DISCOVERY AND CORRECTION OF BIAS IN PRECISION LANDMARK LOCATION

by

Colin Foster

A Thesis Submitted in

Partial Fulfillment of the

Requirements for the Degree of

Master of Science

in Engineering

at

The University of Wisconsin-Milwaukee

December 2012

ABSTRACT

DISCOVERY AND CORRECTION OF BIAS IN PRECISION LANDMARK LOCATION

by

Colin Foster

The University of Wisconsin-Milwaukee, 2012

Under the Supervision of Professor Brian Armstrong

Precision Landmark Location (PLL) estimation is an integral part of 3D motion tracking. Circular landmark location estimation is one method of PLL. Current methods of estimation lead to systematic errors with a magnitude of up to .02 pixels. Estimation inaccuracies of this magnitude lead to unacceptable errors in depth measurement, the largest source of error. In the scope of this thesis, inadequacies in circular landmark location are uncovered and techniques to correct these errors are analyzed, tested, and demonstrated. Deviations in simulated images are seen to be reduced by a factor of three and the variances of real-world data were reduced by half. This thesis predicts and observes increased accuracy in the 3D motion tracking technology.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to thank everybody who has helped my educational career over the past two years. I have learned more than I could have possibly imagined in the process.

Specifically I would like to thank Professor Brian Armstrong for taking me under his wing and providing me the opportunities I have had. Robb Barrows, Todd Kusik, Qiaotian Li, and Drew Douglas have been incredibly supportive and helpful by maintaining an encouraging learning environment. Professor Ethan Munson has offered tremendous insight in finding and resolving the problems seen in this Thesis. My parents: for everything. Danielle: for putting up with me. And of course all of my family and friends for the support through the years.

NIH/NIDA Grant: R01DA021146-A1

Part I

Background

1 Introduction

1.1 Moiré Phase Tracking (MPT)

3D motion tracking is a commonly used technology that spans a diverse number of applications. Animation, automation, medicine, simulation, and entertainment are all examples of industries that utilize and benefit from this technology. Many of the technologies rely on multi-camera setups and can require elaborate setup and calibration which cannot easily be relocated [1, 2, 3]. Technologies can range from the use of Doppler radars [4] and active targets [3].

1.1.1 What is MPT

Moiré Phase Tracking (MPT) is a 3D motion tracking technology developed by Brian Armstrong [5] that addresses many of the downfalls of current 3D motion tracking technologies. MPT uses a *single camera* as a sensor and tracks specially designed targets (Figure 1). The target relies on the detection of the starburst landmark in the center of the target, the location of the four circular landmarks, and the resolution of the periodic moiré patterns. The analysis of all of this information yields six degree of freedom (6-DOF) pose estimation of X, Y, Z and pitch, roll and yaw [6].

1.1.2 How MPT is used

MPT Technology is already being researched and used in a number of applications. Universities in the US and Germany, including the University of Wisconsin-Milwaukee, University of Hawaii, and universities in Freiburg and Magdeburg Germany are all participating in the research.

One specific area of research is to create motion corrected images in Magnetic Resonance Imaging (MRI) scans. MRIs rely on a patient to remain steady during the duration of the scan, which can last very long [7]. If the patient moves during an image it can become blurred and sometimes be unusable for medical diagnosis. Imaging patients who's movements are difficult to control are especially problematic.

Motion tracking during an medical imaging scans is widely studied [8, 2, 9]. Some systems are designed to work in MR imaging scenarios while others work only in non-magnetic situations.

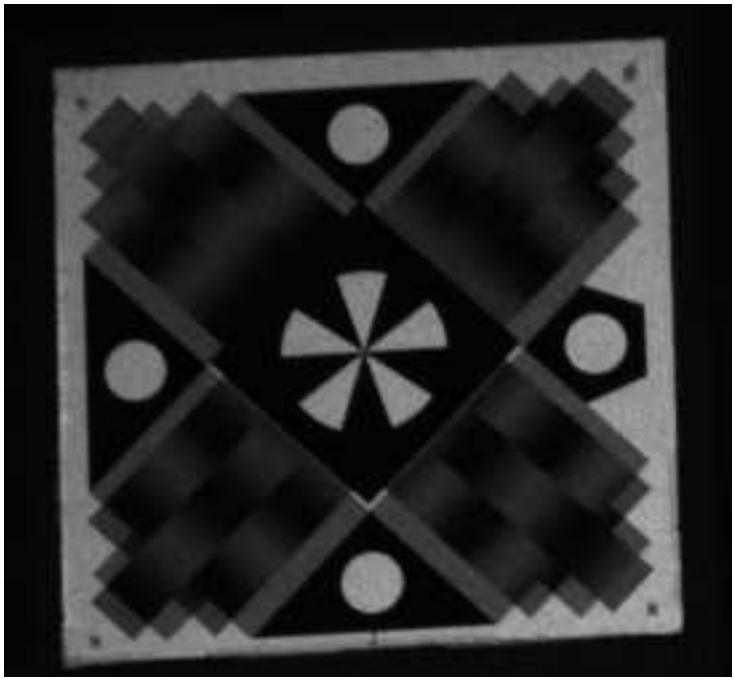

Figure 1: Photograph of an MPT Target.

MPT targets can be placed on patients while they are in the bore of an MRI machine. The data can be processed and report movement to the MRI machine. The scanner can correct for the motion during the scan. Figure 2 shows the product of the use of the technology. In the image, the two images on the left have no correction and the two images on the right have motion correction. Additionally, papers have been published utilizing MPT technology to produce motion corrected MR images [9].

Along with MRIs, MPT is also being used to perform 3D motion tracking of a person in motion. For this, multiple targets are affixed to a person who can perform different tasks. These tasks include jumping, running in various paths, walking, etc. When this data is combined with data from a force plate, a recreation of the extremities can be performed and analyzed, as seen in Figures 3 and 4. The use of MPT technology has been shown to aid in the assessment of injury susceptibility and diagnostics [10, 11].

1.2 Jump

One phenomena that has been observed is known as a "jump." A jump is a sharp change in the estimated depth (Z) in an orientation where there should be very little change. Jumps are of significantly larger magnitude than the noise and are believed to be attributed to the largest source of error in MPT.

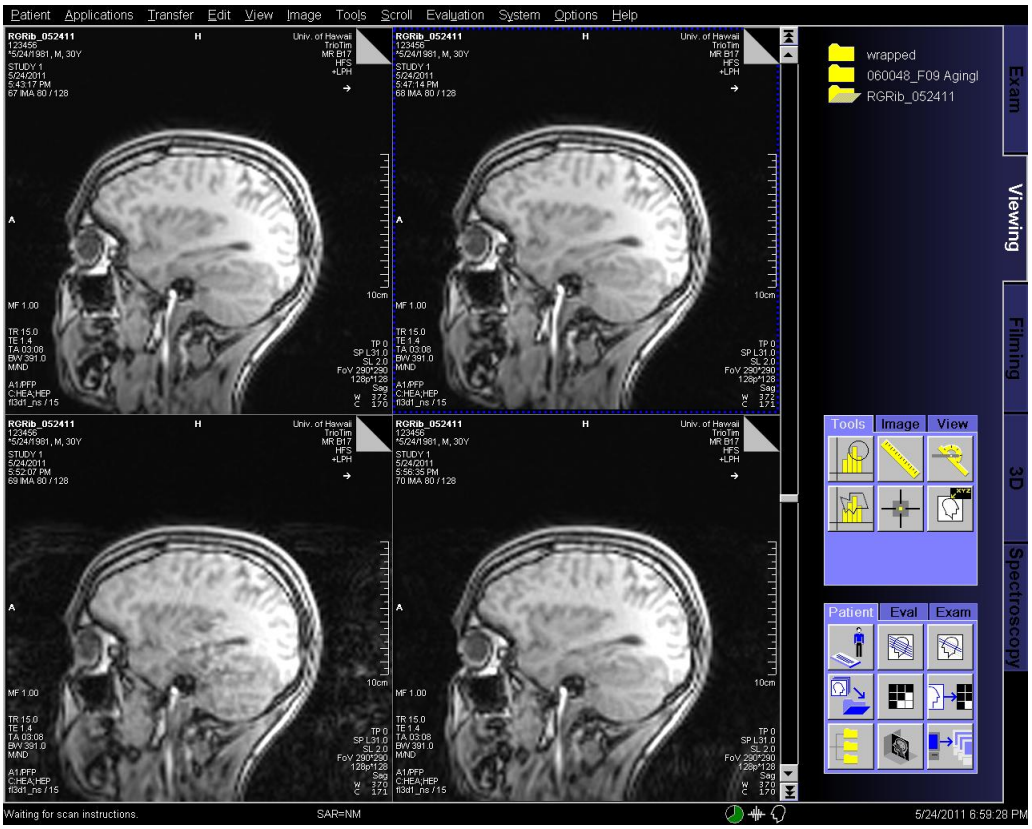

Figure 2: Comparison of standard and motion corrected images. The bottom two images have intentional motion and the right two images have motion correction enabled.

Figure 3: A subject jumping with MPT targets affixed to them.

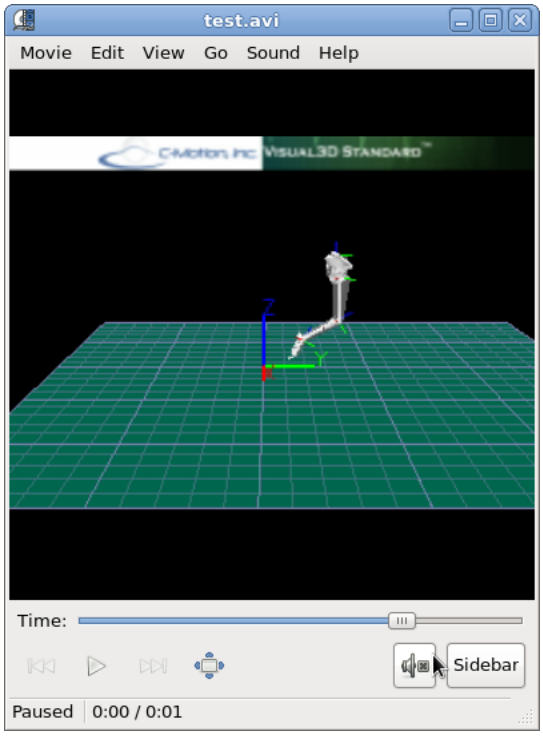

Figure 4: A skeletal recreation of a running subject. Targets were placed on a subject and used to create a model for analysis.

A strong component of the depth estimation of MPT is related to the location of the four circular landmarks. The further away the landmarks are from one another, the closer the target is to the camera. Chapter 2 further explores this idea and shows that landmark estimation is in fact the cause of the jumps.

1.3 Precision landmark location

Precision landmark location (PLL) is the process of taking the image data and determining precisely where the landmark is located. PLL is a fundamental technology used in MPT.

1.3.1 History of precision landmark location

Many PLL techniques have been researched over the past few decades. Much investigation into the detection and location of lines in images [12, 13, 14] which could give accurate position estimation in a single dimension. Other studies have been done on accuracy of circular fiducials, but only of binary images [15, 16, 17, 18, 19]. It has also been shown that the use of larger patterns such as bullseyes [17, 18] or starbursts [20] can offer more accurate location estimation, however the patterns are much larger than circular fiducials and would require a much larger target [21]. It has been claimed that a

bullseye pattern can provide a more precise landmark in the same area as a circular landmark [22], however this analysis was done using a centroid estimation technique. Gutierrez and Armstrong showed significant improvements in precision by using alternative estimation techniques and smaller circular landmarks.

Other studies involve the analysis of multiple images [23, 24] or more complex landmarks [25], which are not applicable to MPT.

Noise is also a common discussion in the accuracy of landmark location estimation [16, 18, 19, 20, 26] but bias is often never mentioned [27, 28]. It is seen that bias in the estimation of circular landmarks exists when approaching the precision of PLL. Additionally, this bias is much larger than the effects of noise on the image. Bias affects the center estimation of the circular landmarks, which affects the distance measurement of MPT.

1.3.2 Previous investigations into biased estimators

Robinson and Milanfar [27] have investigated the effects of biased estimation in image processing. Their motivation was to improve the accuracy of temporal sampling of similar images. These techniques can be used in robotics to analyze motion of a landscape or of the camera relative to the surroundings [28]. They show that unbiased estimators do not exist in image processing and that the effects are significant enough to not be ignored.

The authors use a number of different motion estimation techniques to support their claim. They tested these techniques with multiple different images and large deviations in magnitude of noise. Additionally, they showed that estimation bias occurs in the shifting of randomly generated images where the derivative is known. The authors continue by modelling and analyzing the bias and experimentally validating their predictions.

This previous work is an important introduction to the idea of bias in image registration estimation, however only focuses on the analysis of multiple images and 2-D translation. The groundwork laid by Robinson and Milanfar can be understood and the ideas can be applied to landmark registration within a single image. This continuation of their work is what is done in this Thesis.

Additionally, Kim and Menq [28] have studied the effects of bias with both translation and depth changes. They use a normalized cross correlation technique to detect motion. Although their study investigates the effect of bias, it is limited to the detection of shifts and motion in images. This Thesis investigates the absolute estimation of location from a single image.

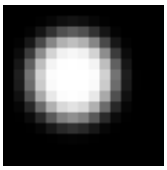

Figure 5: Image of a circular MPT landmark. The center of this landmark is estimated and used for analysis.

1.3.3 Precision landmark location in MPT

The PLL process used in MPT is a two step process. First, points are located where the intensity equals a specific threshold. A landmark is shown in Figure 5 and a cross section showing the intensity and an estimation threshold is shown in Figure 6. The estimated contour points should represent an ellipse, as shown in Figure 7.

The points on the ellipse can then be fit to a known ellipse. The equation from equation 1 where (x_k, y_k) the k^{th} point, (x_0, y_0) the estimated center, and ellipse paramaters $[a, b, c]$, can be used to model an ellipse. Then the error of the estimate as seen in equation 2 can be calculated. A Newton-Raphson algorithm can be performed to minimize the error and give a best estimate of the landmark center. The paramaters used showe the accuracy of this procedure to be on the order of *six milipixels*; approximately 6/1000ths of a pixel.

$$1 = \begin{bmatrix} a & b & c \end{bmatrix} \begin{bmatrix} (x_k - x_0)^2 \\ (x_k - x_0)(y_k - y_0) \\ (y_k - y_0)^2 \end{bmatrix} \tag{1}$$

$$\epsilon = \Sigma \left(\tilde{z}_k \begin{bmatrix} a & b \\ b & c \end{bmatrix} \tilde{z}_k^T - 1 \right)^2 \; ; \; \tilde{z}_k = \begin{bmatrix} x_k - x_0 \\ y_k - y_0 \end{bmatrix} \tag{2}$$

It is also seen that if the contour points found are ideal, accuracy of the center estimation greatly improves. Much of this Thesis involves attempts to identify the sources of error in the contour point estimation in order to approach the ideal contour point locations.

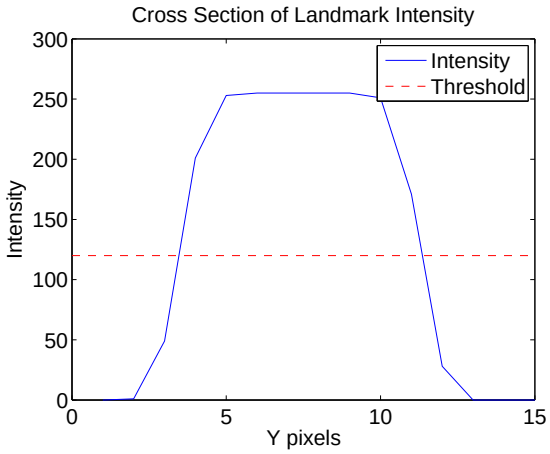

Figure 6: Cross section of a landmark and an intensity threshold. The line intersections are the estimated contour points.

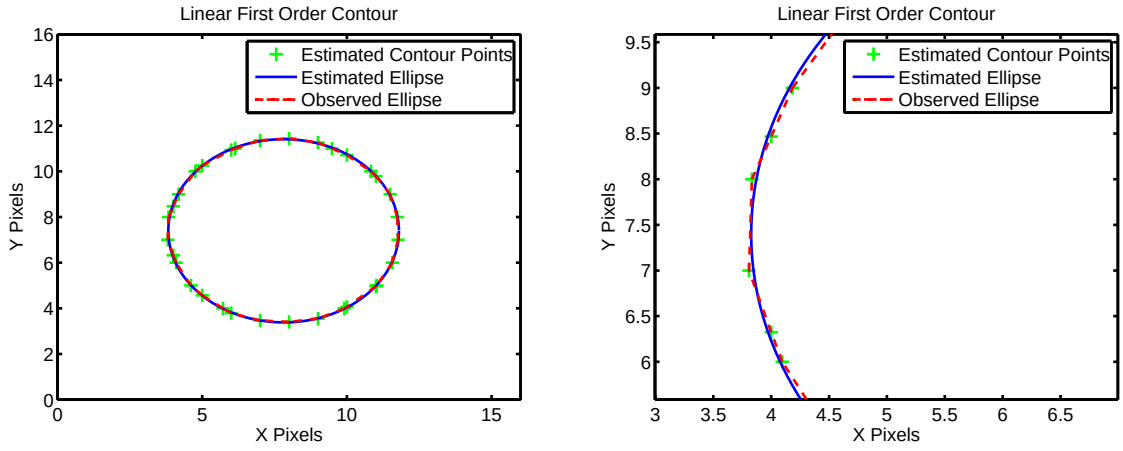

Figure 7: Contour plot of a landmark. The points that create this shape are fit to an ellipse and the center is accepted as the landmark location.

1.4 What is in this Thesis?

This Thesis uncovers specific information relating to the jump seen. Two main points discovered are

1. There is bias in the PLL process and this bias is much greater than the observed noise
2. Curvature in the intensity profile leads to the case where the average intensity over a pixel is not equal to the intensity at the center of the pixel

Also in this Thesis, different numerical interpolation and PLL techniques are investigated. It is shown that the magnitude of PLL bias can lead to jump errors of similar magnitude to empirically observed jumps. It is shown that better interpolation data could lead to more accurate and precise results. And it is shown that different interpolation techniques can improve PLL. The theories are tested and shown to improve results with real-world data.

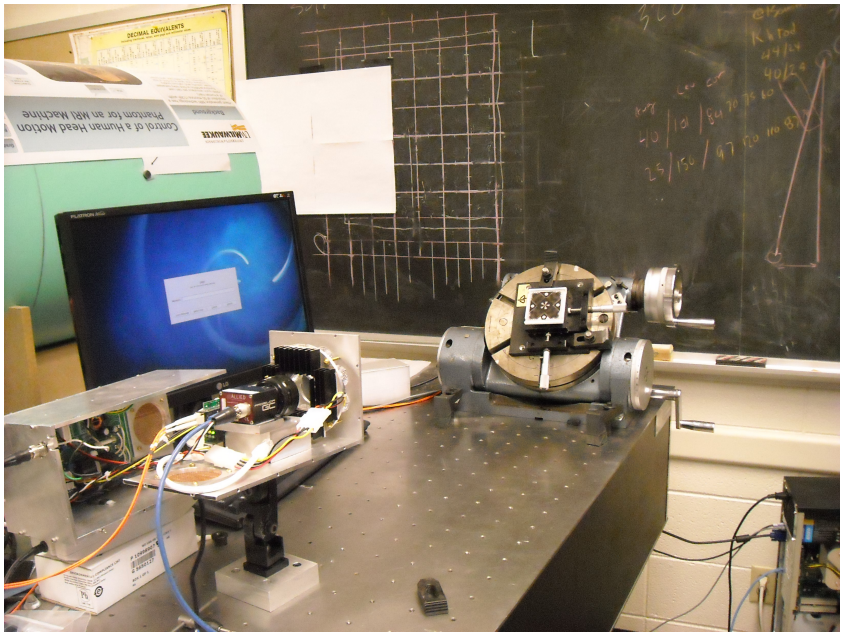

Figure 8: Setup of the jump experiment with a tilting rotary table, MPT target, camera, and computer.

2 Experiments That Establish Certainty That Landmarks Cause Jumps

This chapter explains what the jump phenomena is, how it can be observed, and what has been done to determine that it is a byproduct of PLL.

2.1 Jump observation

To observe a jump, an x-y stage and a tilting rotary table are used. The x-y stage is mounted as near to the center of the rotary table as possible. A target is then mounted near to the center on the x-y stage as seen in Figure 8. The x-y stage can be used to adjust the location of the target. The target is positioned such that the X and Y measurements are as close to constant as possible during a roll of the rotary table. In this configuration the X, Y and Z measurements should be very steady throughout the rest of the experiment. The experiment is intended to create as true a roll about the Z axis in Figure 9 as possible. A recorded jump can be seen in Figure 10. An example of a rotation that is not experiencing a jump can be seen in Figure 11.

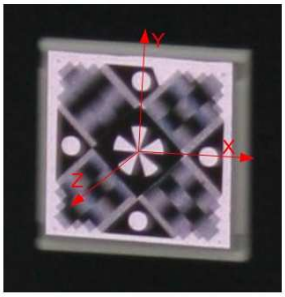

Figure 9: MPT target with X, Y and Z axes represented as the right, top, and out of page planes, respectively.

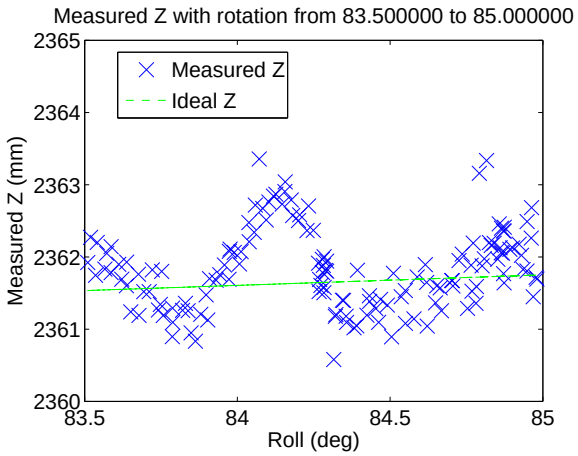

Figure 10: Observed jump phenomenon where a clear bias can be observed around 84 degrees.

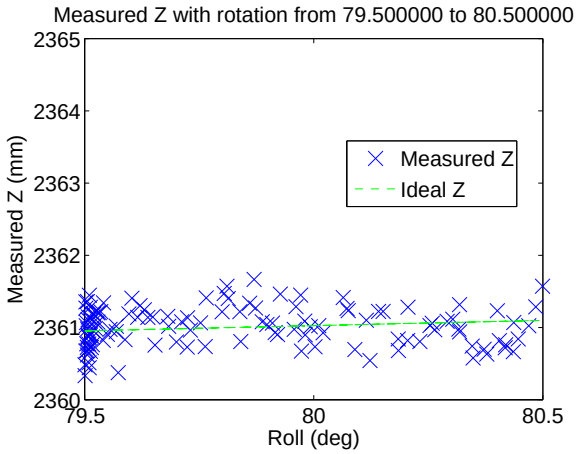

Figure 11: Observation of data without a jump occurrence. All errors appear to be random and show no significant patterns.

2.2 Landmark path interpolation

A numerical experiment was performed by Brian Armstrong that involved a path fitting interpolation algorithm. The algorithm allowed for any, or all, of the parameters to be estimated. The estimation fit was quadratic, and the parameters were the center starburst, the four circular landmarks, and each of the moiré patterns. It was seen that interpolating the four circular landmarks reduced the errors in Z from an RMS of 0.466 mm to .016 mm. Interpolating more of the parameters did decrease the RMS values, but only the combination of the circular landmark estimation resulted in such a significant improvement.

The conclusion of this experiment was that the deviations of the location of the circular landmarks plays a significant role in the jump seen.

2.3 Statistical investigation

An experiment was performed to see if there are any direct correlations between landmark location on an imager, roll angle, and a jump.

The process performed is as follows

1. Locate a jump per Section 2.1, mark the locations of the XY Stage and the roll angles.
2. Create a grid of points to be observed with variables X-Stage, Y-Stage, and Roll Angle. Acquire an image in each configuration.
3. Process the images.
4. Investigate the Z measurement in different scenarios.

   (a) Does the jump happen when the target is only translated?

      i. If so, the phenomena could be an artifact of pixel noise on the imager.

   (b) Does the jump happen at a specific angle, regardless of location?

      i. If so, the phenomena could be an artifact of the target or imager, but not related to a specific pixel.

5. Investigate which, if any, predictors correlate with the measured Z.

The initial investigation from step 4 involved graphing and observing the data in MATLAB. Jumps were observed during both rotation *and* translation. A conclusion was made that *the jump is not an artifact of target orientation alone.*

A statistical regression proposed by Montgomery et al. [29] was used to see if any predictors could model a jump. This is an important step because predictors can lead to a better understanding of the phenomena with respect to multiple dimensions of data. Ideally this understanding would lead to correction. Regressors investigated were

1. The roll angle of the rotary table
2. The first and second order terms of the locations from the X and Y stage
3. The first and second order terms of the estimated location of the target
4. The pitch, roll, and yaw location estimations of the target.

The results are included in Table 1. There are a few things that can be determined by this table.

1. There isn't a direct correlation between roll and a jump.

   (a) Table Roll isn't an effective predictor until 10 of the 15 parameters are used.

2. No subset of the predictors given can provide accurate estimation of a jump; one can not predict a jump given this information.

   (a) The best values of R-squared are around 50% when all of the predictors are used. A sufficient model would have to predict this much more accurately and with fewer variables.

Further regression techniques were attempted with the pixel locations of each landmark. The conclusion that could be made was that the phenomena did not depend on specific locations of the target in space nor specific pixels in the imager.

Vars	R-Sq	R-sq(adj)	Cp	S	XStage	YStage	XStage^2	YStage^2	Radius	Radius^2	X*YStage	X	Y	X^2	Y^2	Pitch	Roll	Yaw	Table Roll
1	8.2	7.5	78.7	.96							X								
1	4.8	4.1	86.1	.97			X												
2	14.4	13	67.3	.93			X							X					
2	12.8	11.4	70.8	.93			X				X								
3	16.4	14.3	65.1	.92			X		X					X					
3	16.3	14.2	65.3	.92			X		X		X								
4	19.8	17.1	59.7	.90	X						X	X	X						
4	16.6	13.8	66.5	.92			X		X				X	X					
5	25	21.8	50.5	.88	X		X					X	X	X					
5	24	20.8	52.5	.88	X		X				X	X	X						
6	33.3	29.9	34.4	.83	X					X	X	X	X		X				
6	29.8	26.2	42	.85	X				X		X	X	X		X				
7	41.3	37.7	19.2	.78	X					X	X	X	X	X	X				
7	40.4	36.8	21	.79	X		X			X		X	X	X	X				
8	43.1	39.2	17.1	.77	X				X	X	X	X	X	X	X				
8	42.9	38.9	17.7	.78	X	X				X	X	X	X	X	X				
9	44.7	40.3	15.7	.77	X	X			X	X	X	X	X	X	X				
9	44.6	40.2	16	.77	X	X				X	X	X	X	X	X			X	
10	47.9	43.3	10.9	.75	X	X			X	X	X	X	X	X	X				X
10	47.6	42.9	11.6	.75	X	X			X	X	X	X	X	X	X		X		X
11	49.0	44	10.5	.74	X	X		X	X	X	X	X	X	X	X				X
11	48.9	43.9	10.7	.74	X	X	X	X	X	X		X	X	X	X				X
12	50.0	44.6	10.4	.74	X	X	X	X	X	X	X	X	X	X	X				X
12	49.7	44.3	10.9	.74	X	X		X	X	X	X	X	X	X	X		X		X
13	50.0	44.1	12.2	.74	X	X	X	X	X	X	X	X	X	X	X	X			X
13	50.0	44.1	12.3	.74	X	X	X	X	X	X	X	X	X	X	X		X		X
14	50.1	43.7	14.1	.74	X	X		X	X	X	X	X	X	X	X	X	X	X	X
14	50.1	43.6	14.1	.74	X	X	X	X	X	X	X	X	X	X	X	X	X		X
15	50.1	43.2	16	.75	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Table 1: Minitab best subset regression. Surprisingly low correlation between all regressors was seen and almost no correlation between roll angle and measured Z was observed.

2.4 Summary

This chapter shows that there is a jump phenomena that happens and how to find jumps. The jump is not an artifact of target orientation alone. No set of predictors can accurately model a jump. An initial theory that small locations on the target were brighter than other locations is discredited because jumps occur regardless of exact target location. The jump also doesn't have any connection with landmarks falling on specific locations of the pixels, which contradicts theories of fixed pattern noise of an image sensor playing a direct role. Instead, there is some interaction between the target and image sensor that requires a deeper investigation. As a result, simulation techniques described in Chapter 3 and analysis of the current routines in Chapter 4 verify that PLL is the cause of the jump and that the corrective techniques investigated in Chapters 6 and 7 can significantly reduce the jump, as seen in Chapter 8.

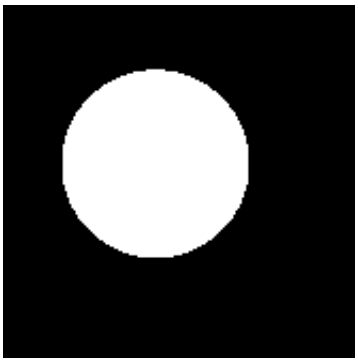

Figure 12: A binary image created as the first step of landmark simulation.

3 Image Simulation

To validate our suspicion that landmark estimation bias was causing the jump, further investigation into circular landmark estimation was needed. Using simulated data instead of experimental data, one knows the exact specifications of the image and can compare that with the estimations. Image Simulation consists of multiple steps, each of which will be described and explained in this chapter.

3.1 High resolution image creation

The first step to creating an artificial landmark is to create a binary, high resolution image. A set of tools were created and used to perform this task. A function would compare a given pixel to a defined ellipse. If the center of the pixel is inside the ellipse, that pixel receives a value of 255 (white). If the center of the pixel is outside the ellipse, the value is 0.

The final result of this is shown in Figure 12.

The next step proposed by numerous authors [16, 20] is to use a 2D Gaussian Kernel convolution to create the illuminance function. The Gaussian Kernel is defined in equation 3

$$Gaussian\,Kernel = \frac{1}{2\pi\sigma_b^2} e^{-\frac{x^2+y^2}{2\sigma_b^2}} \tag{3}$$

This was used to create an appropriately sized[1] grid. The grid was normalized so that the integral of the grid was equal to 1. The next step was to convolve the two matrices. The result of

[1] Appropriately sized means that the points nearest the outside edge are all very close to zero and not still a significant part of the Gaussian Kernel

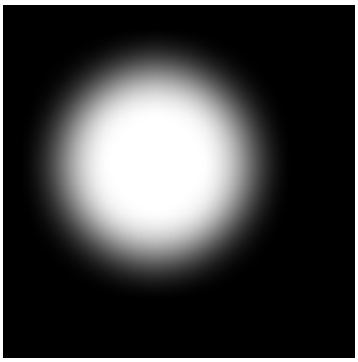

Figure 13: A blurred high resolution image created to simulate the blurring effects of optical systems.

this operation is shown in Figure 13

3.2 Low resolution image creation

After the blurred high resolution image has been created, it must be reduced to a low resolution image. There were two different ways to perform this task that are described in this section.

Pixel averaging The first method of low resolution image generation is to take the average intensity over the area of the pixel. This method would accurately simulate one type of 'ideal' imager; the sensitive area of the pixel is 100% of the pixel area. Real imagers do not have 100% sensitive area and are often in complex shapes [20]. The equation used is shown in equation 4. The function $I(\mathbf{p})$ is called the intensity profile, a continuous function representing the intensity of light at a point $\mathbf{p}$.

$$I_{LR}(\mathbf{p}) = \iint I_{HR}(\mathbf{p})\, d\mathbf{p} \tag{4}$$

Unless explicitly stated, all calculations were done with the low resolution image created by this method.

Pixel sampling The second method of low resolution image generation is one I will call pixel sampling. This method would simulate a different type of 'ideal' imager; one where the sensitive area is 0% of the pixel area and only a single point at the center of the imager. Pixel sampling

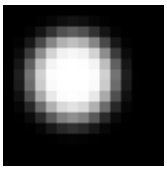

Figure 14: The final simulated low resolution landmark. Landmarks like these were used during all of Part II.

was done by assigning the value of the low resolution pixel to be equivalent to the center of the corresponding high resolution pixel. For example, if the high resolution image were 15 times larger than the low resolution image, the value of the low resolution pixel $[1, 1]$ would be the value of the high resolution pixel $[6, 6]$. Equations 5 and 6 show the homogeneous transformation equivalents of this. The equation used is equation 7 and the final product appears indistinguishable from Figure 14.

$$ {}^{HR}_{LR}T = \begin{bmatrix} 11 & 0 & -5 \\ 0 & 11 & -5 \\ 0 & 0 & 1 \end{bmatrix}, \; {}^{LR}_{HR}T = \begin{bmatrix} .0909 & 0 & .4545 \\ 0 & .0909 & .4545 \\ 0 & 0 & 1 \end{bmatrix} \tag{5} $$

$$ \mathbf{p}_{LR} = \begin{bmatrix} p_{x,LR} \\ p_{y,LR} \\ 1 \end{bmatrix} = {}^{LR}_{HR}T \begin{bmatrix} p_{x,HR} \\ p_{y,HR} \\ 1 \end{bmatrix} \tag{6} $$

$$ I_{LR}(\mathbf{p}) = I_{HR}(\mathbf{p}_0) \tag{7} $$

The last step to create a fully simulated low resolution image is to quantize the image to an 8-bit integer. This was done by simply using a round() command built into MATLAB. The final products appear indistinguishable from Figure 14.

After the simulated images were created we were able to run the current center location algorithms as well as develop and test new algorithms. These algorithms could be modified and tested against the true values because of the fact that the data was simulated.

Part II

Numerical Investigation

4 Initial Interpolation Method

A need to investigate the accuracy and properties of the current interpolation technique was necessary. This could uncover model inaccuracies and give way to additional information regarding landmark estimation. The original technique used is explained, and the results from performing this technique are analyzed in this chapter.

4.1 Linear interpolation

Linear interpolation was the initial technique used for ellipse fitting. The linear interpolation consists of two steps: First a set of edge points are determined, then the edge points are fit to an ellipse. The first step is what is being investigated.

The contour point estimation algorithm used a two step process. It finds transition points, then interpolates between those points. Edge points are where one pixel is above a certain threshold and an adjacent pixel is below the threshold. This process is repeated until all possible edge points are discovered. Once the points are located, the algorithm fits a straight line connecting the values of the interior and exterior points. The algorithm solves for the intersection between the fitting line and the threshold. This intersection is the location of the estimated contour point. A graphic of this is shown in Figure 15.

Mathematical explanation Let $\mathbf{p}_i$ be the interior point with coordinates

$\mathbf{p}_i = (x, y) \in LowResolutionPixelCoordinates$

and $\mathbf{p}_e$ be the exterior point with the intensity profile $I(\mathbf{p})$

The coordinates of $\mathbf{p}_e$ must be one of the following $[(x+1, y), (x-1, y)(x, y+1), (x, y-1)]$. Put another way,

$$\mathbf{p}_e = \mathbf{p}_i + \Delta\mathbf{p} \tag{8}$$

where $\Delta\mathbf{p} \in [(1, 0), (-1, 0)(0, 1)(0, -1)]$. The threshold intensity is Th is known to be such that $I(\mathbf{p}_e) \leq Th < I(\mathbf{p}_i)$.

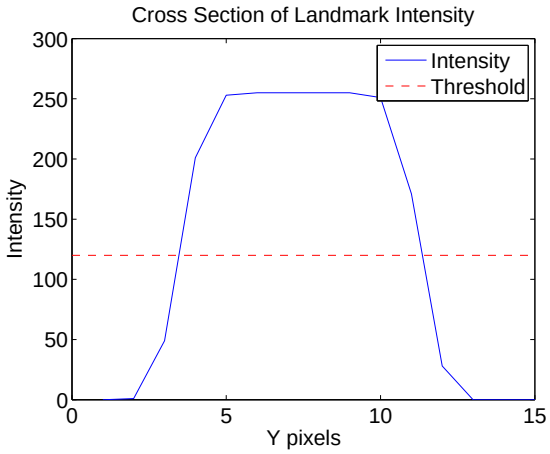

Figure 15: Cross section of a landmark and an intensity threshold. The line intersections are the estimated contour points.

A line can be fit to I and $\Delta\mathbf{p}$ where

$$I(\mathbf{p}_\alpha) = I(\mathbf{p}_i) + \frac{I(\mathbf{p}_e) - I(\mathbf{p}_i)}{|\Delta\mathbf{p}|} * \alpha\Delta\mathbf{p} \tag{9}$$

and $\mathbf{p}_\alpha = \mathbf{p}_i + \alpha\Delta\mathbf{p}$. The equation can be solved for the intersection by setting $I(\mathbf{p}_\alpha) = Th$ and solving for $\mathbf{p}_\alpha$.

The final equation is

$$\mathbf{p}_\alpha = \mathbf{p}_i + (Th - I(\mathbf{p}_i)) \frac{|\Delta\mathbf{p}|}{I(\mathbf{p}_e) - I(\mathbf{p}_i)} \Delta\mathbf{p} \tag{10}$$

which will give a unique solution $\mathbf{p}_\alpha$ in the range $(\mathbf{p}_i, \mathbf{p}_e]$ along $\Delta\mathbf{p}$. The value $\mathbf{p}_\alpha$ is the location of the estimated contour point.

Example In a case where there are nine pixels, say $\begin{bmatrix} 255 & 135 & 81 \\ 145 & 72 & 0 \\ 126 & 15 & 18 \end{bmatrix}$, and a threshold $Th = 120$, there four cases where the intensity is greater than the threshold and five cases where the intensity is less than the threshold. Using standard matrix numbering, we can see that the intensity of the pixel located at $(2,1)$ is 135 and the intensity at pixel $(3,1)$ is 81. This is equivalent to saying $I(2,1) = 135$ $I(3,1) = 81$. Interpolation can be done between these two points to estimate an intensity equal to the threshold.

$$\begin{bmatrix} x \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix} + (120 - 135) \frac{1}{81-135} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

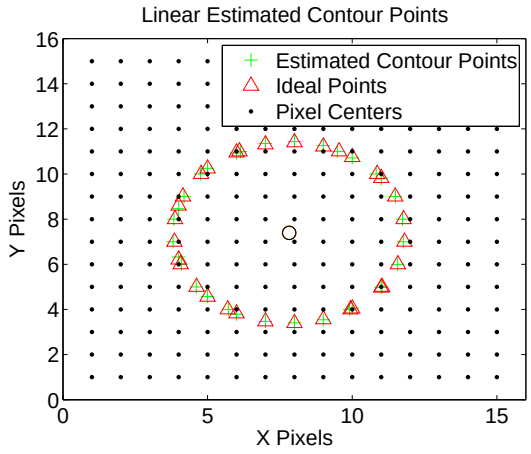

Figure 16: First order ideal and estimated contour points. Both the estimated and ideal points correctly resemble a circle.

Solving for x, gives the estimated contour point of $(2.2778, 1)$.

Similarly the vertical crossover point between 135 and 72 is $\begin{bmatrix} 2 \\ y \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix} + (120 - 135)\frac{1}{72-135}\begin{bmatrix} 0 \\ 1 \end{bmatrix}$.

This gives a second estimated contour point of $(2, 1.2381)$.

Doing this for all other transition points, $(2, 1)$ to $(2, 2)$, and $(3, 1)$ to $(3, 2)$ yields a third and fourth estimated contour points of $(2, 1.3425)$ and $(3, 1.0541)$.

Ellipse fitting The second step is to fit the points to an ellipse. A Newton-Raphson algorithm is used to solve for the least squares fit of an ellipse. Data regarding a Newton-Raphson Algorithm can be found in Section 7.1.1.

4.2 Observations

There are a number of techniques to extract data from the interpolation methods. Section 4.2.1 discusses observations that can be made from looking at individual fits. Section 4.2.2 discusses observations from looking at thousands of images. Both techniques are very useful when investigating the accuracies of different methods.

4.2.1 Looking at a single image

The edge points from linear interpolation can be seen from a single landmark location in Figure 16, with a zoomed in section located in Figure 17. The ideal contour points (shown as triangles) are the result of linear analysis on the high resolution image, and are considered to be the ideal contour point locations.

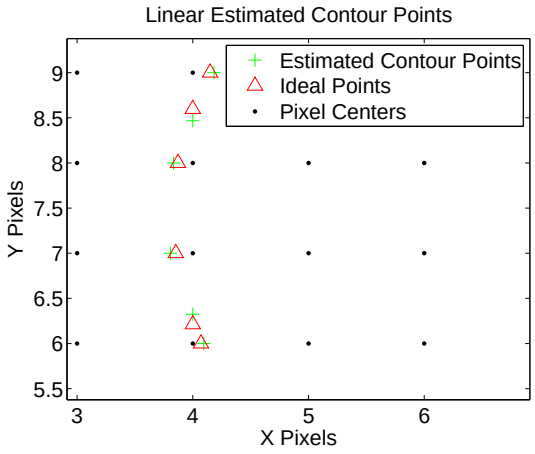

Figure 17: Zoomed in first order ideal and estimated contour points. Differences of varying degrees between the ideal and estimated contour points can be seen in this example.

It can be seen in Figure 17 that the estimated contour points are often interpolated further away from the nearest pixel center. This effect was almost universal with the linear estimation procedure. An hypothesis that came from this observation was that if the estimation procedure could artificially "pull" the estimated contour points toward the nearest pixel, a better center estimation would be the result. This hypothesis led to different corrective techniques, discussed in Chapter 6.

4.2.2 Looking at thousands of images

A MATLAB function was written to loop each method over many different cases. Most of the investigation involved looking into a circular landmark with a radius of 4 pixels and the center of the landmark occupying a 101x101 grid across an entire pixel. Pseudocode of the specific loop would be

```
Index = 0;
for XCenter = 7.00:.01:8.00
for YCenter = 7.25:.01.8.25
CalculateCenterOfImage(XCenter, YCenter);
end; end;
```

The locations of the landmarks were estimated in each case and the deviations from each case are plotted in Figure 18. The main observations will be shown and discussed in the following parts.

Magnitude of the center offsets One feature of Figure 18 is the significant deviations between the measured and true landmark centers. A simple standard deviation calculation from the data

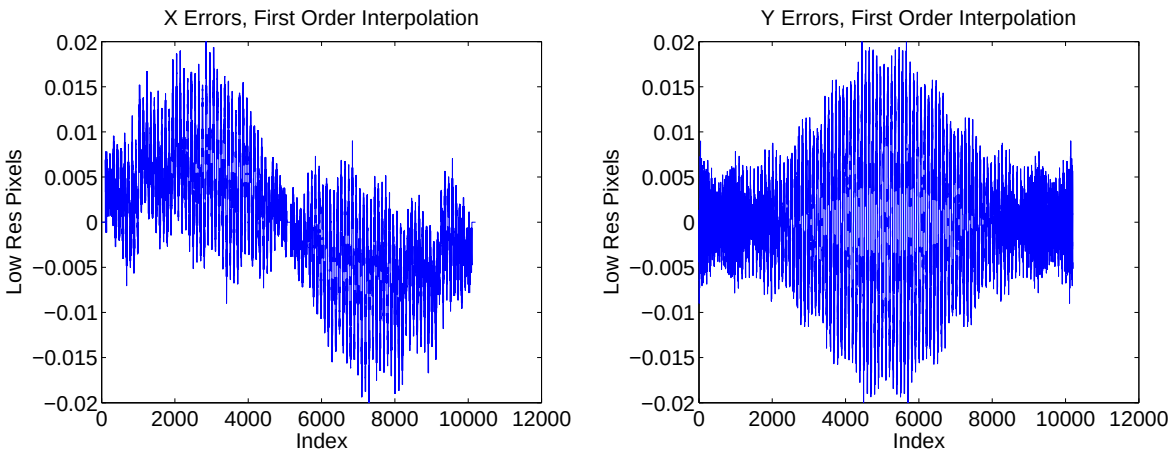

Figure 18: Observed errors from the linear interpolation method.

produced by the pseudocode in 4.2.2;

$$\sigma_{\tilde{y}} = \sqrt{\frac{\sum(\tilde{y}^2)}{n-1}} \tag{11}$$

$$[\tilde{x}_{max}, \tilde{y}_{max}] = [max(|\tilde{x}|), max(|\tilde{y}|)] \tag{12}$$

where $\tilde{y}$ is the difference between the estimated and measured locations, and n is the number of data points. It is seen that $[Dev_x, Dev_y] = [.0067, .0068]$ with maximum offsets in x and y to be .0207 in both cases.

Periodicity of the center offsets Figure 19 shows an apparent periodicity of the errors. The fact that the estimation is periodic and apparently much greater than any higher frequency changes observed supports the conclusion that bias in the estimation process exists.

4.2.3 Could this observed bias be related to the jump?

An experiment to test whether there could be a correlation between landmark location bias and the jump phenomenon was devised. This included a four step procedure.

1. Take an existing image of a target
2. Use the current methods of landmark location to calculate the four circular landmark locations
3. Shift each of the landmarks radially by a magnitude in pixels, comparable to the bias shown by the simulation
4. Observe the shift in Z

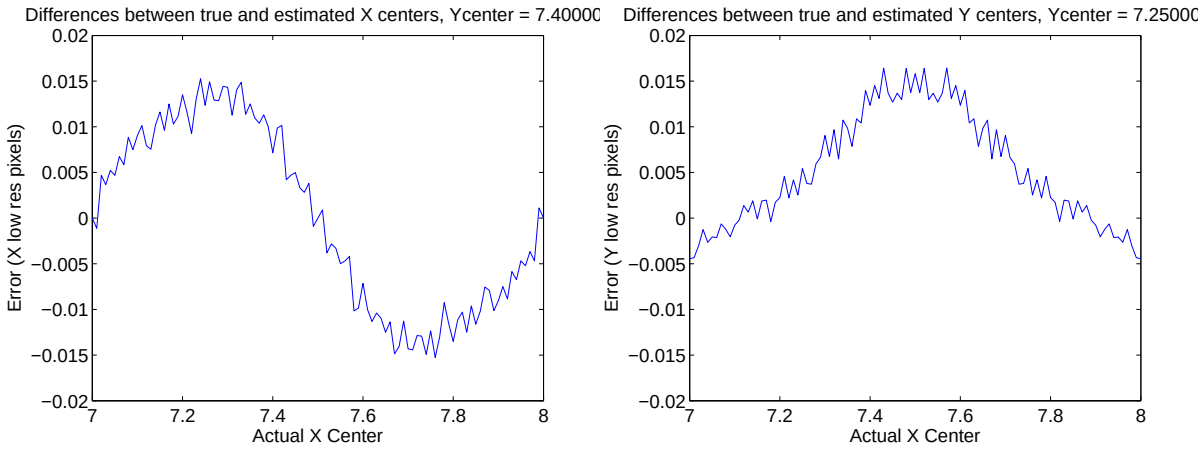

Figure 19: An example of the periodic errors observed in the data. When the center of the landmark was translated across a pixel in one dimension, periodic trends became apparent in most cases.

Step 3 involved taking the initial locations of the four landmarks, call them $\mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4, \mathbf{p}_5$ where $\mathbf{p}_2$ is across from $\mathbf{p}_4$ and $\mathbf{p}_3$ is across from $\mathbf{p}_5$. Two lines were created from these two points, $\overline{\mathbf{p}_{24}}, \overline{\mathbf{p}_{35}}$. Additionally, circles c_2, c_3, c_4, c_5 were created around each point with a radius near the maximum deviations observed, namely $[.02, .02]$ pixels, or a radius of $.02\sqrt{2}$ pixels.

Two sets of data were constructed using new, simulated positions of each point, $\hat{\mathbf{p}_2}, \hat{\mathbf{p}_3}, \hat{\mathbf{p}_4}, \hat{\mathbf{p}_5}$. The first set used the interior intersection of $\left(\overline{\mathbf{p}_{24}}, c_2\right), \left(\overline{\mathbf{p}_{24}}, c_4\right), \left(\overline{\mathbf{p}_{35}}, c_3\right), \left(\overline{\mathbf{p}_{35}}, c_5\right)$ and the Z measurement was observed. The second set used the exterior intersections of these points. The initial observed measurement was $Z_0 = 1090.870$ mm, with the inner and outer measurements being $[Z_{in}, Z_{out}] = [1091.636, 1089.151]$ mm. The magnitude of the particular jump in the experimental data was a range of Z from $[1089.441, 1091.053]$ mm.

The conclusion drawn from this experiment was that the magnitude of the estimation bias *could* be a main contributor to the Jump.

4.2.4 Bias vs noise

To quantify the effect of noise vs bias, an experiment was devised in which random pixel noise was added to each pixel of an image. Estimation was performed on the noisy images and the results were quantified with a standard deviation calculation.

It was found that random pixel noise on the order of 5 counts were necessary in order to double the standard deviation of the errors. The plot of the errors is shown in Figure 20. Imagers can be set such that pixel noise is on the order of one count for appropriate signal-to-noise characteristics. It can therefore be concluded that *the effect of bias is much greater than the effect of noise on precision*

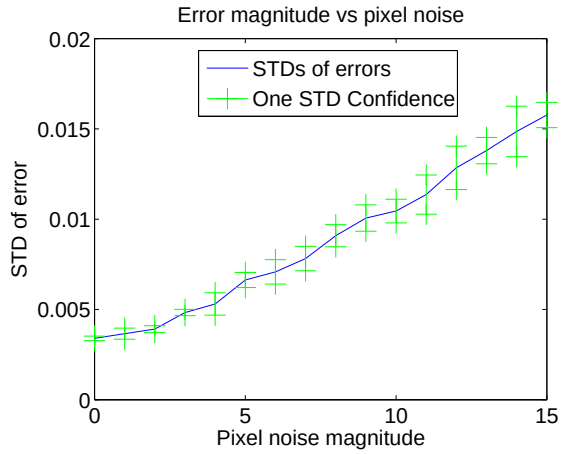

Figure 20: Effects of pixel noise on the observed errors.

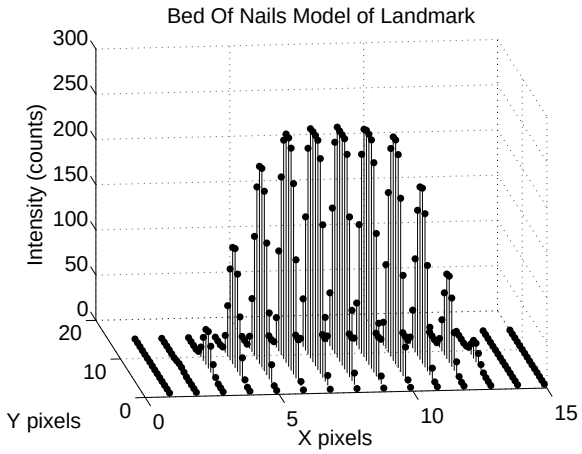

Figure 21: Bed of nails model of a landmark.

landmark location.

4.2.5 The bed of nails model

Pixels only offer two pieces of information: intensity and position. No information about how light is distributed across the pixel is available. Due to these limiting quantization factors of imagers, the only model that can be directly observed is called a bed of nails model, shown in Figure 21.

In the case of PLL, the intensity profile, or distribution of light across a pixel, is complex. It is such that the recorded intensity of the pixel is not equal to the actual intensity at the center of the pixel. Interpolation methods that use information from two inaccurate pixels are inherently inaccurate, and PLL is sensitive to errors of this magnitude. Chapter 6 uses techniques to gather information from more pixels in order to minimize the errors seen from this phenomenon

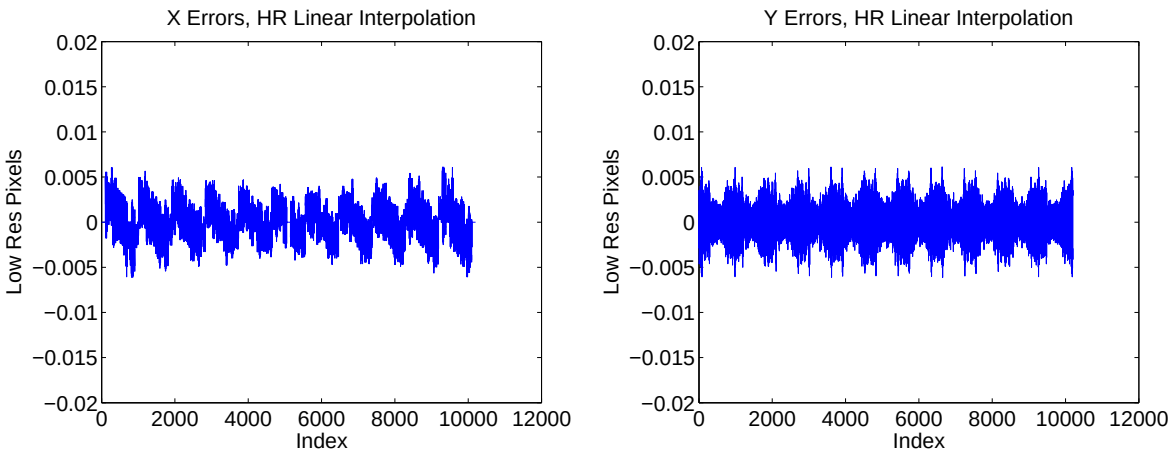

Figure 22: Deviations from high resolution image analysis. This case had 11 times greater linear resolution in the high resolution case.

5 'Perfect' High Resolution Interpolation Techniques

This chapter investigates the possibility of improving estimation if one were given more precise data. It addresses the question of whether or not better image data and/or modelling techniques can improve the interpolation techniques.

5.1 High resolution interpolation

Interpolation was performed on the high resolution images created in Section 3. The process involved using the interpolation axes, $\Delta\mathbf{p}$, from Section 4.1. The data used for interpolation, however, was not subjected to the averaging, sampling, or quantization of the low resolution image. \

The observed deviations form the linear interpolation method were $[Dev_x, Dev_y] = [0.0067, 0.0068]$ mm and a maximum of 0.0207, as seen in Chapter4. By using the information in the high resolution image, the deviations shown in Figure 22 dropped to $[0.001988, 0.002012]$ mm with a maximum of 0.0064 mm. *This is a reduction in errors by a factor of 3.* Table 2 shows this information, as well as the results from Chapter 7.

Could the bias be completely removed?

Our estimation process involved the creation of a high resolution image with a multiplier of 11x11 high resolution pixels per low resolution pixel. It was then investigated whether a larger multiplier could further reduce the errors. By running the same simulation with 33x33 high resolution pixels, the deviations were seen to drop to a magnitude of $[.0004, .0004]$, *a factor of 50 improvement.* The results of this investigation are shown in Figure 22. It was concluded that with enough data regarding

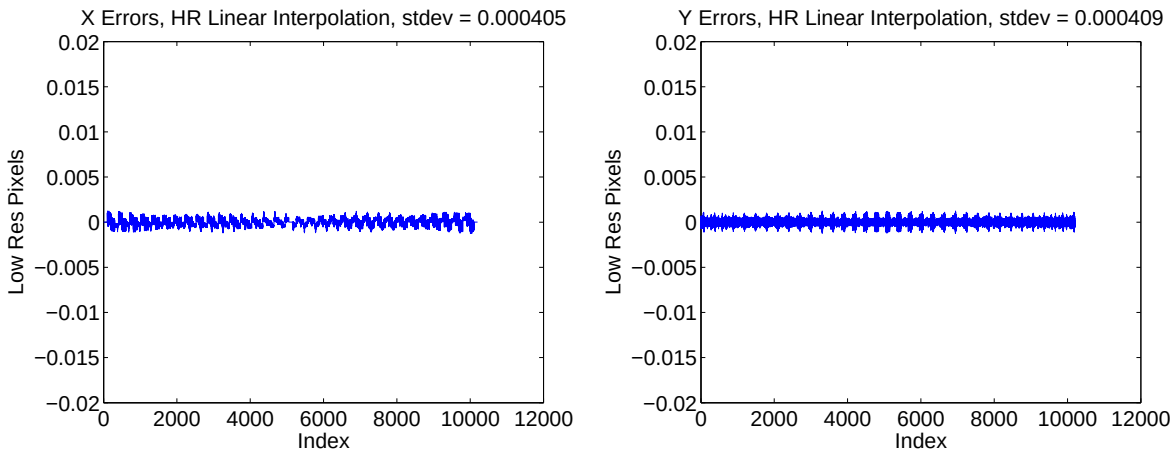

Figure 23: Deviations from the high resolution image analysis with 33 times greater linear resolution in the high resolution case.

the pixel intensity profile, much more precise measurements could be performed.

W*ith sufficiently accurate estimated contour points, the bias could be removed.*

5.2 Spline interpolation

Spline interpolation, described in Section 6.1.3, was performed on the high resolution images in order to create more accurate ideal contour points. The motivation for using this method on high resolution images was that if interpolation led to improvements on high resolution images, it could then be applied to the low resolution images. The spline fit from the four points appears very linear and the points did not deviate significantly from the first order ideal contour points. It can be seen that the deviations in Table 2 show no differences from the linear interpolation case. As a result, this technique was not used for low resolution analysis.

6 Approaches To An Unbiased Estimate Of The Contour Points

Chapters 4 and 5 show that the investigation of more accurate estimated contour points has significant potential for benefit. As a result, several methods were investigated in this chapter. Section 6.1 describes the techniques used and Section 6.2 shows the results of the techniques.

6.1 Techniques

Several different interpolation techniques were evaluated. Some of the techniques used the data from the low resolution images to act as corrective methods, while others used data from the high resolution images and were designed to be more analytical with the possibility of leading to corrective methods. Each of the methods used are described in this section and the results of each method are explained in Section 6.2.

6.1.1 Quadratic interpolation

The quadratic interpolation method is similar to the linear interpolation. The algorithm fits estimated contour points, then fits an ellipse to the estimated contour points. There are some subtle logic changes and some significant contour point estimation algorithm changes.

First, suitable crossover points needed to be determined. For this method, two interior points were chosen and one exterior point. Unsuitable crossover points are found near the points where the direction vector $\Delta\mathbf{p}$ is near to the tangent of the ellipse at that point. In these cases one could only find three adjacent points with the characteristic *Outer Point* $\rightarrow$ *Inner Point* $\rightarrow$ *Outer Point* where *Inner Points* have intensities greater than the threshold, and *Outer Points* have intensities below the threshold. Cases like these were not common, but when they did occur the estimator would use linear interpolation.

Once the suitable points are discovered, a quadratic is fit to the data. The technique used is a Second Order Newton's Divided-Difference Interpolating Polynomial [30].

A second case was considered for interpolation. Two outer points and one inner point were used for interpolation. These two cases are shown in Figures 24 and 25 and show the comparison with the first order interpolation.

Mathematical explanation

Let $\mathbf{p}_{i1}, \mathbf{p}_{i2}$ be the two internal points and $\mathbf{p}_e$ be the external point such that $\mathbf{p}_e = \mathbf{p}_{i2} + \Delta\mathbf{p}$ and $\mathbf{p}_{i1} = \mathbf{p}_{i2} - \Delta\mathbf{p}$.

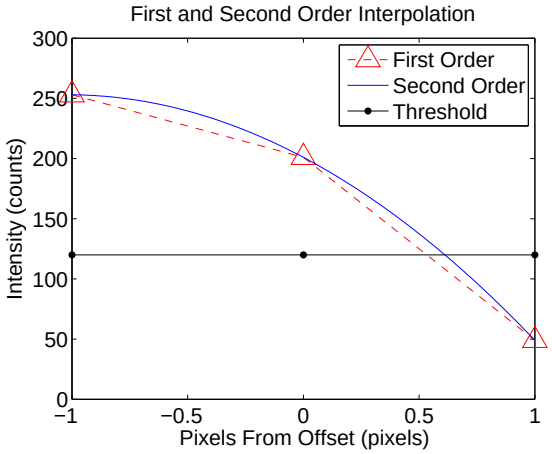

Figure 24: Original second order interpolation method. In this case two points were used from above the threshold and one was used below the threshold.

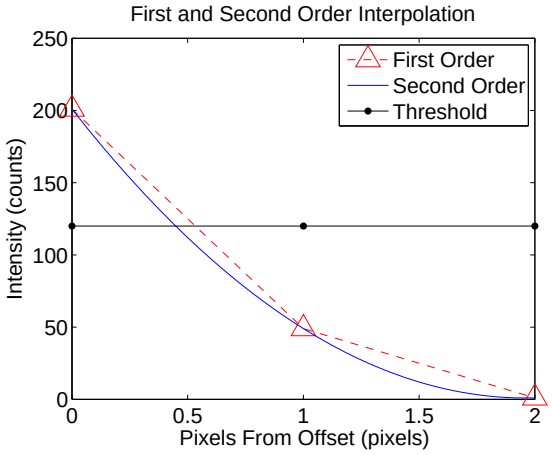

Figure 25: Alternative second order interpolation method. This case used two points below the threshold and one point above the threshold.

Then

$$I(\alpha\Delta\mathbf{p}) \approx I(\mathbf{p}_{i2}) + \frac{I(\mathbf{p}_{i1}) - I(\mathbf{p}_{i2})}{\mathbf{p}_{i1} - \mathbf{p}_{i2}}(\alpha\Delta\mathbf{p} - \mathbf{p}_{i2}) + \frac{\frac{I(\mathbf{p}_e)-I(\mathbf{p}_{i1})}{\mathbf{p}_e-\mathbf{p}_{i1}} - \frac{I(\mathbf{p}_{i1})-I(\mathbf{p}_{i2})}{\mathbf{p}_{i1}-\mathbf{p}_{i2}}}{\mathbf{p}_e - \mathbf{p}_{i2}}(\alpha\Delta\mathbf{p} - \mathbf{p}_{i2})(\alpha\Delta\mathbf{p} - \mathbf{p}_{i1}) \tag{13}$$

Simplified, with $b_0 = I(p_{i2})$, $b_1 = \frac{I(\mathbf{p}_{i1})-I(\mathbf{p}_{i2})}{\mathbf{p}_{i1}-\mathbf{p}_{i2}}$, and $b_2 = \frac{\frac{I(\mathbf{p}_e)-I(\mathbf{p}_{i1})}{\mathbf{p}_e-\mathbf{p}_{i1}} - \frac{I(\mathbf{p}_{i1})-I(\mathbf{p}_{i2})}{\mathbf{p}_{i1}-\mathbf{p}_{i2}}}{\mathbf{p}_e-\mathbf{p}_{i2}}$ we have

$$I(\alpha\Delta\mathbf{p}) = b_0 + b_1(\alpha\Delta\mathbf{p} - \mathbf{p}_{i2}) + b_2(\alpha\Delta\mathbf{p} - \mathbf{p}_{i2})(\alpha\Delta\mathbf{p} - \mathbf{p}_{i1}) \tag{14}$$

We can then let $a_0 = b_0 - b_1\mathbf{p}_{i2} + b_2\mathbf{p}_{i1}\mathbf{p}_{i2}$, $a_1 = b_1 - b_2\mathbf{p}_{i2} - b_2\mathbf{p}_{i1}$, and $a_2 = b_2$ so that

$$I(\alpha\Delta\mathbf{p}) = a_0 + a_1\alpha\Delta\mathbf{p} + a_2(\alpha\Delta\mathbf{p})^2 \tag{15}$$

Setting $I(\mathbf{p}_\alpha) = Th$ and using the quadratic formula, and using the relation $\mathbf{p}_\alpha$=$\mathbf{p}_{i2} + \alpha\Delta\mathbf{p}$. we arrive at

$$\mathbf{p}_\alpha = \mathbf{p}_{i2} + \frac{-a_1 \pm \sqrt{a_1^2 - 4a_2(a_0 - Th)}}{2a_2}\Delta\mathbf{p} \tag{16}$$

The algorithm tests the positive radical case first, $\alpha\Delta\mathbf{p}_1 = \frac{-a_1+\sqrt{a_1^2-4a_2(a_0-Th)}}{2a_2}$ If α falls in $(0, 1]$ then the solution is $\mathbf{p}_\alpha = \mathbf{p}_{i2} + \alpha\Delta\mathbf{p}_1$. Otherwise the second case is verified to fall along the correct range. It then accepts the solution $\alpha\Delta\mathbf{p}_2 = \frac{-a_1-\sqrt{a_1^2-4a_2(a_0-Th)}}{2a_2}$ and $\mathbf{p}_x = \mathbf{p}_{i2} + \alpha\Delta\mathbf{p}_2$.

These points were then used in the same Newton Raphson Ellipse Fitting Algorithm mentioned in Section 4.1.

6.1.2 Sigmoid method

As seen in Figure 17, the estimated contour points have a tendency to be pushed further away from the nearest pixel center. A technique was developed to combat this phenomena by correcting the interpolations with a sigmoidal function instead of a linear one. Sigmoid functions can be used to improve edge detection in images [31].The sigmoid function used was

$$f(w; x) = (1 - w) * 0.5 + \frac{tanh(2\pi(x - .5))}{2} + wx \tag{17}$$

and could produce functions of various sigmoidicity by modifying the linear weight parameter w. Examples of the varying sigmoidicity can be seen in Figure 26.

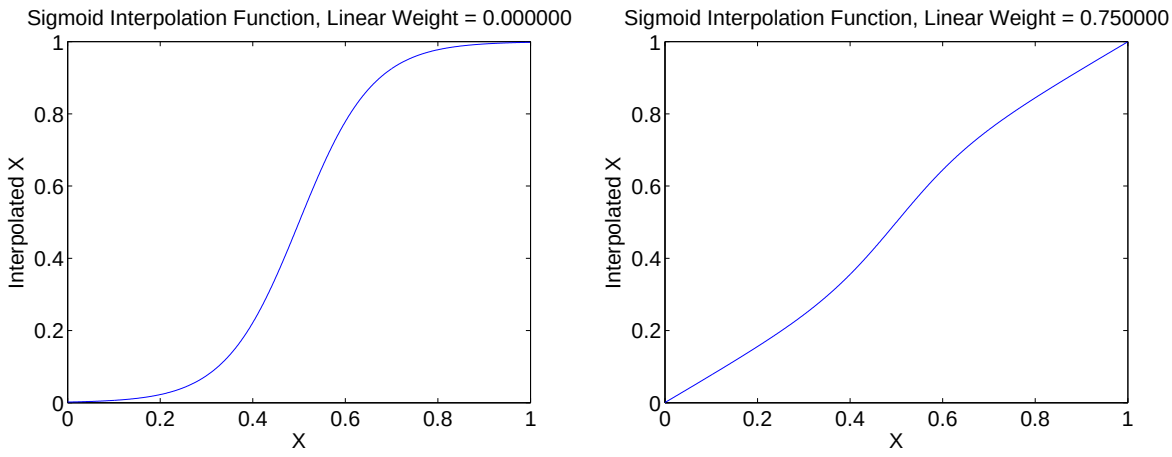

Figure 26: Varying sigmoid interpolation functions used to "pull" the interpolated landmarks toward the nearest pixel center.

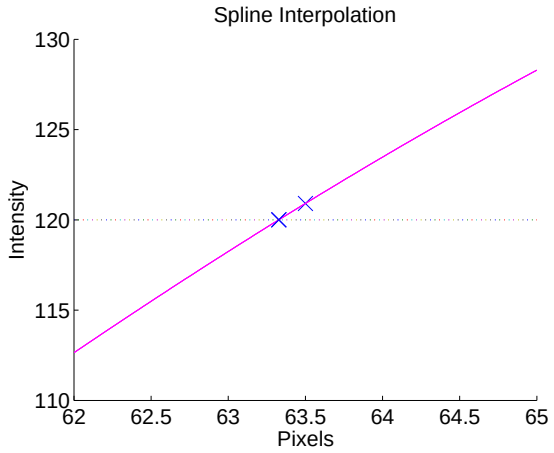

Figure 27: Example of a spline interpolation process used to try to improve high resolution interpolation accuracy.

6.1.3 Spline interpolation

Spline interpolation is similar to the quadratic interpolation of Section 6.1.1, but has a few distinct differences. First, it uses four points: two internal and two external, call them $\mathbf{p}_{i1}, \mathbf{p}_{i2}, \mathbf{p}_{e1}, \mathbf{p}_{e2}$. Second, it uses a Third Order Newton's Divided Difference Interpolating Polynomial. An example of the interpolation function can be seen in Figure 27.

Newton's Divided Difference Interpolating Polynomial is defined as

$$f_n(x) = b_0 + \sum_{i=1}^{n} b_i \prod_{j=0}^{i} (x - x_j) \tag{18}$$

$$b_0 = f(x_0)$$

$$b_1 = f[x_1, x_0]$$

where

$$f[x_i, x_j] = \frac{f(x_i) - f(x_j)}{x_i - x_j}$$

and

$$b_{n>1} = f[x_n, x_{n-1}, ..., x_1, x_0]$$

where

$$f[x_n, x_{n-1}, ..., x_1, x_0] = \frac{f[x_n, x_{n-1}, ..., x_1] - f[x_{n-1}, x_{n-2}, ..., x_0]}{x_n - x_0}$$

So $b_0 = I(p_{i1})$

$b_1 = \frac{I(p_{i2}) - I(p_{i1})}{p_{i2} - p_{i1}}$

etc.

This polynomial is then solved using a Secant Method loop. From using two set points, x_{-1}, x_0 the function determines $x_{i+1} = x_i - \frac{f_n(x_i)(x_{i-1} - x_i)}{f_n(x_{i-1}) - f_n(x_i)}$. When $\frac{x_{i+1} - x_i}{x_{i+1}} < \epsilon$ the loop has converged to the final point, and x_{i+1} is accepted as the estimated contour point.

6.1.4 Grid interpolation

A method to introduce surface curvature involved using a grid interpolation estimation procedure. The estimator would use a 3x3 grid in low resolution coordinates and fit a third order polynomial to the data. The algorithm would then interpolate along the same axis as the $\mathbf{\Delta p}$ axis in the Linear Interpolation 4.1 in order to locate a crossover point. A Secant Method loop was used to determine the final point. This calculation was only done using all of the data from the high resolution image.

Let $I(x, y) \approx f(x, y)$

$$f(x, y) = \left[1, x, y, x^2, y^2, xy, x^3, x^2y, xy^2, y^3\right] R \tag{19}$$

By defining $\mathbf{p}_0 = (x_0, y_0)$ and $\mathbf{p} = (x_i, y_i) = (x, y) - \mathbf{p}_0$, then

$$I(\mathbf{p}) \approx f(\mathbf{p}) : \mathbf{p} \in [(-1.5, 1.5), (-1.5, 1.5)] \tag{20}$$

A matrix can then be constructed from the known values.

$$A_{(i,j)} = \left[1, x_i, y_j, x_i^2, y_j^2, x_i y_j, x_i^3, x_i^2 y_j, x_i y_j^2, y_j^3\right] \tag{21}$$

$A_{(i,j)} \in \Re^{qx10}$ where $q = m*n*h^2$, m and n are the number of rows and columns of the simulated image and h is the linear resolution difference between the high and low resolution images.

The offset $\mathbf{p}_0$ was necessary because the matrix $A_{(i,j)}$ was ill-conditioned with large values of x and y. $A_{\mathbf{p}}$ was a well-conditioned matrix when x and y were in low resolution coordinates, so the inversion process used below is possible.

The known values of $I(\mathbf{p})$and $\mathbf{p}$ allow the least squares estimation for $R \in \Re^{10}$

$$R = \left(A^{'} A\right)^{-1} A^{'} I(\mathbf{p}) \tag{22}$$

We then have a fully defined function $f(\mathbf{p}) = A(A'A)^{-1} A'I(\mathbf{p})$ that is used as the prediction model.

A Secant algorithm is used to solve $f(\mathbf{p} + \alpha_i \Delta\mathbf{p}) = Th$ for the appropriate value of α_i. The value $\mathbf{p}_\alpha = \mathbf{p} + \alpha\Delta\mathbf{p} + \mathbf{p}_0$ is defined to be the estimated contour point.

It has been seen that sometimes the solution to $f(\mathbf{p} + \Delta\mathbf{p}) = Th$ would not exist or not be discovered by the Secant Method. These cases were undesirable and were ignored in Standard Deviation (Equation 11) and Maximum Error (Equation 12) calculations.

An example of a pixel grid shape can be seen in Figure 28. The figure shows the original and fit images, as well as a surface plot of the area. The black crosses in the image represent each estimation iteration through the secant method.

6.1.5 Pixel surface interpolation

Pixel Surface Interpolation is the same as Grid Interpolation except that the domain of A is limited to the area of a single pixel instead of a 3x3 grid. There were points where a solution couldn't be found, either because the algorithm would have to extrapolate to find a solution or the Secant Method couldn't converge. In these cases the single points were omitted and the ellipse fitting method was performed with fewer data points.

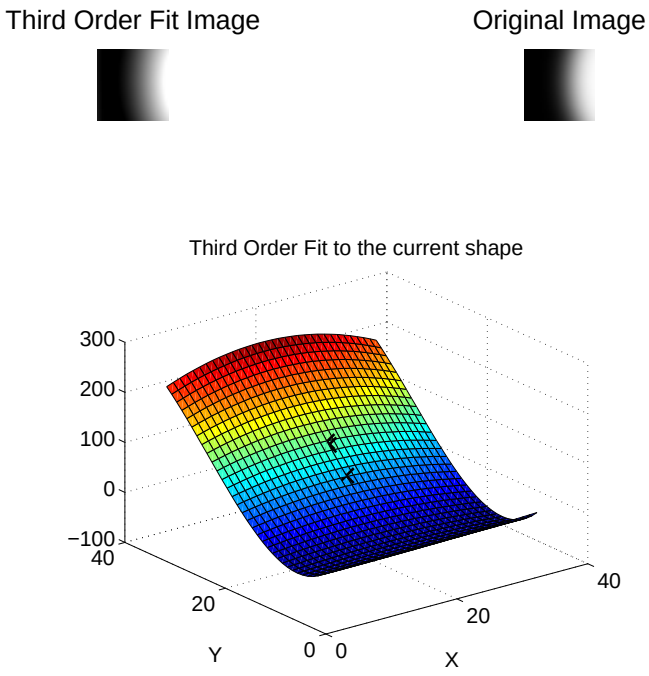

Figure 28: A pixel grid interpolation profile using 3x3 pixels and high resolution data.

An example of a pixel surface shape can be seen in Figure 29. The figure shows the interpolated surface across the pixel, as well as the black crosses, which represent each iteration through the secant method.

6.2 Results

All of the interpolation methods listed in this section were performed on a number of images. The specific images used are described in the pseudo code in Section 4.2.2 and the numbers are results of Equations 11 and 12. The results can be seen in Table 2.

Method	STD_x	STD_y	$\tilde{x}_{max}$	$\tilde{y}_{max}$
Linear	.0067	.0068	.0207	.0207
Quadratic	.0050	.0051	.0157	.0157
Pixel Grid [a b]	.0052	.0053	.0161	.0198
Pixel Surface [b]	.0020	.0021	.0067	.0067
Spline [b]	.0020	.0020	.0064	.0064
High Resolution, 11 Multiplier [b]	.0020	.0020	.0064	.0064
High Resolution, 33 Multiplier [b]	.0004	.0004	.0012	.0012

[a] Some data points did not converge and were omitted from the calculations
[b] These Calculations were performed using high resolution data

Table 2: Table of results from various elliptical estimation methods.

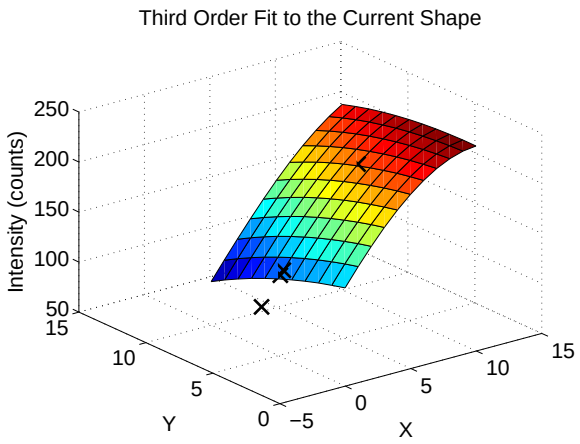

Figure 29: A pixel surface interpolation profile using high resolution data across a single low resolution pixel.

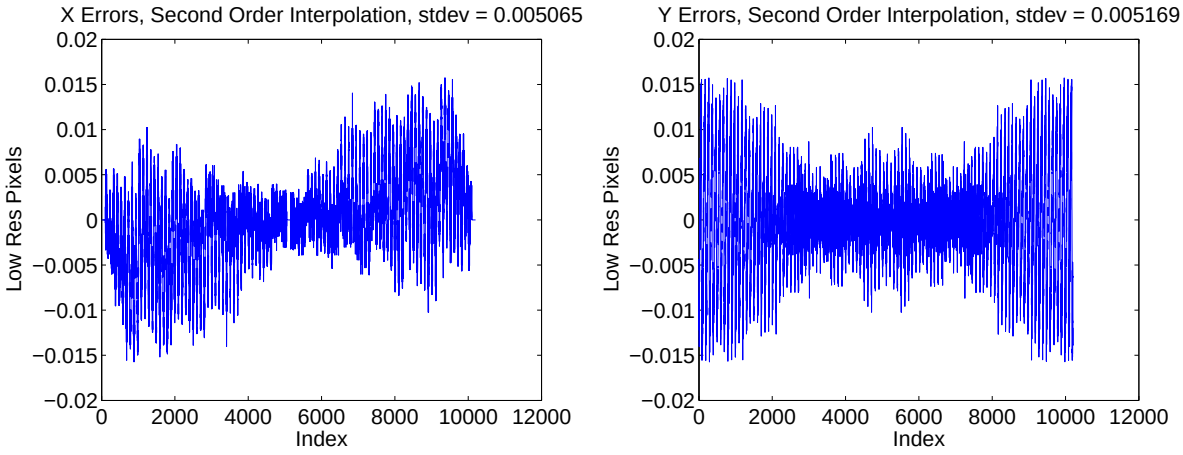

Figure 30: Results from the second order interpolation method.

6.2.1 Quadratic interpolation

To attempt to achieve a more precise estimate of the landmark location the method described in Section 6.1.1 was used. The reasoning was that the transition from interior to exterior pixels was not linear and perhaps a quadratic estimation would better approximate this nonlinear transition.

Magnitude of the center offsets The results of running pseudo code from 4.2.2 is shown in Figure 30. The deviations from using equation 11 is $[Dev_x, Dev_y] = [.0051, .0052]$ mm with extremum $[.016, .016]$ mm. It seems that the benefit of using a quadratic interpolation method instead of a linear method did not improve the estimation significantly enough to justify using this technique.

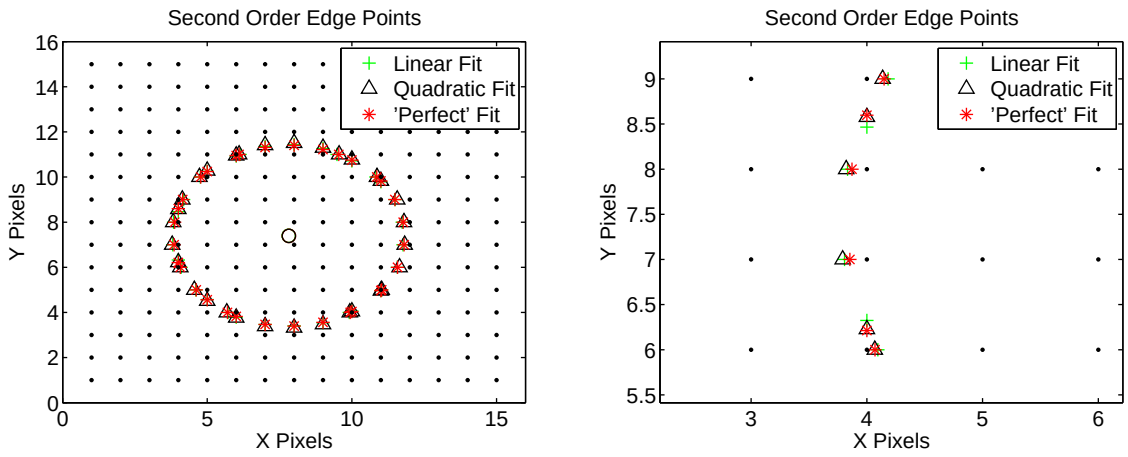

Figure 31: First order estimated contour points vs second order estimated contour points.

Location of the new estimated contour points Figure 31 show the two methods of quadratic interpolation and linear interpolation (the estimated contour point). There is no definite structure as to whether the estimation is nearer to or further from the nearest pixel. The zoomed in portion shows a third plot of the ideal contour points from the high resolution image. In many, but not all, cases the quadratic fit is much closer to the ideal contour points.

6.2.2 Sigmoid method

The results from the sigmoid look-up table were unpromising. Many attempts were made to improve the estimation by tuning the sigmoidicity. This technique couldn't accurately shift the estimated contour points toward the ideal contour points with enough consistency to improve center estimation. As a result, this technique was not considered ideal.

6.2.3 Spline interpolation

The results of the spline interpolation on high resolution image data is shown in Figure 32. These figures don't show any significant improvement from the linear interpolation method of high resolution image data and was not considered beneficial.

6.2.4 Grid interpolation

The grid interpolation method was a technique to attempt to model and understand the curvature of the landmarks. The though was if we could get a sufficient model from the high resolution data and it could be well understood, the curvature of the landmark could then be modelled from the low resolution data. An example of the surface created can be seen in Figure 28. The model did

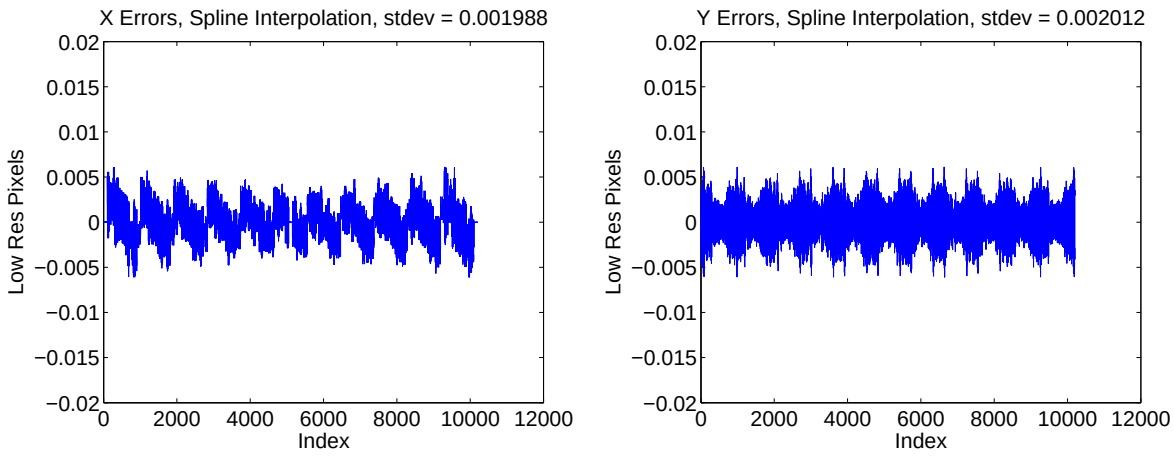

Figure 32: Spline interpolation method deviations of the centers.

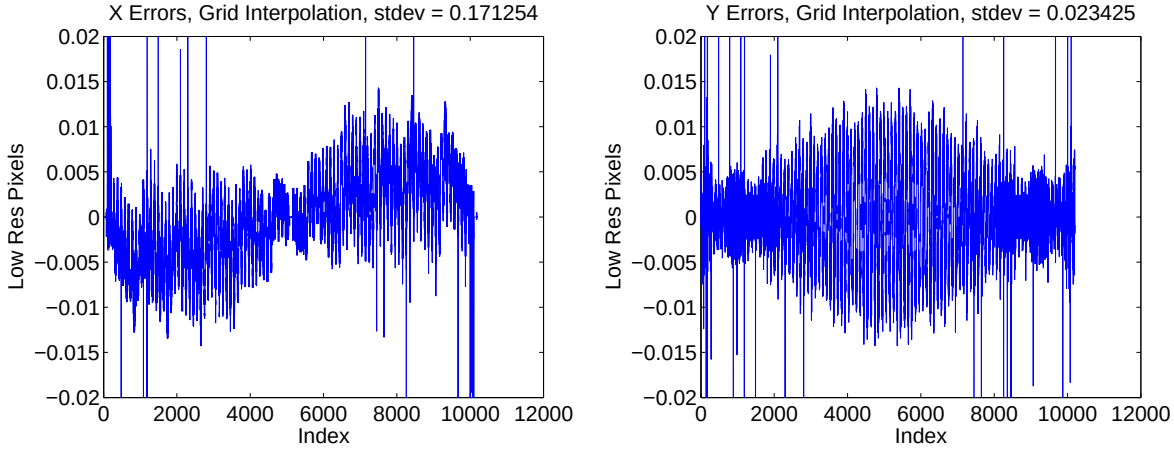

Figure 33: Grid interpolation method deviations of the centers.

not sufficiently predict the data and did not offer any significant estimation benefits, as can be seen in Figure 33. The points that are out of bounds of the plot are where the interpolation failed and the fit estimation was not possible. As mentioned, those points are omitted from standard deviation and maximum error calculations.

A comparison of the cross locations form the grid method can be seen in Figure 36. The estimations are often seen closer to the perfect crosses, but by varying degrees. Also, the complexity of the regressed surface does not guarantee that a crossover point on the interpolation axis exists.

6.2.5 Pixel surface interpolation

After noticing that the curvature of the intensity profile across the pixel could affect the model, an attempt was then made to just investigate the pixel profile. In two simple cases where the intensity across a pixel is constant or linear, the intensity at the center is the same as the average intensity.

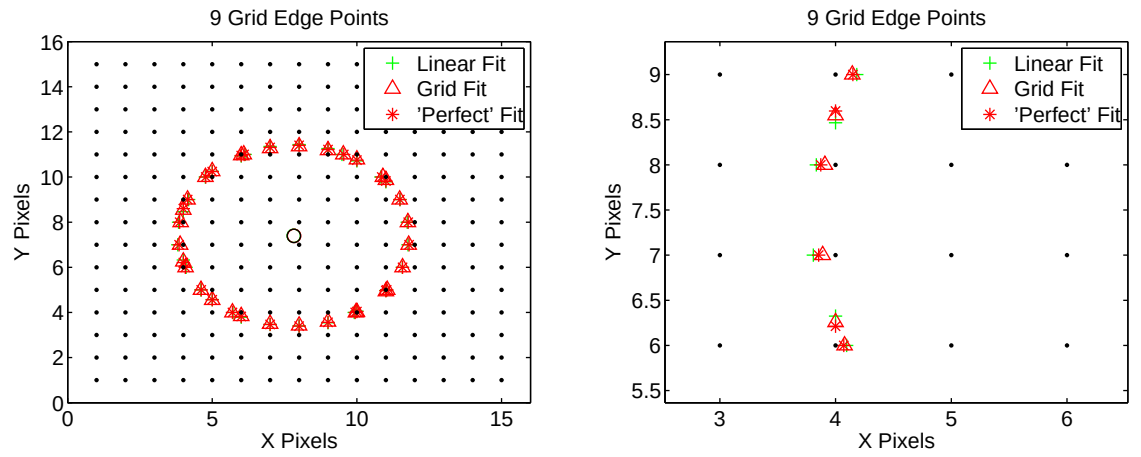

Figure 34: First order interpolation vs grid interpolation estimated contour points.

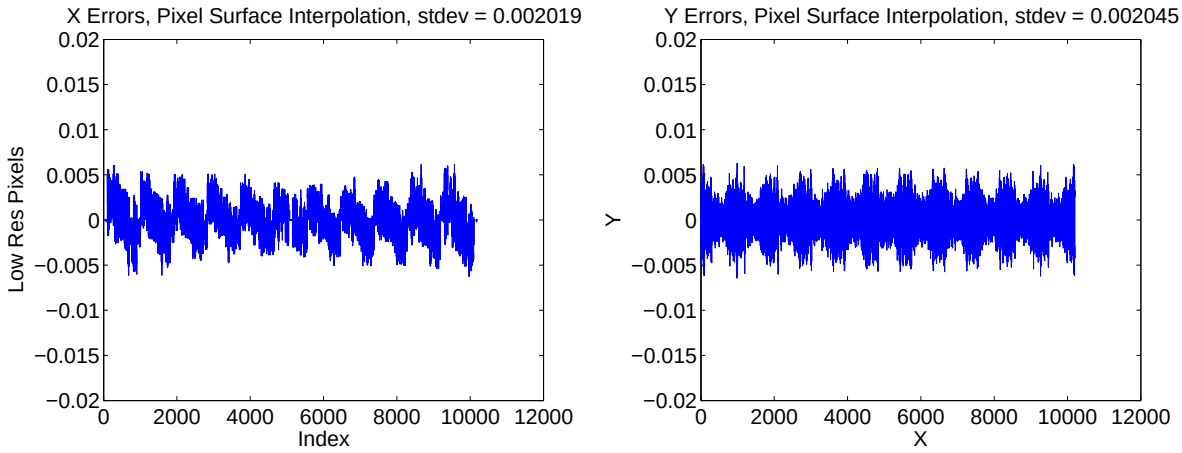

Figure 35: Surface interpolation method deviations of the centers.

This is not the case when more complex profiles are considered. An example of a modelled pixel surface is shown in Figure 29. Modelling the profile across a pixel was hypothesized to be able to give insight as to what exactly could be considered in such cases and possibly assumptions that could be made given certain low resolution data.

The comparison of the first order and the pixel surface estimated contour point locations can be seen in Figure 36. There seems to be a significant improvement toward the 'perfect' points, but the estimation does not offer significant improvement over the linear estimation. The results can be seen in Figure 35.

No clear patterns were observed regarding the contour across a pixel in the high resolution data. As a result, it would be very difficult to accurately model the intensity profile across a pixel in all cases. As a result, this process is not practical for use in actual imaging scenarios.

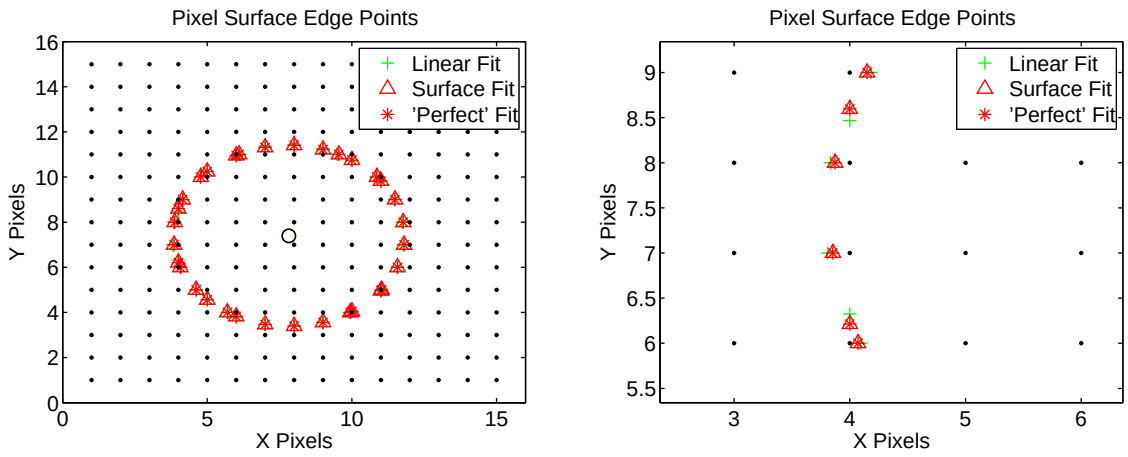

Figure 36: First order interpolation vs pixel surface interpolated estimated contour points.

6.3 Conclusion

Curvature in the illuminance function introduces bias in the estimated contour points, which biases the landmark location. Several techniques were implemented in order to reduce bias in the estimated contour point locations but were unsuccessful. The effect of the bed of nails model is detrimental to every method that has been tested for contour point interpolation of PLL. Chapter 7 attempts to utilize all of the data in order to model a landmark to increase PLL accuracy.

7 Butterworth Tepuy Investigation

Attempts to improve estimation of contour points were shown to be unfruitful. Investigation led to the possibility of fitting a model to the entire data instead of single pixels. The thought was that utilizing all of the pixel data would make the model less sensitive to errors in the bed of nails model.

7.1 Techniques

Two techniques are investigated in this chapter, both are variations on the Butterworth Tepuy model.

7.1.1 Butterworth tepuy fit

Gutierrez and Armstrong [20] propose a fit for a circular or ellipsoidal landmark known as the Butterworth-Tepuy function. The function is supposed to reflect an image having a flat top, sharp transition region, and a flat bottom. The Butterworth Tepuy function has ten parameters which we used as a $\hat{U}$ vector.

$$\hat{U} = [x, y; a, b, c, x_0, y_0, L_{max}, L_{min}, s_b]^T \tag{23}$$

The expected illuminousity function is given as

$$E(x,y) \approx E_{BT}\left(x, y; \hat{U}\right) = \frac{L_{max} - L_{min}}{1 + \left(\frac{(x-x_0)^2 + (y-y_0)^2}{R(x,y)^2}\right)^{\frac{R(x,y)}{s_b}}} + L_{min} \tag{24}$$

The function $R(x,y) \in \Re^p$ is defined as

$$R(x,y) = R(x,y;a,b,c,x_0,y_0) = \sqrt{\frac{\frac{(x-x_0)^2}{(y-y_0)^2} + 1}{a\frac{(x-x_0)^2}{(y-y_0)^2} + 2b\frac{(x-x_0)}{(y-Y_0)} + c}} \tag{25}$$

Let $E(x,y) \in \Re^p$ where $p = m * n$, the size of the low resolution image. As described in [20], Equation 25 reflects the distance of a point relative to the ellipse in figure 37.

Taking $E_{BT}\left(\hat{U}\right) = E(x,y)$, then

$$\Phi(x,y) = \frac{\delta E_{BT}}{\delta U} = \left[\frac{\partial E_{BT}}{\partial a}, \frac{\partial E_{BT}}{\partial b}, \frac{\partial E_{BT}}{\partial c}, \frac{\partial E_{BT}}{\partial x_0}, \frac{\partial E_{BT}}{\partial y_0}, \frac{\partial E_{BT}}{\partial L_{max}}, \frac{\partial E_{BT}}{\partial Lmin}, \frac{\partial E_{BT}}{\partial s_b}\right]^T \tag{26}$$

$$\epsilon = E_{BT}\left(\hat{U}\right) - E(x,y) \tag{27}$$

where $\Phi(x,y) \in \Re^{px8}$and $\epsilon \in \Re^p$.

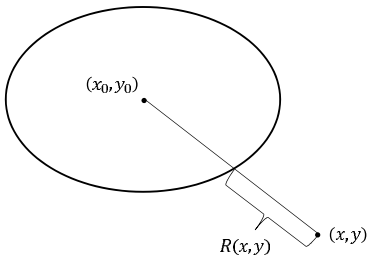

Figure 37: Graphical representation of the R(x,y) function.

By setting

$$\epsilon = \delta E_{BT} = \frac{\delta E_{BT}}{\delta U} \delta \hat{U} = \Phi \delta \hat{U} \tag{28}$$

a Newton-Raphson Algorithm can be used to solve for $\delta \hat{U} \in \Re^8$ that minimizes ϵ. This uses the left-pseudo inverse of Φ and solving

$$\delta \hat{U} = -\left(\Phi' \Phi\right)^{-1} \Phi' \epsilon \tag{29}$$

Setting

$$\hat{U}_{i+1} = \hat{U}_i + \delta \hat{U}_i \tag{30}$$

and iterating until an exit condition is met, namely the relative difference

$$\frac{\delta \hat{U}_i}{\hat{U}_i} < \alpha \tag{31}$$

a final model $E_{BT}\left(x, y; \hat{U}\right)$ could be reached.

An example of a Butterworth Tepuy landmark can be seen in Figure 39 with the actual landmark shown in Figure 38.

7.1.2 Butterworth Tepuy fit with 2D Simpsons integration

One last method used proposed by Gutierrez and Armstrong[20] was a 2D-Simpsons Integration method in conjunction with the Butterworth-Tepuy fit. The 2D Integration was meant to give a

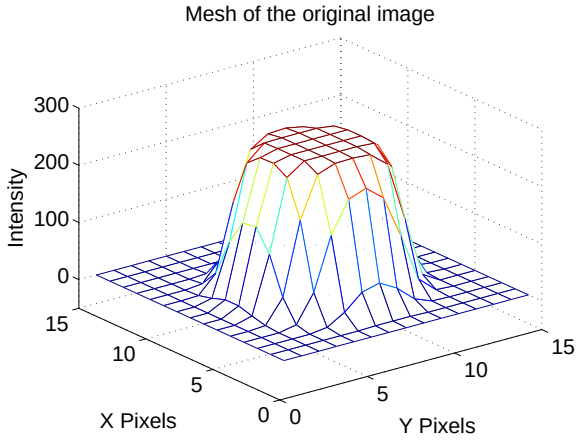

Figure 38: 3D plot of an original landmark.

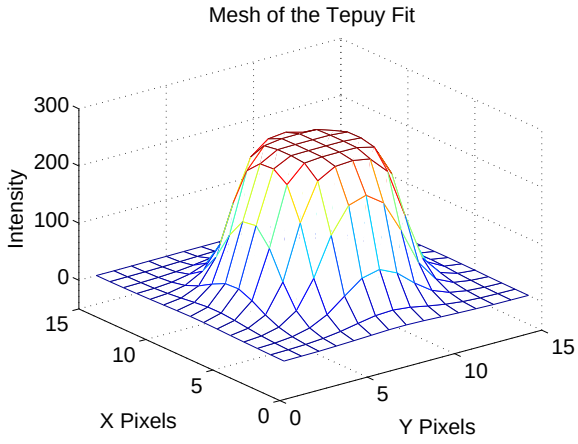

Figure 39: 3D plot of a tepuy fit estimation to a landmark.

more accurate estimation of pixel values based on a 100% sensitive area pixel model.

A 3x3 grid of points within a pixel were used where if $[x_c, y_c]$ represents the center of the pixel and the total pixel area is $[x_c \pm 0.5, y_c \pm 0, 5]$ then the grid would include $\Delta x = \left[\frac{-1}{2}, 0, \frac{1}{2}\right]$, $x = x_c + \Delta x$ and, similarly, $\Delta y = \left[\frac{-1}{2}, 0, \frac{1}{2}\right]$, $y = y_c + \Delta y$.

Then, for 2D Simpsons Integration to take place, a weight matrix was created,

$$W = \begin{bmatrix} 1 & 4 & 1 \\ 4 & 16 & 4 \\ 1 & 4 & 1 \end{bmatrix}$$

This matrix was condensed to a column vector

$$W = [1, 4, 1, 4, 16, 4, 1, 4, 1]^T$$

and the vector of modeled values

$$E = \left[E_{BT}\left(x + \Delta x, y + \Delta y; \hat{U}\right)\right] \tag{32}$$

was created.

The Simpsons Integral in a 3x3 grid is then defined as

$$\iint E_{BT}\left(x, y; \hat{U}\right) dxdy \approx \frac{1}{9} h_x h_y W * E \tag{33}$$

where h_x and h_y are defined to be the magnitude of the steps in Δx and Δy, respectively.

These new points were fit using the Newton-Raphson algorithm described in Section 7.1.1.

7.2 Results

7.2.1 Butterworth Tepuy fit

The Butterworth-Tepuy fit was performed on the low resolution data and *produced the most accurate results* of any of the low resolution methods. An image of a landmark and the Butterworth Tepuy fit can be seen in Figures 38 and 39, respectively. The deviations from the center estimates can be seen in Figure 40. The magnitude of the deviations appears significantly smaller than the deviations seen in the original linear case in Figure 18.

There are many differences between the Butterworth Tepuy method and the contour point esti-

Method	STD_x	STD_y	$\tilde{x}_{max}$	$\tilde{y}_{max}$
Linear	.0067	.0068	.0207	.0207
Quadratic	.0050	.0051	.0157	.0157
Butterworth	.0023	.0024	.0085	.0085
Butterworth 2D	.0024	.0025	.0087	.0087
Pixel Grid [a] [b]	.0052	.0053	.0161	.0198
Pixel Surface [b]	.0020	.0021	.0067	.0067
Spline [b]	.0020	.0020	.0064	.0064
High Resolution, 11 Multiplier [b]	.0020	.0020	.0064	.0064
High Resolution, 33 Multiplier [b]	.0004	.0004	.0012	.0012

[a] Some data points did not converge and were omitted from the calculations
[b] These Calculations were performed using high resolution data

Table 3: Table of results from all corrective methods.

mation methods.

- Taking the entire model into account

The Butterworth-Tepuy fitting process takes into account the entire model, not just two or more pixel values. This has certain advantages and disadvantages, described below.

- Curvature is accounted for in the model

The Butterworth Tepuy model has curvature, whereas Linear Models do not. Therefore, given accurate fits, the errors across a pixel could be greatly reduced.

- All of the data is considered

By increasing the number of data points, there is a reduction in the sensitivity to noise. Linear Interpolation cases simply ignore almost all of the data above and below a specific threshold. The Butterworth-Tepuy algorithm utilizes all of this data to attempt to predict a model. Theoretically more data yields a better fit.

- There is no dependence on a threshold

A threshold variable is not considered in a Butterworth Tepuy model. This reduces one less user-created variable that needs to be implemented in the model.

- The model doesn't consider uneven lighting profiles

Not all side effects are good. In this case, if the lighting profile were uneven; say, the flat top or bottoms weren't flat, the model would currently be unable to predict this. In the future this could probably be implemented into the model if it were seen to be beneficial.

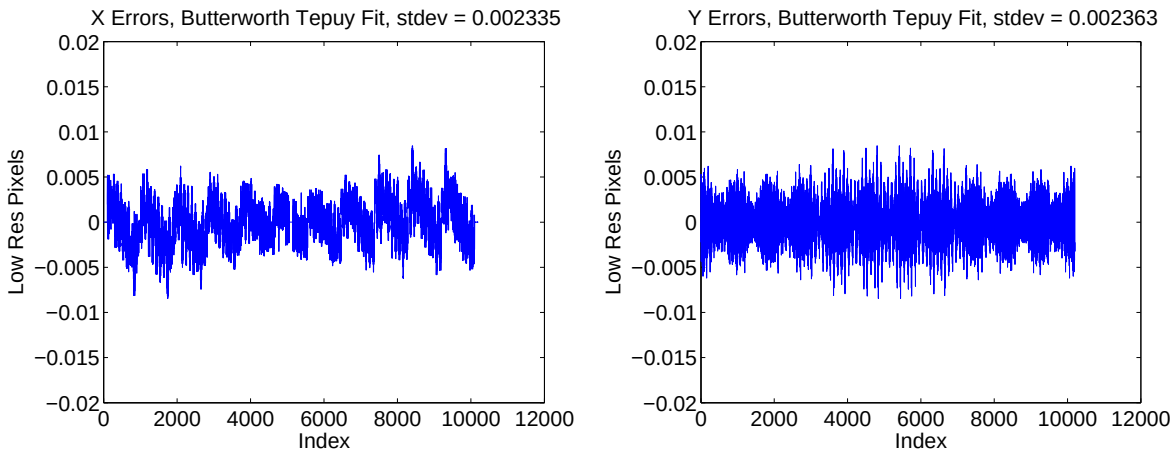

Figure 40: Butterworth Tepuy estimated deviations from center.

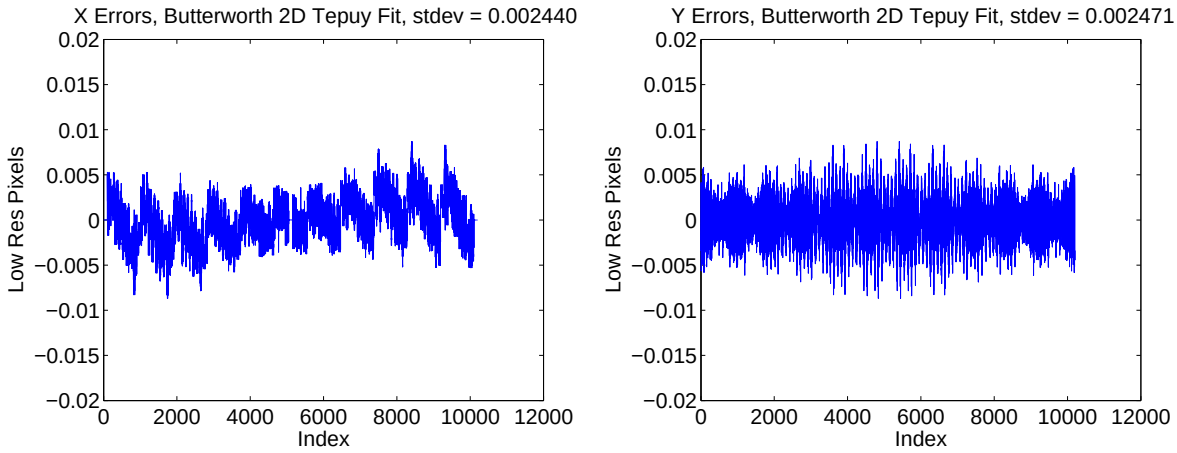

Figure 41: 2D Butterworth Tepuy estimated deviations from center.

7.2.2 Butterworth Tepuy fit with 2D Simpsons integration

Despite promising hypotheses, this integration did not greatly improve the estimation. Investigation could be performed as to whether a larger integration grid (a 3x3 grid was used) could improve accuracy. This method was not pursued further, as the Butterworth-Tepuy model seemed sufficiently accurate to draw conclusions. The results from the investigation can be seen in Figure 41 and the differences between the integrated method and the original Butterworth Tepuy method can be seen in Figure 42.

7.3 Conclusion

The Butterworth Tepuy algorithm offered the best corrective method analytically from the low resolution image data. The technique is robust enough for practical implementation and accurate

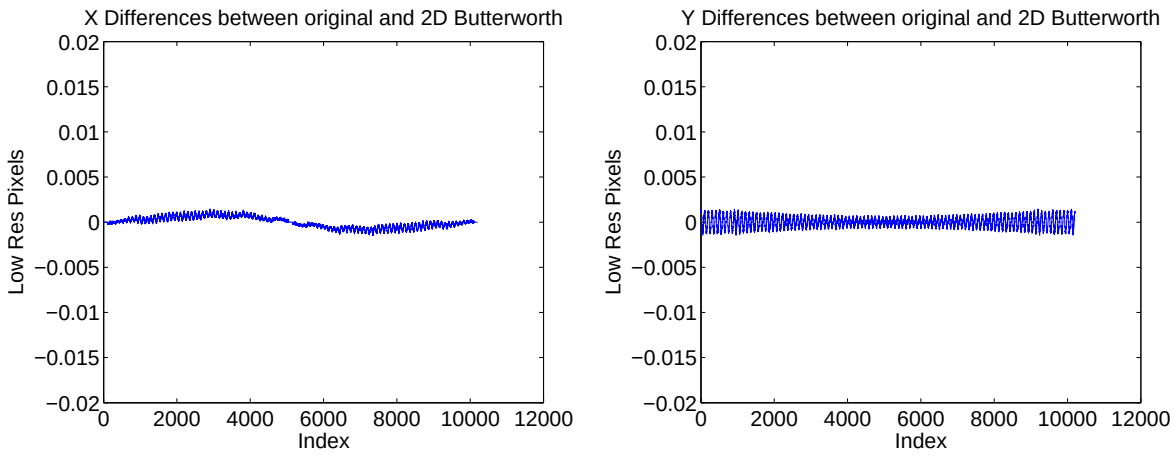

Figure 42: Differences between the two Butterworth methods.

enough to justify use. The magnitude of correction from the Butterworth Tepuy algorithm is similar to having the original high resolution pixel info, or 121 times more image data. The results from the Butterworth Tepuy method are very promising and could yield much better estimations with little jump attributed to bias.

Part III

Results and Conclusion

8 Butterworth Tepuy method in a jump

A method was created to make the Tepuy algorithm robust enough to deal with actual images. This is necessary because pixels that are outside the tepuy might be affected by the nearby white regions of the target. The algorithm performs as follows.

1. Locate possible pixels near a tepuy
2. Find which of those pixels are outside the tepuy
3. If the pixels are outside the tepuy, ensure that they are dark enough
4. Fit a tepuy, placing weights on the transition regions
5. Search to find if any pixels outside the tepuy are unusually higher than the others and omit them
6. Repeat steps 4 and 5 for a more precise fit
7. Fit a final tepuy

The resulting technique was used to analyze images in Section 8.1.

8.1 Observing depth vs roll angles

The images from the jump referenced in Chapter 2 were re-analyzed with the Butterworth Tepuy algorithm in place of the contour point estimation algorithm.

The results of the initial observation can be seen in Figure 43. This figure shows angles where there are patterns of bias. Between 84 and 84.5 degrees of rotation, there seems to be a spike in the positive Z direction, away form the regression line of the data. There seem to be a number of these phenomena over the range of the experiment.

Figure 44 shows the result of the data with the tepuy algorithm applied. The variance of the tepuy corrected data was less than half of the variance of the standard data. Additionally, the apparent spikes in the data appear to be all but removed completely. It appears that the bias has been removed and Gaussian noise dominates the errors. The two data sets over a smaller angle

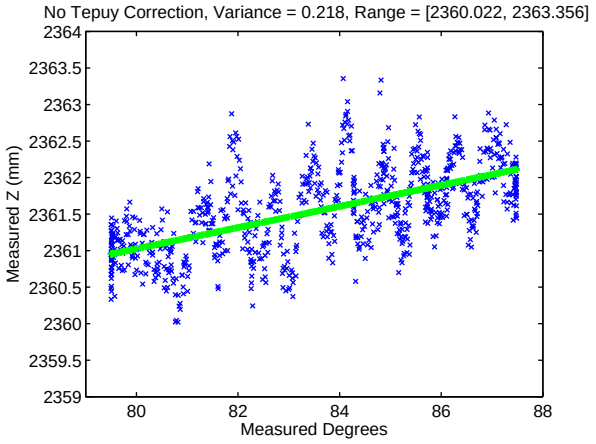

Figure 43: Linear interpolated contour point technique Z measurements vs calculated roll.

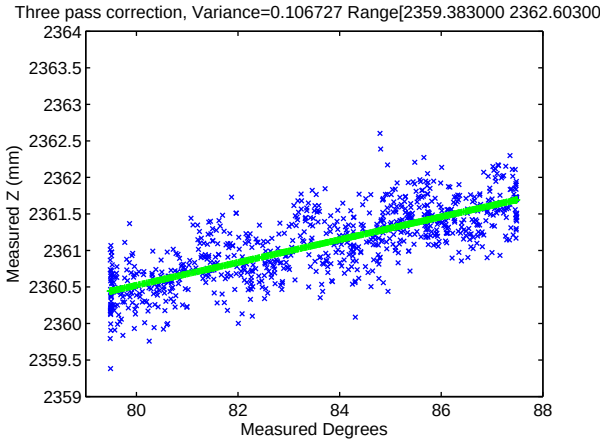

Figure 44: Butterworth Tepuy estimation technique Z measurements vs calculated roll

can be seen in Figure 45. The histograms of Figure 46 appear to support the idea that the tepuy corrected method has much more random errors than the ellipse algorithm.

The differences in Z (Figure 47) show that the Tepuy estimation measures closer to the camera than the ellipse algorithm. The magnitude of the differences are on the order of 1mm. This distance appears to be affected by the sharpness of the weighting algorithm used, which is described in Section 8.4.

8.2 Observing locations of each ellipse

Observations can be made on the position of each ellipse individually. In the image set, one ellipse is especially interesting because it consists of two significantly different magnitudes of motion. The movement of the estimated location of X with respect to roll shows a range of only .25 pixels, whereas

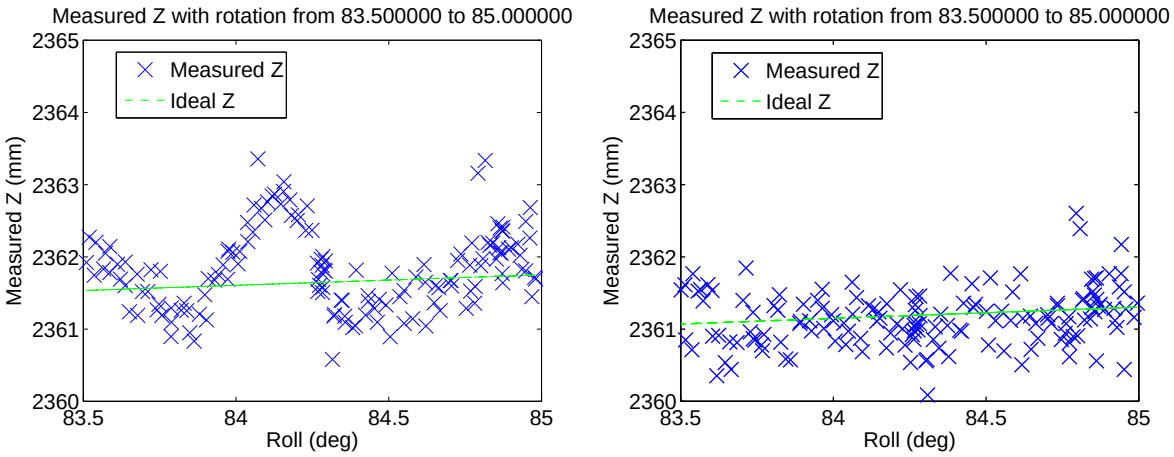

Figure 45: Linear estimation of contour points and Butterworth Tepuy corrected estimation of a jump.

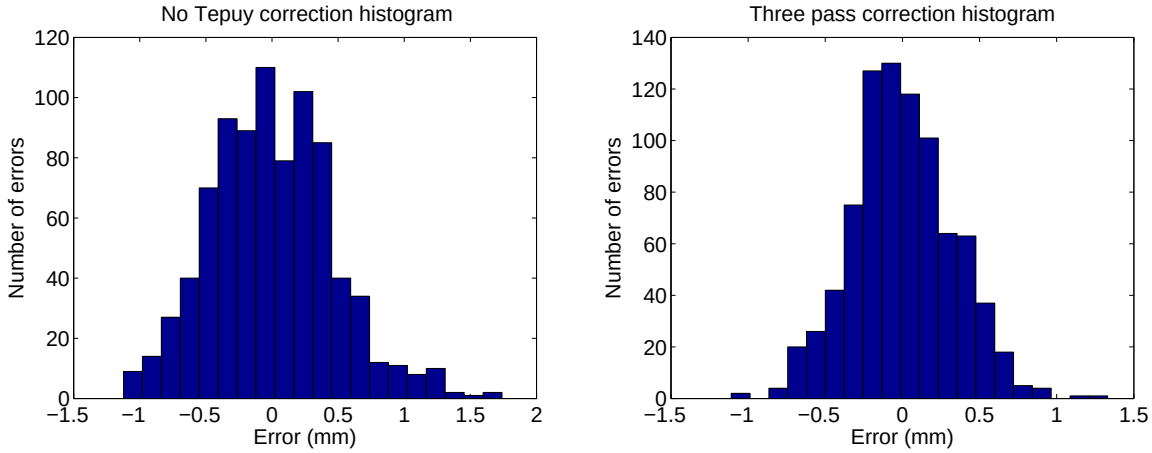

Figure 46: Histograms of the errors with linear contour point estimation (left) and Butterworth Tepuy estimation (right).

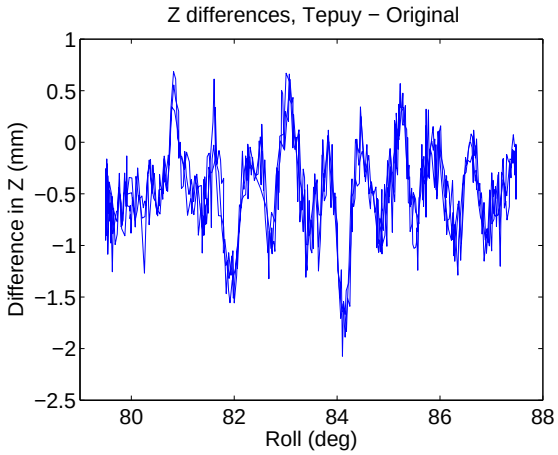

Figure 47: Differences in measured Z between ellipse fitting and Butterworth Tepuy techniques.

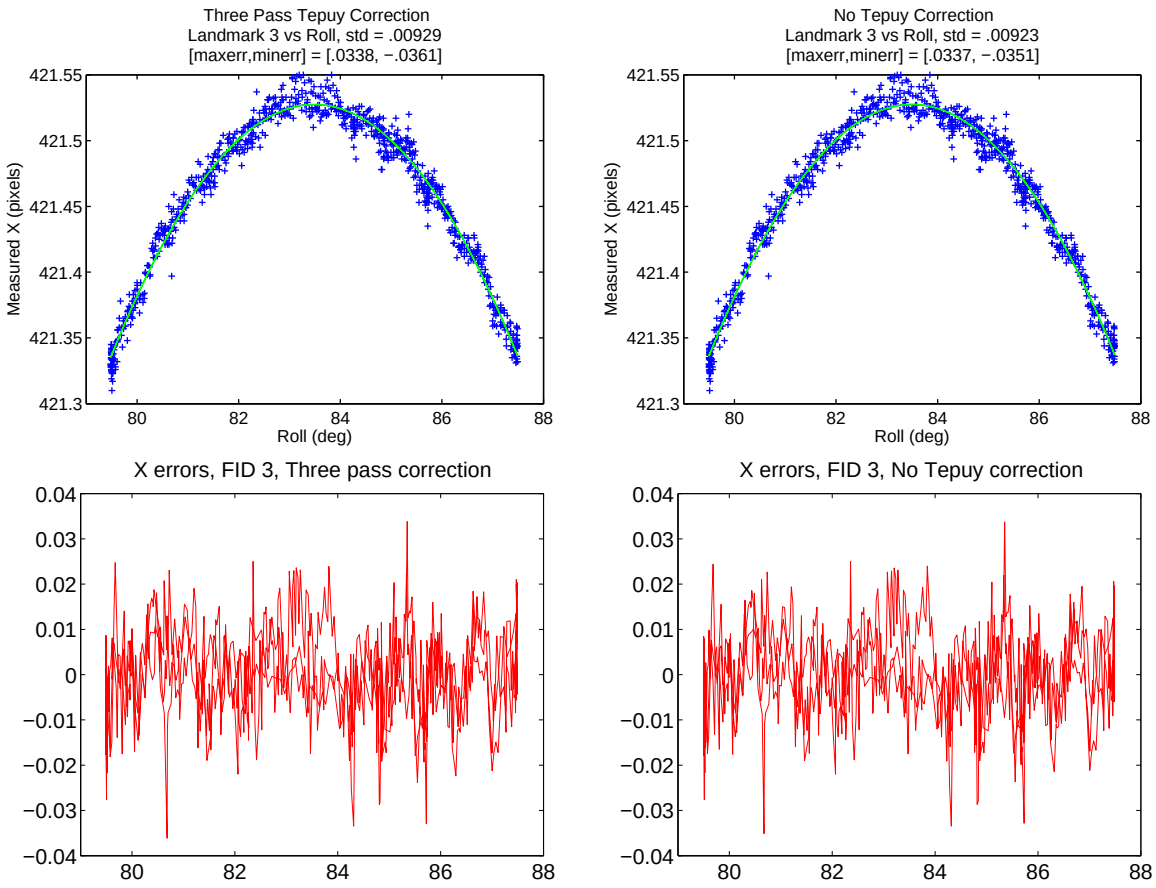

Figure 48: Estimated X movement of the circular landmark using the Butterworth Tepuy method (left) and the ellipse method (right)

the Y movement shows movement over 12 pixels. This motion can be seen in Figures 48 and 49. The estimation and deviations from a second order regression can also be seen in the figures.

In these figures it is not clear that the Tepuy fit offers a better estimation. Errors and trends can be seen in both estimation procedures, and the magnitude of the errors are very similar. Although a second-order regression of Position vs Roll was believed to be sufficient, it is possible that higher order paths might be necessary in order to estimate the path of a landmark. Additionally, roll is the X-axis of each of the plots, which is also a measured quantity. It is possible that errors observed could actually be amplified by the miscalculation of roll angle.

8.3 Effects of the multiple-pass tepuy algorithm

As described at the beginning of this chapter, the tepuy fitting algorithm takes multiple passes in order to improve estimation. The effects of the measurements from each pass can be seen in Figure 50 with the errors shown in Figure 51.

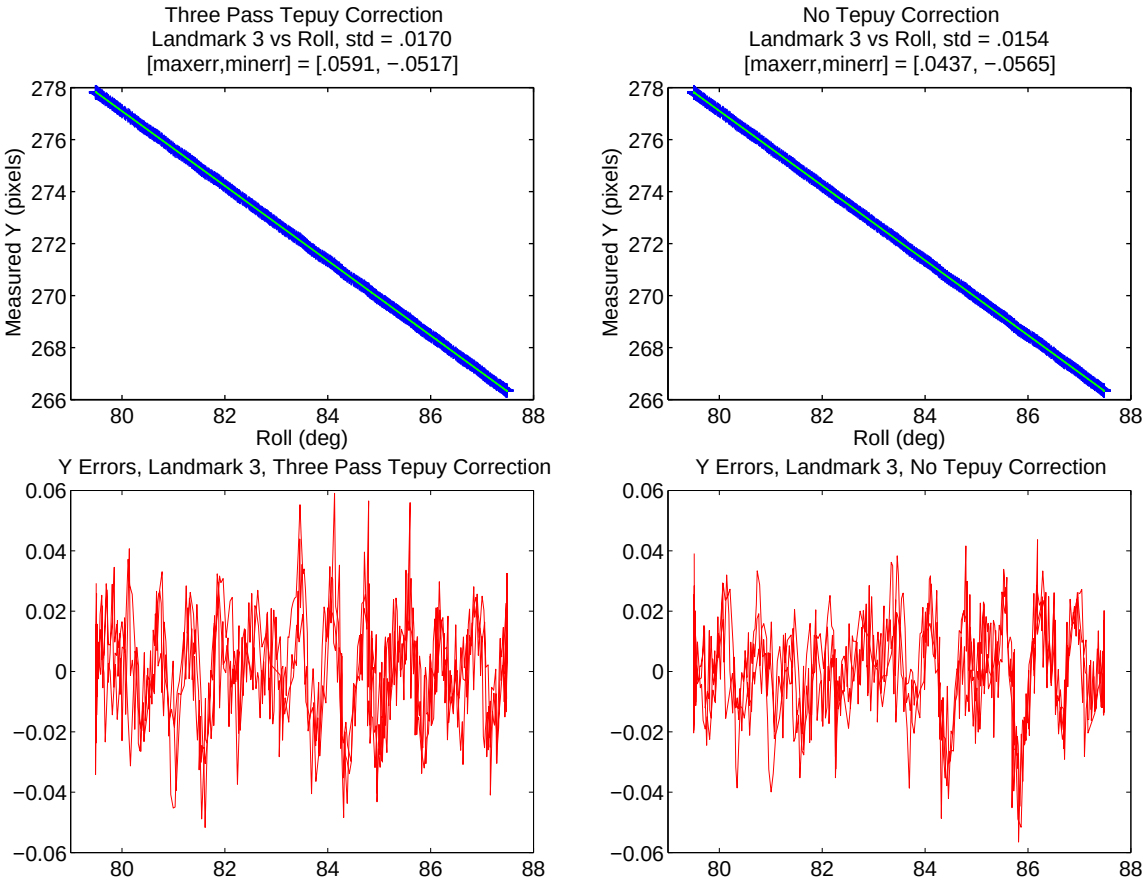

Figure 49: Estimated Y movement of the circular landmark using the Butterworth Tepuy method (left) and the ellipse method (right)

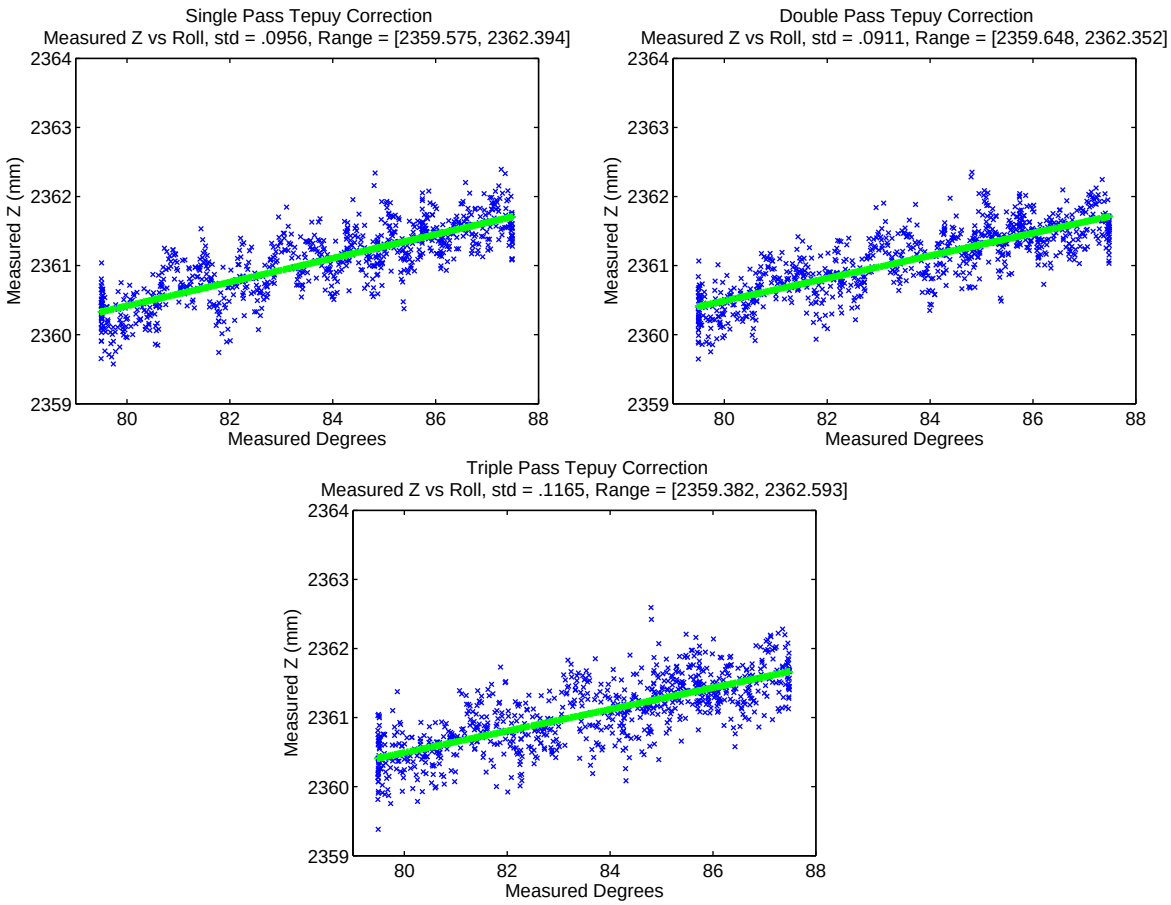

Figure 50: Effects of multiple passes on the tepuy algorithm. Single pass (upper left), double pass (upper right) and triple pass (lower) Butterworth Tepuy algorithms.

It is noted that the variance of the measurements actually increases with each pass. The errors, however, appear to be slightly more Gaussian after the third pass than the first, so it is possible that the biases were further reduced after the third pass.

A strategy to combat this phenomena would be to modify the target design to allow for more guaranteed outside pixels. In this case, more data would be kept for analysis and could allow for a more accurate fit.

8.4 Effects of weights applied to the Butterworth Tepuy algorithm

The data was weighted to put an emphasis on the transition data instead of the flat regions. The idea of weighting the transition regions of contour was shown in Chiorboli and Vecci[32]. The function used to perform this was

$$Weight\,(\mathbf{p}) = e^{-k*\left(\frac{I_{BT}(\mathbf{p})-\bar{I}}{I_{max}-I_{min}}\right)^2}$$

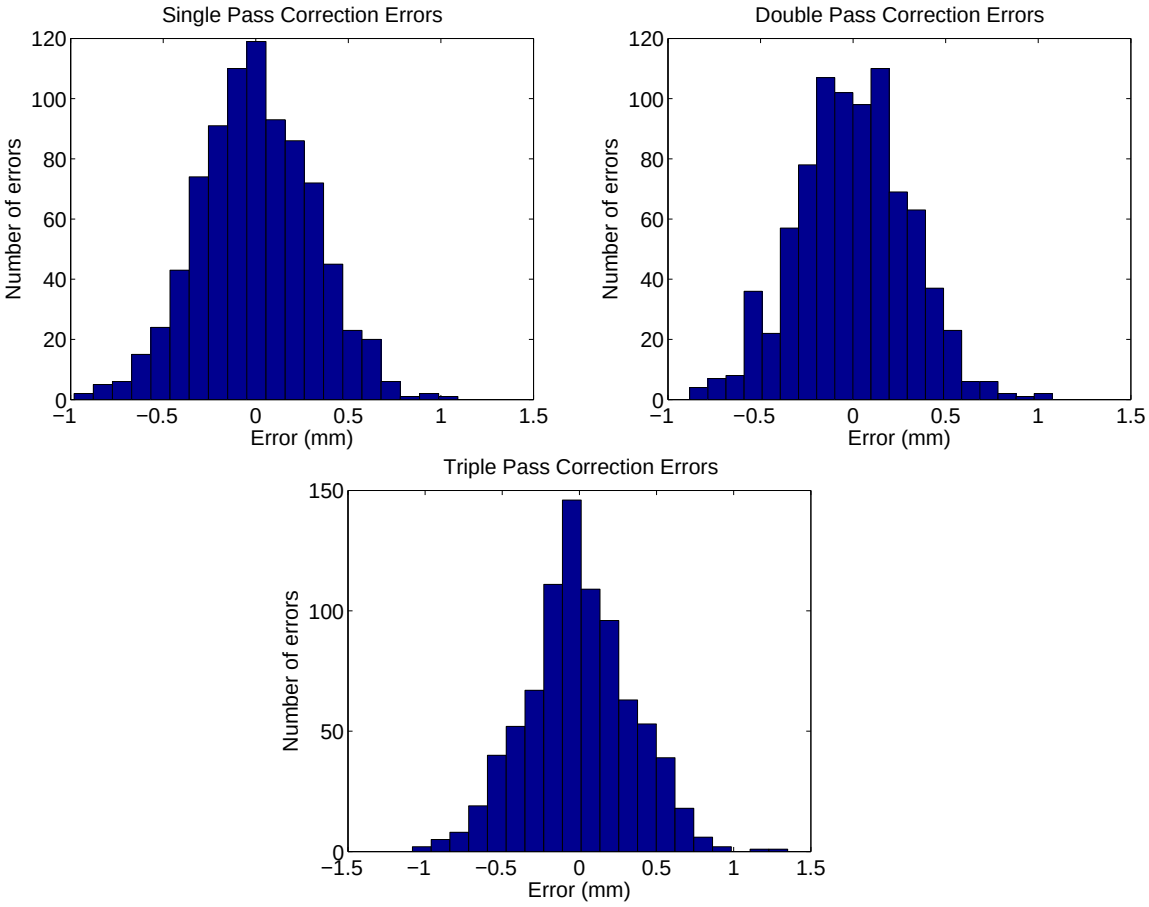

Figure 51: Histogram of the errors from each pass of the tepuy algorithm. Single pass (upper left), double pass (upper right) and triple pass (lower) Butterworth Tepuy algorithms

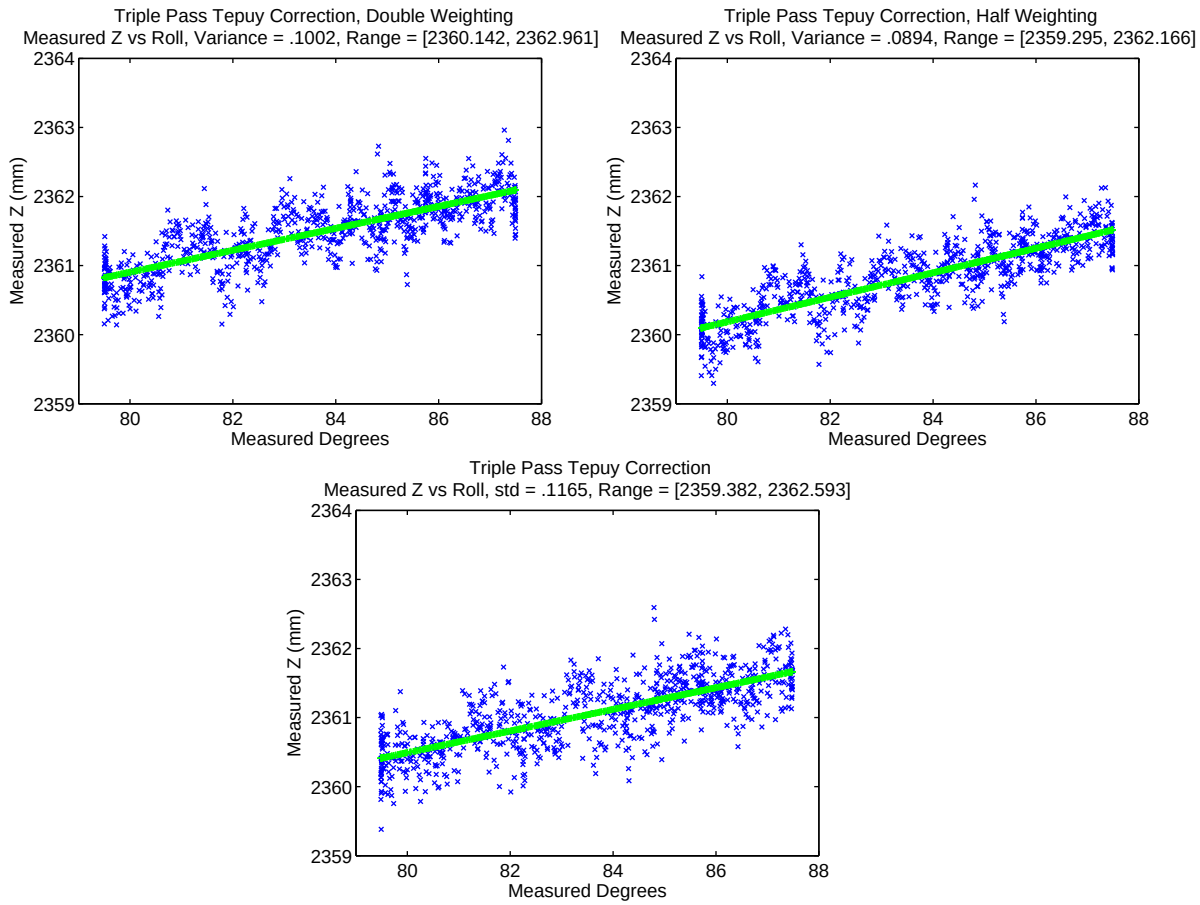

Figure 52: Effects of weights on the Tepuy algorithm. Double weights (upper left), half weights (upper right) and standard (lower) are shown.

where k is a constant, $I_{BT}(\mathbf{p})$ is the estimated intensity at the pixel, $\overline{I}$ is the average intensity over the pixels being analyzed, and $I_{min,max}$ are the minimum and maximum intensities, respectively.

Originally $k = 2$ was used. The graphs of using different values of k are shown in Figure 52 with the errors in Figure 53. Biases seem much less of a factor when the original, $k = 2$ weighting was used than the other two weighted cases. When weighting is turned off, the variance increases.

8.5 Introducing a gradient to the Butterworth Tepuy algorithm

The calculation for the Butterworth Tepuy was modified to support a 2D gradient, as such.

$$E(x,y) \approx E_{BT}\left(x,y;\hat{U}\right) = \frac{L_{max}\left(1+g0x+g1y\right)-L_{min}}{1+\left(\frac{(x-x_0)^2+(y-y_0)^2}{R(x,y)^2}\right)^{\frac{R(x,y)}{s_b}}} + L_{min} \tag{34}$$

This gradient was added to simulate non-uniform lighting over the observed area. An example of a single fit can be seen in Figure 54 where the circles represent the estimation and the crosses

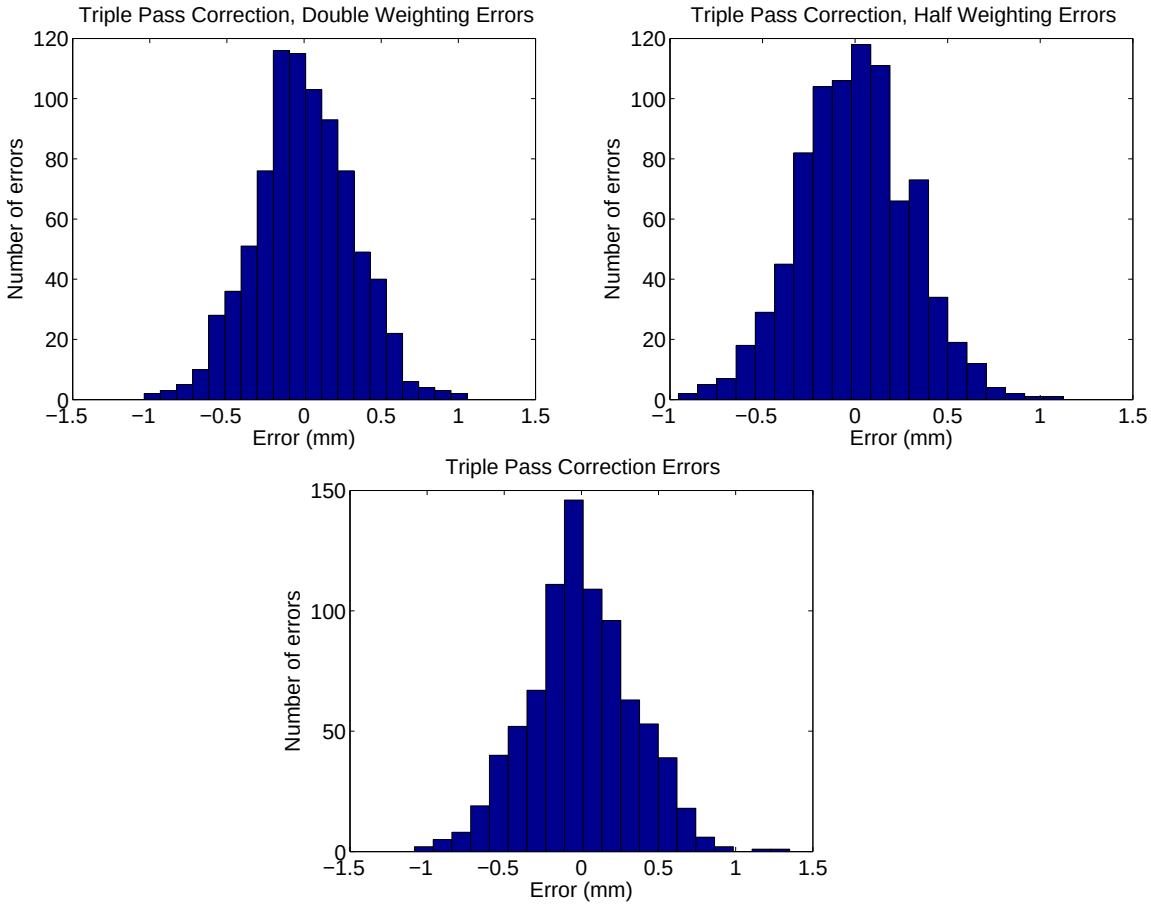

Figure 53: Errors from different weights of the Tepuy algorithm. Double weights (upper left), half weights (upper right) and standard (lower) are shown.

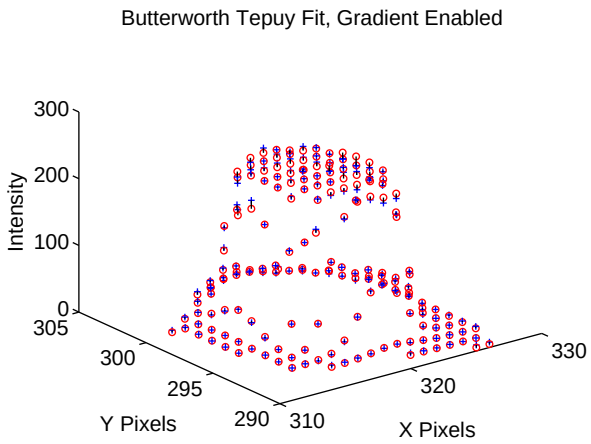

Figure 54: Estimation of a Tepuy with a gradient added.

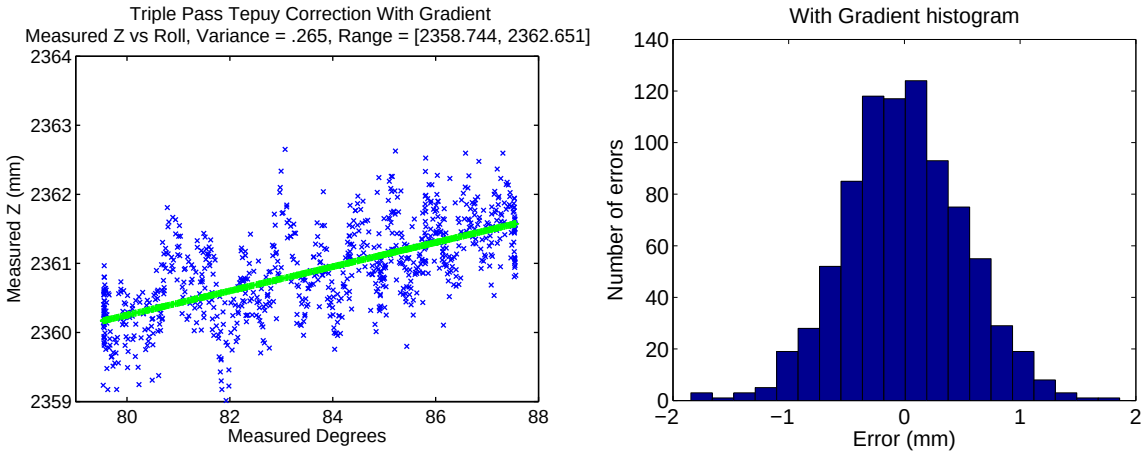

Figure 55: Errors with a gradient introduced in Butterworth Tepuy algorithm.

represent the observed data. The full results of the 840 images are shown in Figure 55. The variance increased using this process and the bias trends seem to appear in the data. Although this was considered to be a helpful addition to the Butterworth Tepuy algorithm, it seems that this technique does not improve accuracy.

The maximum gradients observed were on the order of $.035\frac{counts}{pixel}$ with most observations being an order of magnitude smaller than this.

9 Conclusions/Recap

9.1 Bias as a source of error

It has been shown in Section 4.2.2 that bias exists in precision landmark location. This bias is much larger than deviations attributed to noise, as shown in Section 4.2.4. It has also been shown in Section 4.2.3 that this bias could be the main cause of the observed jump phenomena. Butterworth Tepuy estimation minimizes this bias and theoretically should minimize the jump phenomena. Real-world experiments show this to be the case, and the effects of jumps appear to be reduced significantly.

9.2 Curvature of intensity profile causes bed of nails to be inaccurate

Curved intensity profiles yield cases where the average intensity and intensity at the center are two different values. These differences cause a systematic inaccuracy in precision landmark location. Proper estimation procedures and estimation models as introduced in Section 7.1.2 should be able to combat the bed of nails inadequacies and provide a better estimation.

9.3 Curvature causes linear interpolation to be inaccurate

Contour point estimation is not accurate enough to perform precision landmark location. They are not near enough to the true transition locations. This model must be improved upon or, in the case of Butterworth Tepuy estimation, abandoned completely. Several methods to improve this technique were quantitatively investigated in this Thesis.

9.4 Butterworth Tepuy offers more accurate landmark estimation than the ellipse algorithm

Errors attributed to bias were reduced by a factor of 3 by using the Butterworth Tepuy algorithm. The degree of improvement is similar in magnitude to having 11 times the amount of data during a linear interpolation. Several correction methods were proposed and tested, and the Butterworth Tepuy method had the best results.

The deviations from real world data were shown to be significantly reduced. This is a strong verification of the prediction that the Tepuy algorithm offers much more accurate results than the ellipse algorithm.

9.5 Impact of this Thesis

This research offers insight to what causes the errors in MPT. Before the research was performed, hypotheses were constructed as to why the jump phenomenon was taking place, many of which suspected imperfections in the imager. The research has shown that the jump phenomenon would still happen in an ideal imager. Additionally, the research uncovers two facts regarding contour estimation: the intensity of a pixel is not equal to the intensity at the center of a pixel, and the contour between two pixels is non-linear. This is due to curvature in the intensity profile across the imager. In studied cases referenced by Tian and Huhns[23] errors were acceptable on the order of ± 1 to 0.1 pixels. The accuracy of MPT is sensitive to errors of $\pm .002$ pixels. The precision of this research is believed to be unparalleled.

This research continues by implementing a corrective algorithm for this phenomenon. Simulation showed that improvements should be observed and experimental validation supports this theory. MPT and any other technology that requires the precise location estimation of circular landmarks can benefit from the findings of this work.

9.6 Future investigations

There are a few different investigative paths that could be further pursued in order to improve precision landmark location.

9.6.1 Further investigation into modeling the curvature of landmarks

An improvement could potentially be achieved by investigating landmark structure relative to the bed of nails model. There could potentially be a contour point estimation procedure that would improve the accuracy of the contour points. This could improve the ellipse estimation technique.

9.6.2 Gather more contour points before the interpolation process

Potentially utilizing more or different interpolation axes could prove to be beneficial. Instead of interpolating along pixel boundaries, one could create contours with several neighboring pixels and interpolate along one or several axes. There is potential for improvement by investigating specific direction vectors, namely ones tangential to the contour of the ellipse being investigated.

9.6.3 Using more refined 2D Simpsons Integration techniques in order to get a better Butterworth Tepuy estimation

It is very counter-intuitive that the 2D Simpsons Integration technique did not improve landmark estimation. Perhaps using more precise integration techniques with more samples could yield promising results.

9.6.4 Robust statistical methods to improve Butterworth Tepuy estimation

Investigations have been done to attempt to create better Butterworth Tepuy estimations using roust statistics, but have not yielded promising results. A deeper understanding and further attempts to refine the statistical methods could prove beneficial and lead to a better center estimation.

9.6.5 Modifications to the Butterworth Tepuy model

The Butterworth Tepuy model uses the parameter $R(x, y)$. This parameter represents the distance from a point to the defined ellipse on a line through the ellipse center. A more appropriate model of $R(x, y)$ would represent the distance to a line normal to the ellipse. During most of the investigation a circular landmark was used in which case the two lines are the same.

Further studies regarding the introduction of higher-order gradients could be tested in order to account for non-uniform lighting conditions. It is known that non-uniform lighting is the case in these images, and perhaps knowing that model more precisely could improve estimation.

9.6.6 Modifications to the target

Multiple passes were necessary in order to discover which pixels to use in fitting the Butterworth Tepuy model. This step might not be necessary at all if the targets were designed to have a more consistent dark region outside of the landmark. This effect was not necessary to take into consideration in the current model but for future revisions it might be beneficial.

References

[1] S. Mihradi, Ferryanto, T. Dirgantara, and A. Mahyuddin, "Development of an optical motion-capture system for 3D gait analysis," in *2011 International Conference on Intsrumentation, Communication, Information Technology and Biomedical Engineering.* Bandung, 2011, pp. 301 – 303.

[2] V. W. Zhou, A. Z. Kyme, S. R. Meikle, and R. R. Fulton, "Event-by-event motion compensation for small animal PET," in *2007 IEEE Nuclear Science Symposium Conference Record.* Honolulu, 2007, pp. 3109 – 3114.

[3] Z. Kertesz and I. Lovanyl, "3D motion capture methods for pathological and non-pathological human motion analysis," in *Information and Communication Technologies, 2006. ICTTA '06.* Damascus, 2006, pp. 1062 – 1067.

[4] B. Armstrong and B. S. Holeman, "Target tracking with a network of Doppler radars," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 34, no. 1, pp. 33–48, 1998.

[5] B. Armstrong, T. Verron, L. Heppe, J. Reynolds, and K. Schmidt, "RGR-3D: Simple, cheap detection of 6-DOF pose for tele-operation, and robot programming and calibration," in *Proc. 2002 Int. Conf. on Robotics and Automation.* IEEE: Washington, 2002, pp. 2938–2943.

[6] K. M. O'Connor, B. S. R. Armstrong, J. Weinhandl, T. P. Kusik, and R. T. Barrows, "Validation of a single camera 3D motion tracking system," in *Proc. Annual Meeting of the American Society of Biomechanics.* ABS: State College, Penn., Aug 2009, p. 980.

[7] A. G. Webb, *Introduction to Biomedical Imaging.* Wiley, 2003.

[8] J.-H. Kim, J. Nuyts, Z. Kuncic, and R. Fulton, "A method of motion tracking during CT for motion correction," in *2011 IEEE Nuclear Science Symposium Conference Record.* Valencia, 2011, pp. 3816 – 3819.

[9] J. Maclaren, B. S. R. Armstrong, R. T. Barrows, K. A. Danishad, T. Ernst, C. L. Foster, K. Gumus, M. Herbst, I. Y. Kadashevich, T. P. Kusik, Q. Li, C. Lovell-Smith, T. Prieto, P. Schulze, O. Speck, D. Stucht, and M. Zaitsev, "Measurement and correction of microscopic head motion during magnetic resonance imaging of the brain," *PLoS ONE*, vol. 7, no. 11, p. e48088, 11 2012.

[10] B. Armstrong, M. Botum, M. Farrah, K. O'Connor, and S. Watts, "An innovative diagnostic tool for reducing traumatic knee injuries," in *Proc. Annual Meeting of the American Society of Biomechanics.* ABS: Stanford, CA, aug 2007, poster Session P9-9.

[11] J. Weinhandl, B. S. R. Armstrong, J. E. Earl, T. P. Kusik, R. T. Barrows, and K. M. O'Connor, "Assessment of a potential ACL injury risk protocol," in *Proc. Annual Meeting of the American Society of Biomechanics.* ABS: State College, Penn., Aug 2009, p. 1141.

[12] V. S. Nalwa and T. O. Binford, "On detecting edges," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 699–714, 1986.

[13] A. Goshtasby and H.-L. Shyu, "Edge detection by curve fitting." *Image Vision Comput.*, vol. 13, no. 3, pp. 169–177, 1995. [Online]. Available: http://dblp.uni-trier.de/db/journals/ivc/ivc13.html

[14] L. O'Gorman, "Subpixel precision of straight-edged shapes for registration and measurement," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 7, pp. 746–751, Jul. 1996. [Online]. Available: http://dx.doi.org/10.1109/34.506796

[15] A. Efrat and C. Gotsman, "Subpixel image registration using circular fiducials." *Int. J. Comput. Geometry Appl.*, vol. 4, no. 4, pp. 403–422, 1994. [Online]. Available: http://dblp.uni-trier.de/db/journals/ijcga/ijcga4.html

[16] D. Havelock, "Geometric precision in noise-free digital images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 1065–1075, 1989.

[17] L. O'Gorman, A. Bruckstein, C. Bose, and I. Amir, "Subpixel registration using a concentric ring fiducial," in *Pattern Recognition, 1990. Proceedings., 10th International Conference on*, vol. ii, jun 1990, pp. 249 –253 vol.2.

[18] S.-W. Shih and T.-Y. Yu, "On designing an isotropic fiducial mark." *IEEE Transactions on Image Processing*, vol. 12, no. 9, pp. 1054–1066, 2003. [Online]. Available: http://dblp.uni-trier.de/db/journals/tip/tip12.html

[19] D. Havelock, "The topology of locales and its effects on position uncertainty," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, pp. 380–386, 1991.

[20] J. A. Gutiérrez and B. S. R. Armstrong, *Precision Landmark Location for Machine Vision and Photogrammetry: Finding and Achieving the Maximum Possible Accuracy.* London: Springer Verlag, 2008.

[21] C. B. Bose and I. Amir, "Design of fiducials for accurate registration using machine vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 12, pp. 1196–1200, Dec. 1990. [Online]. Available: http://dx.doi.org/10.1109/34.62609

[22] L. O'Gorman, A. M. Bruckstein, C. B. Bose, and I. Amir, "A comparison of fiducial shapes for machine vision registration." in *MVA*, 1990, pp. 253–256. [Online]. Available: http://dblp.uni-trier.de/db/conf/mva/mva1990.html

[23] Q. Tian and M. N. Huhns, "Algorithms for subpixel registration," *Comput. Vision Graph. Image Process.*, vol. 35, no. 2, pp. 220–233, Aug. 1986. [Online]. Available: http://dx.doi.org/10.1016/0734-189X(86)90028-9

[24] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach.* Prentice Hall Professional Technical Reference, 2002.

[25] A. M. Bruckstein, L. O'Gorman, and A. Orlitsky, "Design of shapes for precise image registration," *IEEE Trans. Inf. Theor.*, vol. 44, no. 7, pp. 3156–3162, Sep. 2006. [Online]. Available: http://dx.doi.org/10.1109/18.737548

[26] "Uncertainty characterization in image-based measurements: A preliminary discussion," *IEEE Trans. Instrumentation and Measurement*, vol. 49, no. 5, pp. 1101–1107, 2000.

[27] D. Robinson, S. Member, P. Milanfar, and S. Member, "P.: Fundamental performance limits in image registration," *IEEE Transactions on Image Processing*, 2004.

[28] J. H. Kim and C.-H. Menq, "Visually Servoed 3-D Alignment of Multiple Objects with Subnanometer Precision," *IEEE Transactions on Nanotechnology*, vol. 7, no. 3, pp. 321–330, May 2008. [Online]. Available: http://dx.doi.org/10.1109/TNANO.2007.914997

[29] D. C. Montgomery, E. A. peck, and G. G. Vining, *Introduction to Linear Regression Analysis, Fourth Edition.* Wiley, 2006.

[30] S. Chapra and R. Canale, *Numerical Methods for Engineers, 5th edition.* McGraw Hill, 2006.

[31] "Edge detection with sub-pixel accuracy based on approximation of edge with erf function," *Radioengineering*, vol. 20, no. 2, pp. 516–524, 2011.

[32] G. Chiorboli and G. P. Vecchi, "Comments on "design of fiducials for accurate registration using machine vision"," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 12, pp. 1330–1332, 1993.