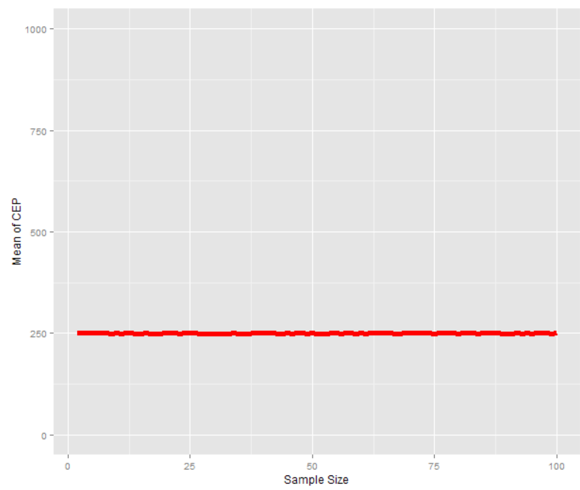

Figure 3. Mean circular error probable measurement against sample size.

Now that we have two measures we can compare them using a combination of their sample mean and variances together to understand the quality of the measures. Figure 4 is the mean plotted as before but now with bands between the 5th and 95th percentile of the 20,000 observations. As sample size increases the width between the 5th and 95th percentile shrinks drastically and therefore has less variability. The shrink in variance is not observed in the grouping measurement.

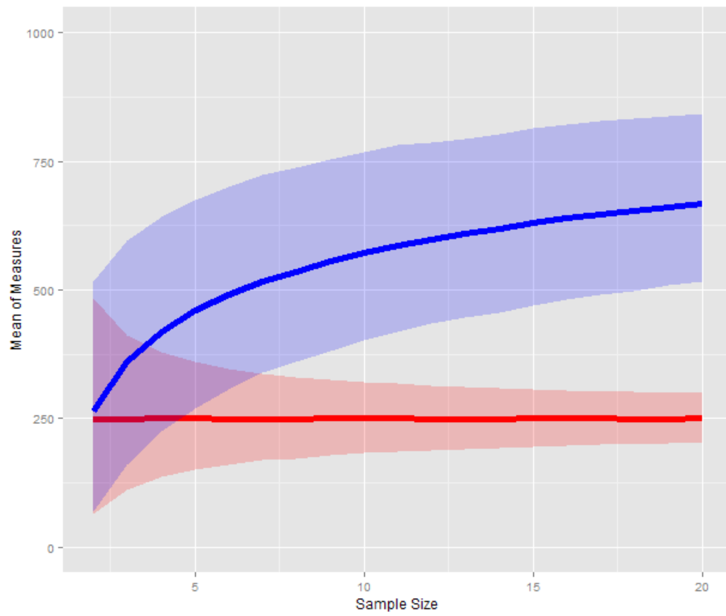

Figure 4. Plot of the mean grouping measurement and the mean Circular Error Probable measurement with area shaded between 5th and 95th percentiles.

The index of dispersion is $\frac{S^2}{\overline{x}}$ where S^2 is the sample variance and $\overline{x}$ is the sample mean. In a high quality measure we hope that the variability is small in comparison to the mean and therefore the index of dispersion should be small. Figure 5 shows the index of dispersion of the grouping measurement and the circular error probable measurement for each sample size. Clearly, the Circular Error Probable is again superior by having a smaller index of dispersion for all sample sizes. The Circular Error Probable's obvious superiority as a measure compared to grouping indicate it should be the measure of choice for our software analysis.

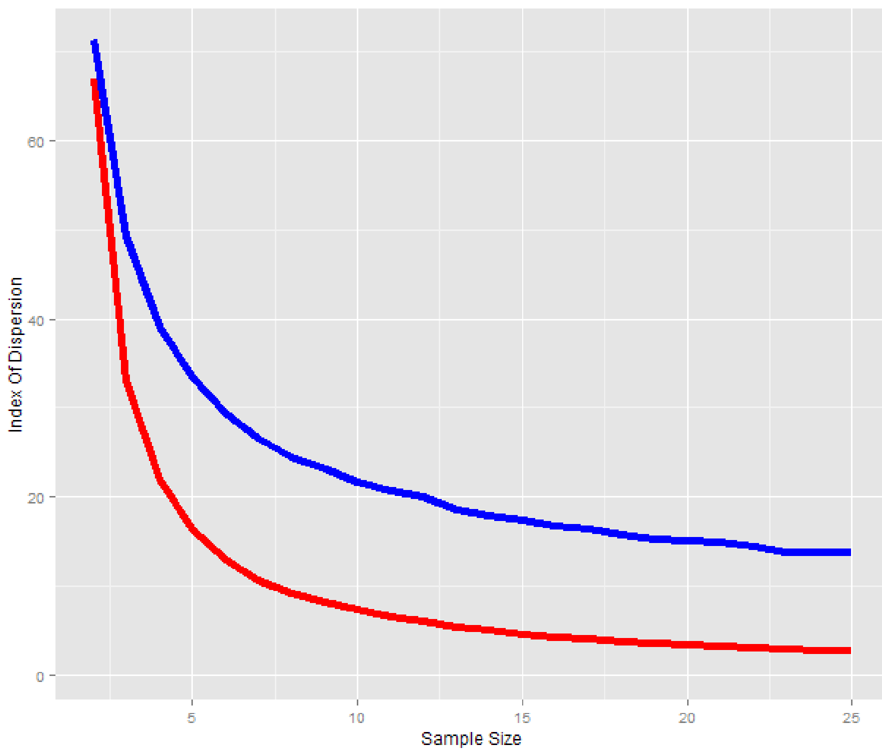

Figure 5. Plot of the index of dispersion for the grouping and circular error probable measurement.

Recall the circular error probable estimate found earlier,

$$\widehat{CEP_1} = \sqrt{\ln(2)\frac{\sum_{i=1}^{n}[(x_i-\overline{x})^2+(y_i-\overline{y})^2]}{n}}\left(\sqrt{n}\frac{\Gamma^2(n)}{\Gamma(^1/_2(2n+1))}\right).$$

$\widehat{CEP_1}$ is the precision CEP because the formula uses sample mean with $\overline{x}$ and $\overline{y}$ to adjust for when $\mu_x \neq 0$ and $\mu_y \neq 0$. If we assume we know $\mu_x = 0$ and $\mu_y = 0$ then we can reduce the MLE estimate found to yield

$$\widehat{CEP_2} = \sqrt{\ln(2)\frac{\sum_{i=1}^{n}[{x_i}^2+{y_i}^2]}{n}}\left(\sqrt{n}\frac{\Gamma^2(n)}{\Gamma(^1/_2(2n+1))}\right).$$

$\widehat{CEP_2}$ can be thought of as the accuracy CEP because it assumes the center to be at (0,0) and does not correct with the sample mean. The circular error probable measure can now measure the two main characteristics of interest in shooting; accuracy and precession.

3. Program

3.1 Program Outline

The goal is to convert a physical target into a series of coordinates representing the center of the holes on the target. The program is written in Matlab and utilizes the image processing toolbox. The program takes an image and a few auxiliary variables and locates all the holes in the target. The centers of the found holes are saved as a comma delimited spread sheet. These files then can be combined for a larger analysis.

Using a digital scanner, high quality images of a target can be generated. The scanner uses records at a fidelity of approximately 300 pixels per inch. The pictures recorded are 3229 pixels by 2480 pixels. Below is an example target with 5 sub targets each with 3 shots. Each hole in the target is outlined in a blue circle which was detected by the program.

Figure 6. Mini Target three shots each. Blue circles outline each circle detected by the program.

The program is composed of four distinct subroutines. Figure 7 shows an outline of the software and how it works. The following sections will report in detail the process in each routine.

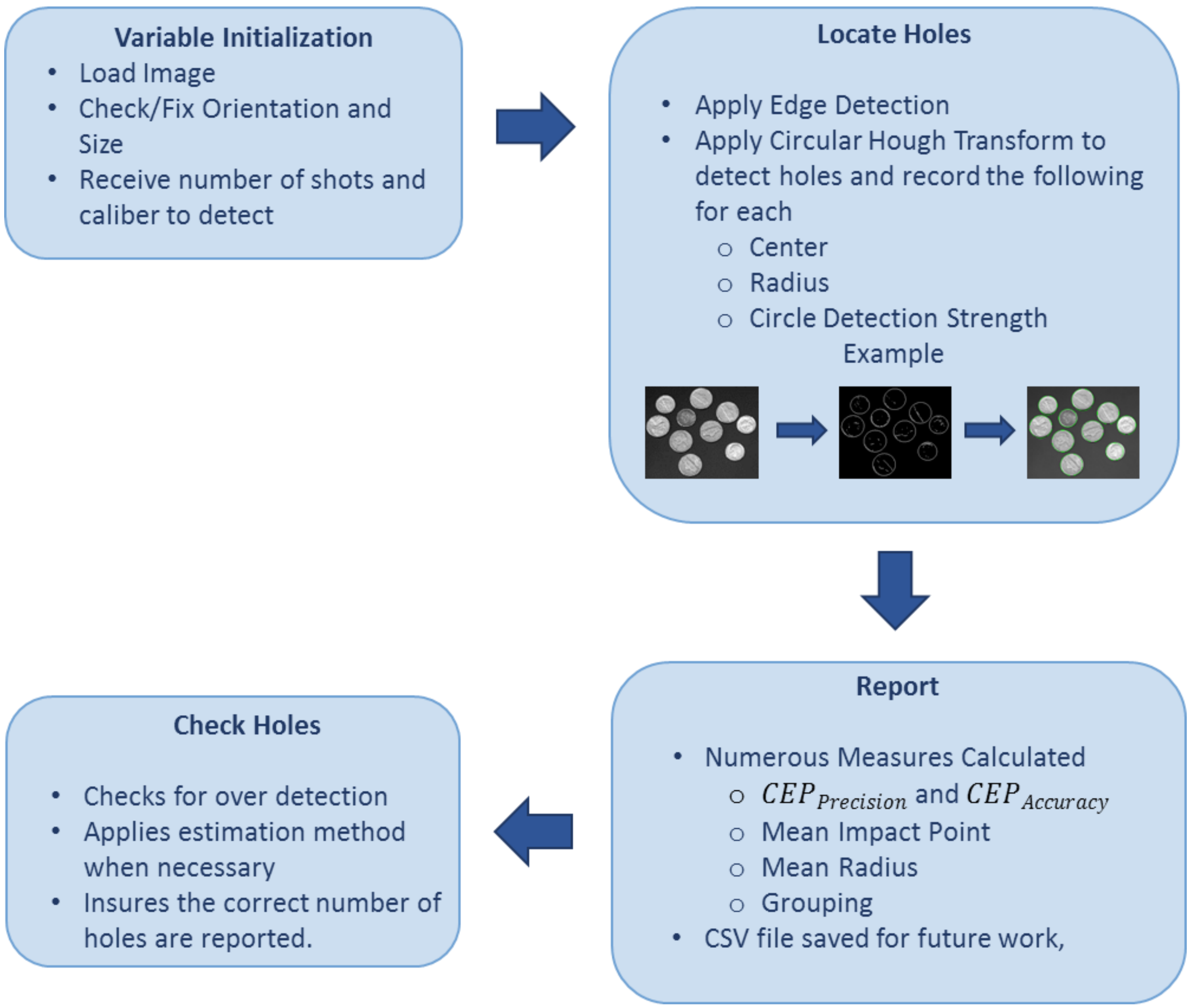

Figure 7. Software Flowchart.

3.1.1 Variable Initialization

The first routine has three goals. The initial variables must be assigned values, the target image must be loaded into Matlab and the orientation of the image must be checked/fixed.

Let R be the radius of the holes to be detected and let N be the number of holes to be detected. R is utilized by the Circular Hough Transform (CHT) in the Locate Holes routine. This insures the CHT detects only holes with a radius approximately of size R. N is utilized in the Check Holes routine to guarantee N circles are sent to the Report routine.

The next step is to load the image into Matlab. The color scanner used produces an image file that contains three data values for each pixel in the form of a JPEG file. These data values are for the amount of red, blue, and green in a value from 0 to 255. This is an unnecessary amount of information and is simplified by averaging the three colors to create a black and white image. The image is now converted into the matrix M of size 3229 by 2480 with each cell value equal to the corresponding pixels value. Therefore, a white pixel corresponds to a 255 value in the matrix and a black pixel such as one caused by a bullet hole correspond to a 0.

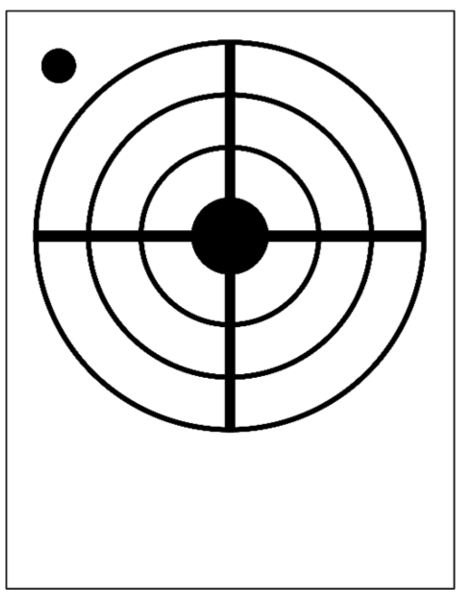

Figure 8. Example target in correct orientation.

Figure 8 shows the designed target in the correct orientation. Figure 9 shows the other possible orientations from incorrectly scanning the target. Therefore, the next step is to check and, if incorrect, fix the orientation of the scanned target. The center of the target in Figure 8 is at (1240, 1665) and a second circle is located that is offset and located at (300,300) with a radius of 200. The Circular Hough Transform is applied to the image to locate the center of the bull's eye and offset circle. The values are recorded with the variables (x_c, y_c) for the bull's eye and (x_o, y_o) for the offset circle.

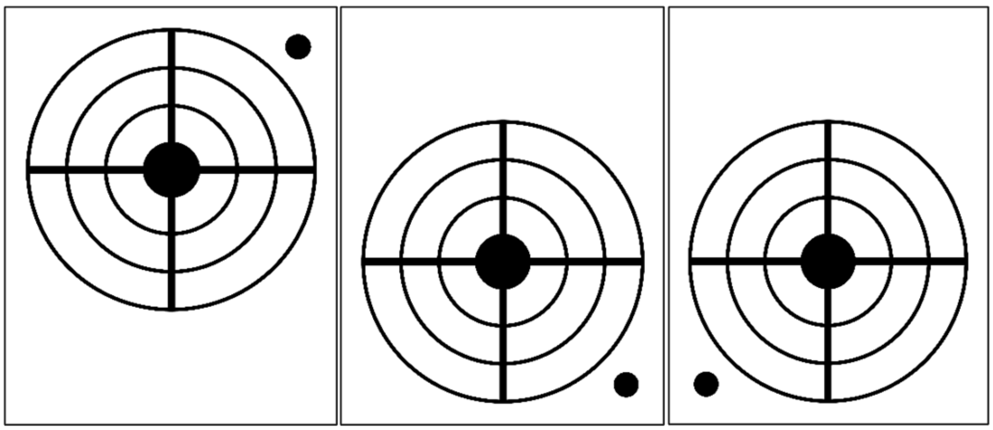

Figure 9. Three other possible orientations of target.

The two circle centers can then be used to determine the orientation of the target. The necessary flips are then applied to correct the image. If $y_c < 1615$ then flip the image over the x axis. If $x_o > 1240$ then flip over the y axis. This will always result in the correct orientation shown in Figure 8.

3.1.2 Locate Holes

The Locate Holes routine scans the image using feature detection algorithms to find the holes in the image with a radius near R. MATLAB's imfindcircles function locates the holes in the image using the Circular Hogue Transform. The function returns a (x_i^*, y_i^*, r_i, a_i) for $i \in (1,2,, \dots, n)$, where r_i is the radius of the detected circle, the point (x_i^*, y_i^*) is the center location of the detected circle and where a_i is a metric for the strength of the detection. The strength metric is determined from the accumulator score generated for the detected circle.

Matlab's imfindcircles works by first applying an edge detection process to the image. The image that is created shows in white the edges of the original image. This is illustrated in Figure 10. Now the Circular Hough Transform can be applied to the image.

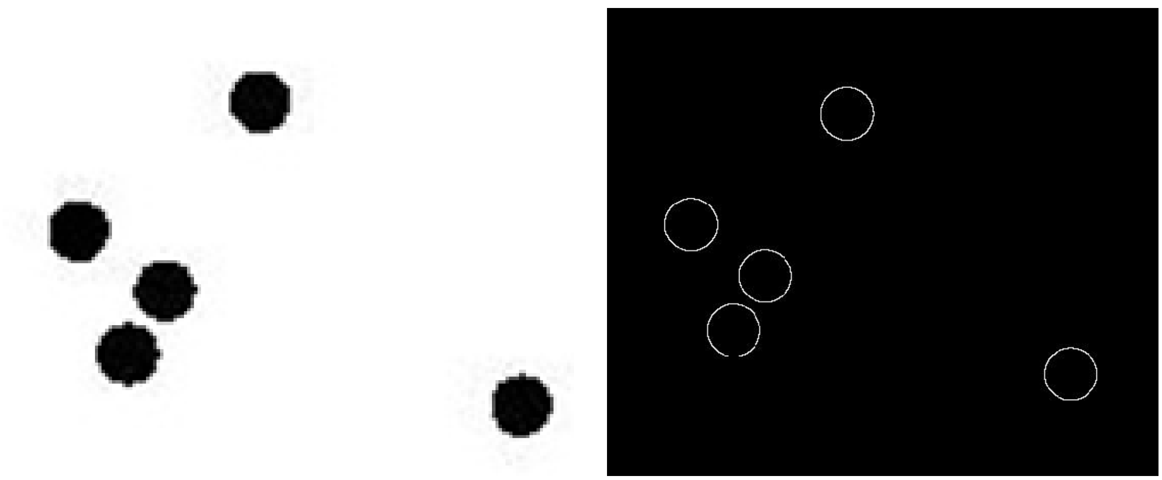

Figure 10. Left side scanned holes in target. Right side is after the edge detection is applied.

The Circular Hough Transform goes to each white pixel and draws a circle with a radius R. Then at every point along this drawn circle 1 vote is applied to the accumulator space. The points with the highest accumulator score are from overlapping circles and must be the center of a hole. Below is an illustration from Artherton that shows the Circular Hough Transform process.

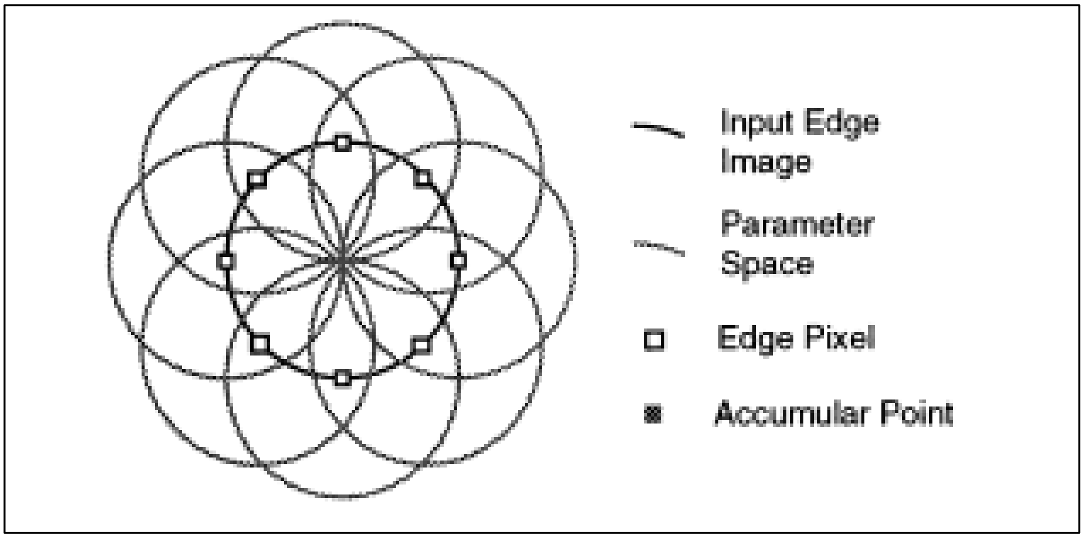

Figure 11. Illustration of Circular Hough Transform. Middle circle's center is detected by drawing circles with center at the middle circles edge. Every point along the drawn circle applies one vote to the accumulator space.

The imfindcircles performs flawlessly as long as there are no intersections between holes. The error in between the real center point and the detected center point is nonexistent under ideal conditions. Yet, if there is an intersection between holes the imfindcircle function will miss all of the intersecting holes.

3.1.3 Check Holes

Recall earlier the Find Holes routine discovered n holes which can be bigger or smaller then the true number of holes N. The most common failure is when there are overlapping holes and the Hough Transform misses them. Missing values are a huge problem because generally we have small sample size and if there is an intersection then at least two will be missing values. Also, the Hough Transform may have found many more holes then in reality. Therefore, the Check Holes insures the best detected holes are used and when not found estimates are located.

First, the routine addresses over detection. This is caused by detecting one hole multiple times. Figure 12. Illustrates over detection with 7 found holes but only 5 holes are truly there. Consider (x_i^*, y_i^*, r_i, a_i) for $i \in (1,2,,\dots,n)$ as before but with at least one overlapping detected holes. Two detected holes, (x_A^*, y_A^*, r_A, a_A) and (x_B^*, y_B^*, r_B, a_B), are defined to overlap significantly when $\sqrt{(x_A^* - x_B^*)^2 + (y_A^* - y_B^*)^2} < R/2$. This occurs when the two holes are within half the radius of each other. Only one of a group of significantly overlapping holes is saved the rest of the holes are deleted. The one chosen has the largest a, the metric for the strength of the circle detection.

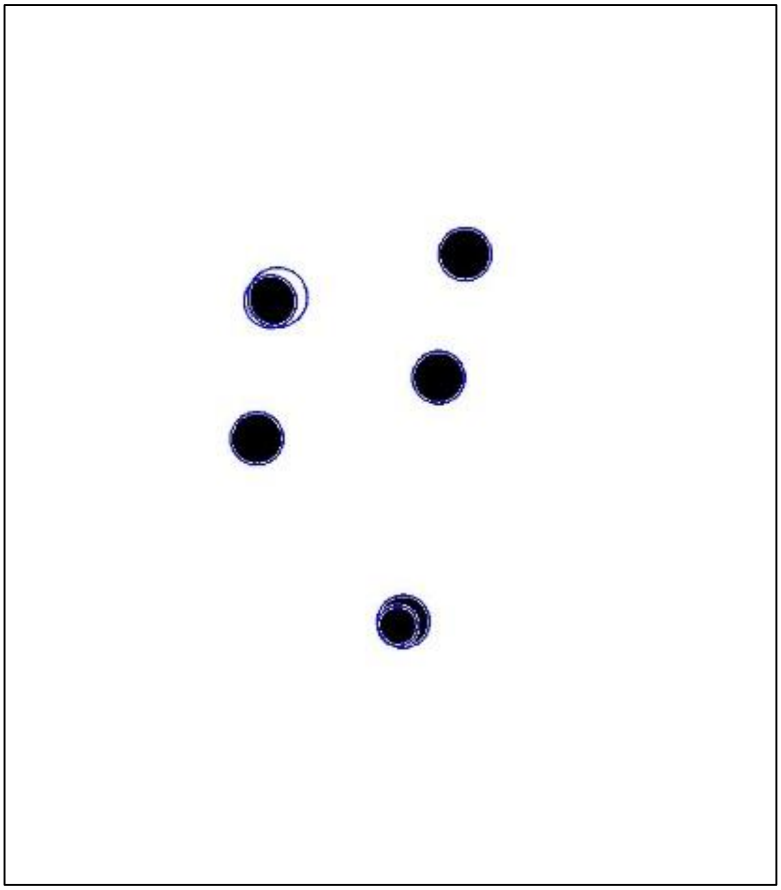

Figure 12. Example of over detection. Here $N = 5$ but $n = 7$ because two holes are detected twice.

Now consider the case $n < N$, this is when not enough holes are detected by the program. This is generally caused by holes overlapping and the Circular Hough Transform failing. An estimate for the $N - n$ missing holes must be determined. We delete all pixels found within the circles found by the Circular Hough Transform. Therefore, at each (x_i^*, y_i^*) we proceed in a spiral converting the image value to white. Left over will be the intersecting holes that were not detected or not found and random noise. This process is shown in Figure 12 where the left side is before and after. All pixels coordinates whose color value is less than 15 are averaged to produce the estimate for the missing $N - n$ holes. Note that all missing values are assigned the same estimate value.

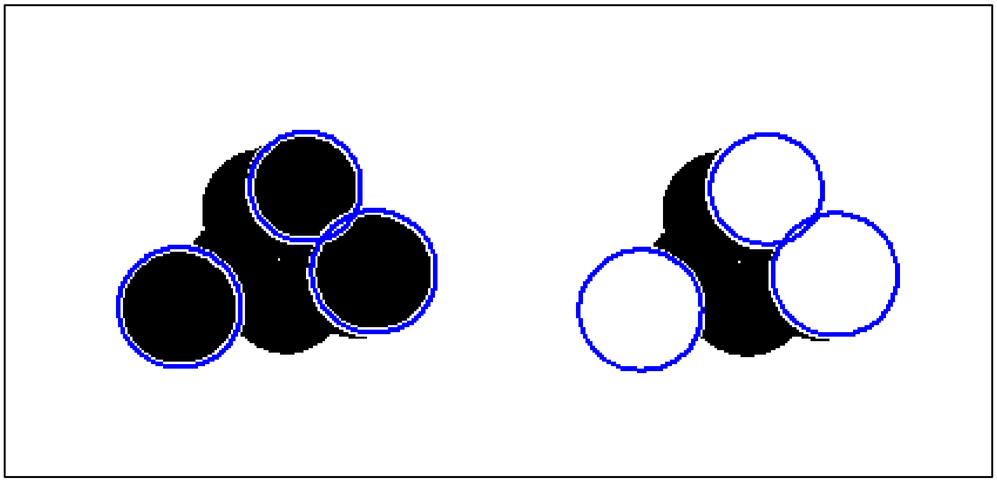

Figure. 13 The above picture has 8 overlapping circles but only 3 are detected. To the right is the image after deletion for estimation.

When there are only two overlapping holes. The distance between the true values and the estimated values is then within R. As the number of holes that are to be estimated and number of groups increase, the error increases significantly. I believe this is where most of the error in the program originates and future work should be applied to develop better estimation methods for missing values.

Now consider the case $n > N$, this is when too many holes are detected by the program. The process used for over detection is applied again but under stricter conditions. Two detected holes, (x_A^*, y_A^*, r_A, a_A) and (x_B^*, y_B^*, r_B, a_B), are defined to overlap when $\sqrt{(x_A^* - x_B^*)^2 + (y_A^* - y_B^*)^2} < R$. As before, only one observation from a set of overlapping holes is kept dependent on a, the metric for the strength of the circle detection.

Unfortunately, this does not guarantee $n = N$. If we still have $n > N$ then the N holes with the largest a are chosen. If we then have $n < N$ the estimation process is run to detect the $N - n$ missing values. When the correct number is found, that is $n = N$, the program continues to the Report routine. The methods described can be improved greatly. Further work will be focused on this section to find better techniques.

3.1.4 Report

The goal is to convert the scanned target into a series of Cartesian coordinates (x_i, y_i) for $i \in (1,2,,\dots,N)$. We receive the series of coordinates (x_i^*, y_i^*) for $i \in (1,2,,\dots,N)$ from the Check Holes routine. The coordinate system of (x_i^*, y_i^*) has the origin at the top left of the image as in the matrix notation. Recall, the target centers location given by (x_c, y_c) found earlier. This will be the origin of the of final coordinates system (x_i, y_i).

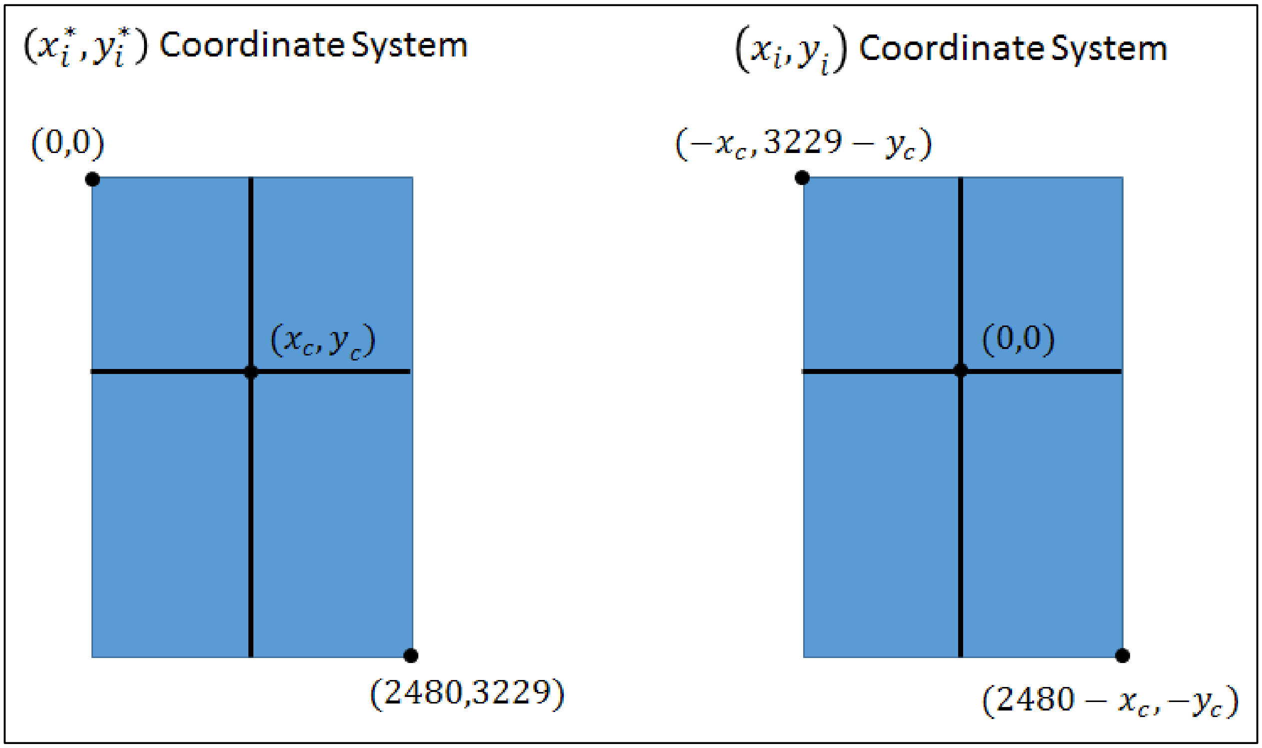

Figure 14. Orginal coordinate system on left from matrix notation. Final coordinate system on right with the origin located at (x_c, y_c)

To yield the (x_i, y_i) Cartesian coordinates with the origin at the target center. The transformation to (x_i, y_i) is defined as

$$x_i = x_i^* - x_c \text{ and } y_i = (3229 - y_i^*) - y_c.$$

Now in the final coordinate system a multitude of descriptive measurements are calculated for the detected holes. These can be used by shooters to understand characteristics of their performance.

$$\overline{r} = \frac{\sum_{i=1}^{N} \sqrt{{x_i}^2 + {y_i}^2}}{N}$$

$$\overline{x} = \frac{\sum_{i=1}^{N} x_i}{N}$$

$$\overline{y} = \frac{\sum_{i=1}^{N} y_i}{N}$$

$$CEP_{Precision} = \sqrt{\ln(2) \frac{\sum_{i=1}^{N}[(x_i - \overline{x})^2 + (y_i - \overline{y})^2]}{N}} \left(\sqrt{N} \frac{\Gamma^2(N)}{\Gamma({}^1/_2\,(2\mathrm{N}+1))} \right)$$

$$CEP_{Accuracy} = \sqrt{\ln(2) \frac{\sum_{i=1}^{N}[{x_i}^2 + {y_i}^2]}{N}} \left(\sqrt{N} \frac{\Gamma^2(N)}{\Gamma({}^1/_2\,(2\mathrm{N}+1))} \right)$$

$$Grouping = \max\left\{ \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2} \right\} \text{ for every } i \text{ and } j \in (1,2, \dots, N)$$

Furthermore a csv (comma-separated value) file is constructed and saved with the (x_i, y_i) for $i \in (1,2,, \dots, N)$. The data can now be used in the future and used in combination with several files to perform a study or analysis.

3.2 Analysis of Program Accuracy

The software reliably produces an (x_i, y_i) for $i \in (1,2,,\dots,N)$. As discussed, sometimes these are just estimated values and even under ideal conditions there is a tiny amount of error. Therefore, the software must be analyzed to understand its performance. A simulation is designed to test the program. The simulation takes a sample of data and produces an image that is analyzed by the software. We can now compare the true sampled data with the found data from the software.

First a method is developed to create an image based on sampled data. The image file appears with perfect black circles always free from defects. In the real world there will be additional error from printing and scanning therefore this simulation is considered a best case scenario.

Each image starts as a matrix M which is 3229 by 2480 in size filled with the number 255 which represent a white pixel. Consider a sample of center points for holes given by $(x_i, y_i) \sim iid\ MVN\left(\begin{bmatrix}1240\\1615\end{bmatrix}, \begin{bmatrix}100 & 0\\0 & 100\end{bmatrix}\right)$ for $i \in (1,\dots,n)$. We then change all (x, y) in M that satisfy following $\sqrt{(x-x_i)^2 + (y-y_i)^2} < r$ for $i \in (1,2,,\dots,n)$. The result is the matrix M that has 0s or 255s for each value where each center hole from the sample is represented in 0s in a circular shape of radius r. Therefore, when the matrix is viewed as a black and white image we get a series of black circles on a white background.

The software analysis is then applied to the image resulting in the center coordinates given by $(\hat{x}_i, \hat{y}_i)$ for $i \in (1,2,,\dots,N)$. We then calculate the following,

$$r_i = \sqrt{(x_i - \overline{x_i})^2 + (y_i - \overline{y_i})^2}$$

$$\hat{r}_i = \sqrt{\left(\hat{x}_i - \overline{\hat{x}_i}\right)^2 + \left(\hat{y}_i - \overline{\hat{y}_i}\right)^2}$$

$$MeanError = \frac{\sum_{i=1}^{N}[\hat{r}_i - r_i]}{N}$$

$$AbsoluteMeanError = \frac{\sum_{i=1}^{N}|\hat{r}_i - r_i|}{N}$$

$$MSE = \frac{\sum_{i=1}^{N}(\hat{r}_i - r_i)^2}{N}$$

$$Rsq = 1 - \frac{\sum_{i=1}^{N}(\hat{r}_i - r_i)^2}{\sum_{i=1}^{N}(r_i - \overline{r})^2}$$

$$CEP_{Precision} = \sqrt{\ln(2)\frac{\sum_{i=1}^{N}[(x_i - \overline{x})^2 + (y_i - \overline{y})^2]}{N}}\left(\sqrt{N}\frac{\Gamma^2(N)}{\Gamma(^1/_2\,(2\mathrm{N}+1))}\right)$$

$$\widehat{CEP}_{Precision} = \sqrt{\ln(2)\frac{\sum_{i=1}^{N}\left[\left(\hat{x}_i - \overline{\hat{x}_i}\right)^2 + \left(\hat{y}_i - \overline{\hat{y}_i}\right)^2\right]}{N}}\left(\sqrt{N}\frac{\Gamma^2(N)}{\Gamma(^1/_2\,(2\mathrm{N}+1))}\right)$$

$$CEP_{error} = \widehat{CEP}_{Precision} - CEP_{Precision}.$$

This will provide a clear understanding for one simulation, at one value of N. Yet, will very if another simulation were run. Therefore, we repeat taking a sample and collecting values 300 times. We will let K represent the number of times we sample and calculate for a single N. Therefore, we can now proceed and calculate summary information for the 300 sets of observations.

$$MeanMeanError = \sum_{i=1}^{K} MeanError_i$$

$$MeanAbsoluteMeanError = \sum_{i=1}^{K}\frac{Rsq_i}{K}$$

$$MeanMSE = \sum_{i=1}^{K}\frac{MSE_i}{K}$$

$$MeanRsq = \sum_{i=1}^{K}\frac{Rsq_i}{K}$$

$$MeanCEPError = \frac{\sum_{i=1}^{K}\left[CEP_{error_i}\right]}{K}$$

$$AbsoluteMeanCEPError = \frac{\sum_{i=1}^{K} \left| CEP_{error_i} \right|}{K}$$

$$MaxCEPError = Max[\left| CEP_{error_i} \right|]$$

$$MSECEP = \sum_{i=1}^{K} \frac{{CEP_{error_i}}^2}{K}$$

$$RsqCEP = 1 - \frac{\sum_{i=1}^{K} (\widehat{CEP}_{Precision_i} - CEP_{Precision_i})^2}{\sum_{i=1}^{K} (CEP_{Precision_i} - \overline{CEP}_{Precision})^2}$$

The last part of the simulation is to address the number of shots per target. As discussed before the software will encounter problems when overlapping circles exist and they occur in higher frequency with number of shots per target N. Therefore, we must perform the calculations above for a variety of N. To cover a large range of values and focus on smaller sample sizes the simulation was chosen that $N \in (3, 5, 7, 10, 15, 20, 25, 30, 40, 50)$

N	Mean Mean Error	Mean Absolute Mean Error	Mean MSE	Mean Rsq	Mean CEP Error	Absolute Mean CEP Error	Max CEP Error	MSE CEP	Rsq CEP
3	-0.119	0.341	10.689	0.766	0.085	0.198	16.998	1.792	0.998
5	-0.223	0.481	13.782	0.993	-0.132	0.202	10.230	0.932	0.999
7	-0.440	1.026	51.867	0.971	-0.287	0.479	15.778	3.418	0.993
10	-1.286	1.870	115.239	0.963	-0.653	0.750	19.752	4.232	0.984
15	-3.595	4.489	308.809	0.887	-1.584	1.670	13.082	8.305	0.964
20	-6.735	7.827	616.130	0.819	-2.824	2.868	13.564	15.183	0.915
25	-9.928	11.211	895.483	0.748	-4.075	4.085	17.230	26.285	0.815
30	-14.159	15.593	1308.177	0.642	-5.808	5.815	19.615	45.305	0.641
40	-22.060	23.447	2049.428	0.465	-8.850	8.850	19.587	90.442	-0.248
50	-30.964	32.305	3037.891	0.220	-12.653	12.653	30.079	175.363	-1.601

Figure 15. Simulation data table. All units are in pixels.

There is a wealth of information contained in the table. The $MeanMeanError$ is the average error in pixels for the radius for a given N. Therefore, for $N = 3$ the predicted values of the radius are off by -.119 pixels or approximately .000397 inches. This continues to grow with sample size at for $N = 50$, the error has grown to -30.964 pixels or approximately -.1032 inches. The $MeanAbsoluteMeanError$ shows very similar but slightly more extreme error.

The $MeanCEPError$ shows a minimum mean error at $N = 3$ with an error of .085 pixels or approximately .000283 inches. As before this error grows with sample size till $N = 50$ which has a $MeanCEPError$ of -12.653 or approximately -.04218 inches. Furthermore, the maximum of $MaxCEPError$ was detected for $N = 50$ and the error was only 30.079 pixels or .1003 inches.

Mean Squared Error is the cornerstone for measuring model accuracy. Except it is used as a comparison between two models to choose which predicts better. The $MeanMSE$ is strictly increasing with respect to N, indicating that the smaller the sample size the better. Yet, MSE CEP shows a slightly different picture. The software performed best at predicting the CEP when $N = 5$. Therefore, the program predicts the radius and the CEP better for lower sample size N.

R-Squared is a heavily used statistic because of its nice interpretation. Unfortunately, this is a non-linear process and is not bounded above 0. Yet, it still has a maximum of 1. Therefore, for the small values of N we see that the $MeanRsq$ and $RsqCEP$ are relatively high indicating a strong prediction capability. Yet, as stated before, as N increases problems arise and the model begins to suffer greatly.

The intent of this project was to convert a physical target into a series of coordinates representing the center of the holes on the target. The software succeeds in producing the series of coordinates every time. Furthermore, the prediction for $N = 5$ and similar is superb.

Improvements can still be made to the program to decrease the error further. The creation of the simulation architecture allows direct comparison of future programs. The programs with features that provide lower MSE's are performing better and their changes should be adopted.

Bibliography

1. Elder, Richard L. "Am Examination of Circular Error Probable Approximation Techniques." Thesis. Air Force Institute of Technology, 1986. <http://www.dtic.mil/dtic/tr/fulltext/u2/a172498.pdf>.

2. Marques, Oge. Practical image and video processing using MATLAB. Hoboken, NJ: Wiley/IEEE P, 2011.

3. Mathews, Paul. "The Circular Normal Distribution." Mathews Malnar and Bailey, Inc. <http://www.mmbstatistical.com/ToT/distofr.pdf>.

4. Moranda, P.B. Comparison of Estimates of Circular Probable Error. Journal of the American Statistical Association , Vol. 54, No. 288 (Dec., 1959) , pp. 794-800.

5. Nelson, William. Use of Circular Error Probability in Target Detection. Tech. Deputy Commander of Developmental Programs Electronic Systems Division Air Force Systems

6. T.J Atherton, D.J. Kerbyson. "Size invariant circle detection." Image and Vision Computing. Volume 17, Number 11, 1999, pp. 795-803.