LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

UCRL-PROC-223352

# Best Angle to Orient Two Intersecting Lines

A. A. S. Awwal, S. W. Ferguson, P. B. Shull

August 3, 2006

SPIE Optics & Photonics
San Diego, CA, United States
August 13, 2006 through August 17, 2006

# Best angle to orient two intersecting lines

Abdul A. S. Awwal, Walter Ferguson and Peter Shull*
Laser Engineering Division, National Ignition Facility
University of California, Lawrence Livermore National Laboratory, Livermore, CA. 94551
*Mechanical Engineering, Stanford University, Summer student at LLNL
*E-mail:* awwal1@llnl.gov

## ABSTRACT

Fiducials in the form of intersecting straight lines are used to align the target in the final target chamber of the National Ignition Facility of Lawrence Livermore National Laboratory. One of the techniques used to locate these lines is the Hough transform. When two lines intersect at a 90 degree angle, it is tempting to orient the lines to horizontal and vertical directions. There are other possible angles at which the lines may be oriented. One question that arises while designing the fiducials is whether there is a preferred angle or range of angles that leads to higher accuracy. This work attempts to answer this question through detailed computer simulation.

Key word: Line detection, Hough transform, Hough filtering, automated optical alignment, rotation angle, image processing

## 1. INTRODUCTION

The National Ignition Facility (NIF), currently under construction at the Lawrence Livermore National Laboratory, is a stadium-sized facility containing a 192-beam, 1.8-megajoule, 500-terawatt, ultraviolet laser system for the study of inertial confinement fusion and the physics of matter at extreme energy densities and pressures [1]. The alignment of this high energy pulsed laser is a critical process requiring accurate measurement. If the beam alignment is not performed properly, expensive optics could be damaged. An automatic alignment (AA) system was designed and implemented to ensure successful delivery of high energy pulse in each of the 192 beams. Currently, 16 beam lines are operational. All the 192 beams will be operational by 2010, which is the start of the National Ignition Campaign.

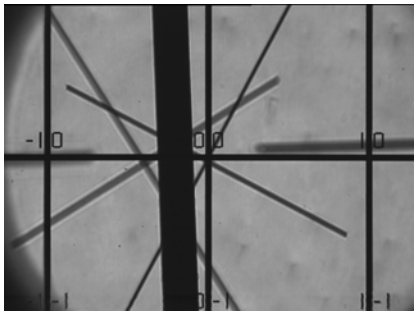

**Fig. 1: TAS CCRS**

A typical image of a target chamber fiducial is shown in Fig. 1. The algorithm needs to find the intersection of the two orthogonal lines that are located to the right of the thick vertical line at the center of Fig. 1. In this paper, we artificially create these lines with various line thicknesses and rotations to evaluate the effect of these variables on the accuracy of

the position measurement. One of the challenges of the simulation is to identify the effect of discretization of the line drawing algorithm on the measurement accuracy.

## 2. BACKGROUND

Hough transform (HT) is a technique for detecting features of a particular shape within an image [2,3]. In one application of HT, it allows detection of lines in an image. In practical implementation of Hough transform [3] each line is represented by two parameters, namely $r-\theta$, which are the distance and the angle of the normal to the to-be-detected line from the center of the co-ordinate system as shown in Fig. 2. With these parameters the equation of the

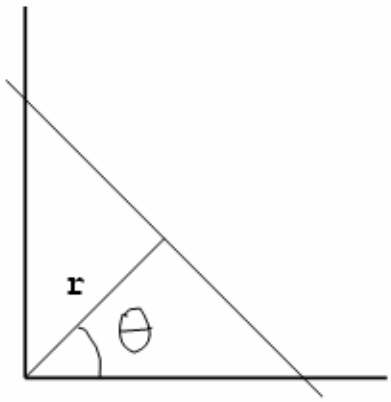


**Fig. 2: A typical line showing the parameters for Hough transform**

straight line is given by

$$x\cos\theta + \quad y\cos\theta = d \qquad \textbf{(1)}$$

Note that a line can be represented by a single $r-\theta$ value. A point is treated as intersections of an infinite number of lines. By calculating the value of $r$ for every possible $\theta$ value for all these lines through a point, a sinusoid curve is created in the $r-\theta$ plane, also known as the Hough plane. The contributions from all such points are accumulated in the Hough space. Two points located on the same line must have common $r$ and $\theta$ values. The two sinusoids corresponding to these points belonging to a line intersect at a common $(r_i, \theta_i)$ in the parameter space. Thus, any straight line with a fixed $(r_i, \theta_i)$ value will appear to have a high peak in the accumulated Hough space, which will be proportional to the number of pixels belonging to the line. Detecting the peak in the Hough space detects the straight lines with the corresponding $(r_i, \theta_i)$ value. A polynomial interpolation in the Hough domain near the peak is used for the peak detection [4,5]

## 3. THE ALGORITHM

In order to determine an optimal orientation angle of two intersecting lines analyzed by the Hough transform, pairs of intersecting lines of varying angle and shading were created and analyzed through a line generation algorithm. Given an intersection point, the algorithm generated 81 pairs of lines at orientation angles between -2 and 2 degrees of step size 0.05 degrees for each unique line shading scheme. The lines were chosen to be 9 pixels wide, because this width matched most closely with test image from the NIF facility. As shown in the block diagram of Fig. 3, each loop of the algorithm drew a horizontal and a vertical line 90 degrees apart at a given orientation angle, performed a Hough transform on the lines to determine the intersection point, compared this intersection point with the actual predetermined intersection point, and saved the absolute distance between the two as the error in pixels for the orientation angle. The error for each orientation angle and for each shading scheme was stored in an array for further analysis.

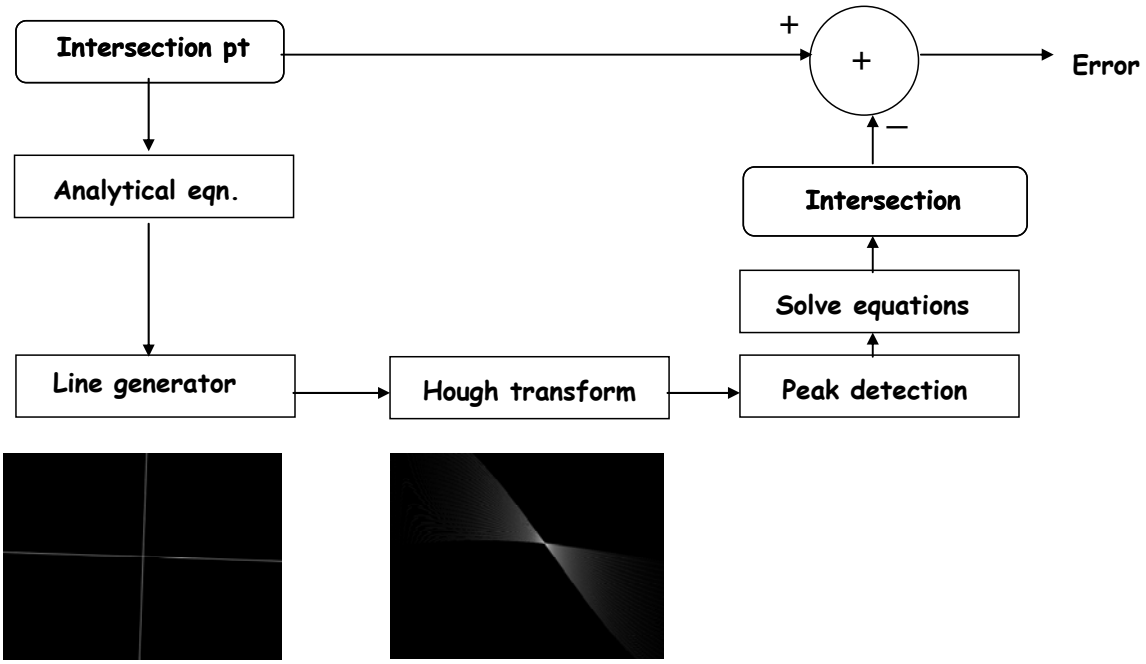


**Fig. 3. The simulation block diagram, shows different computational block**

Two different techniques were used for line drawing. In one case, the theoretical line equation is used to light up pixels after rounding the x,y values. In the second case, two neighboring pixels may be lit up depending on the distance of the line from the two pixels. For a particular pixel, the pixel intensity varies inversely with the distance from the pixel center to the line. If the line passes between two pixel centers, then the intensity will divide inversely proportional to the distance from the respective pixel centers. This is specifically useful when the line is moved by a fraction pixel. In the former case, the fraction moves may not lead to a new line for certain angles, as we shall see later. A 30 degree line passing through the pixel (2,2) is depicted in Fig. 4. When the line passes through the pixel center, the pixel gets the full intensity of 200 counts (such as pixel 2,2 in Fig. 4, assume the lower left lighted pixel is 0,0). If it is away from the center pixel, it loses part of its intensity to the pixel it is next nearest. The intensity is linearly distributed between two pixels.

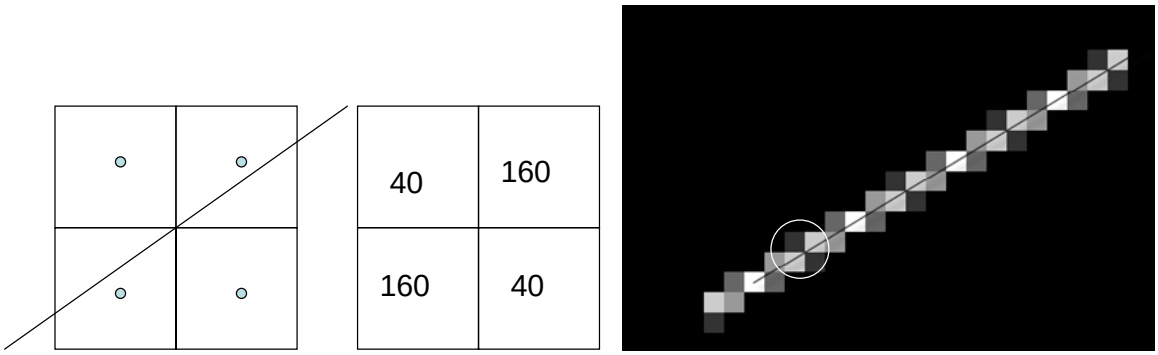


**Fig. 4: A 30 degree showing pixel intensity distribution**

The line generation algorithm performed seven different line shading levels for 9-pixel width lines. Both the unshaded 1-pixel line and 9-pixel wide lines are depicted in Fig. 5. Shading was applied in a descending fashion from the center of the line outward and was symmetric on each side. The center of the line was always set to the maximum brightness level of 255 and showed up as white. A brightness level of 0 showed up as black. Shading levels ranged from no shading to extreme shading where the edges of the line were almost black. Five of the shading levels used a linear shading method such that adjacent pixels outward from the center decreased in brightness in a linear fashion. For example, a shading level of 0_60_120_etc would mean that the center would be at a brightness level of 255, pixels that were one pixel away from the center would be at a level of 195, pixels that were two pixels from the center at 135, pixels that were three pixels from the center at 75, and pixels at four pixels from the center at 15 (Fig. 6). One other shading method was used where the pixels values decreased according to the function SinX / X away from the center of the line. This method closely approximated actual lines as seen by the alignment cameras. Fig. 7 depicts a comparison of SinX / X shading and the actual shading of a laser line on NIF.

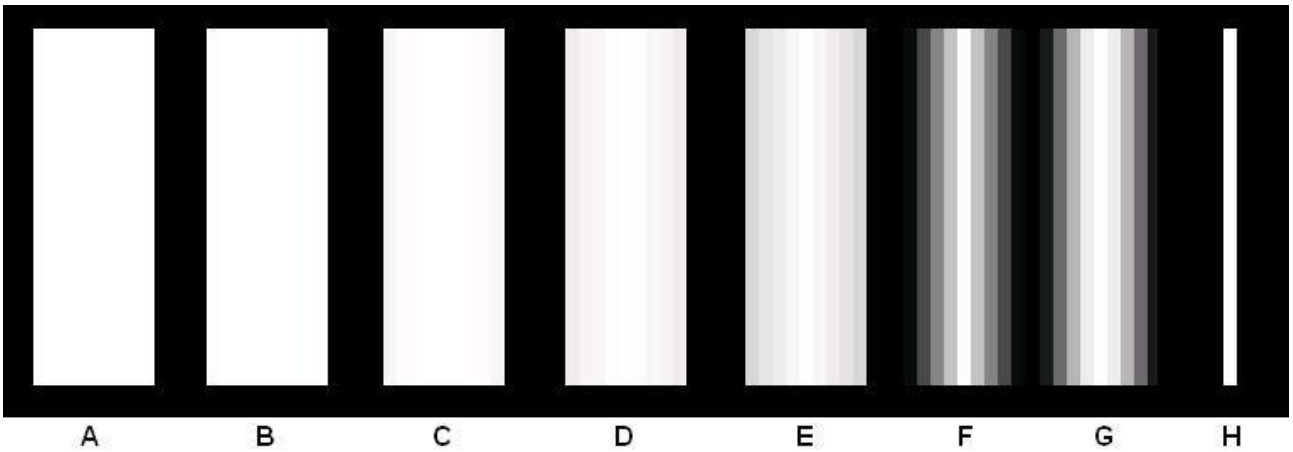


**Fig. 5. Different shading methods for the line generation algorithm.**

From Fig. 5, **A** – No Shading, **B** – 0_1_2_etc Shading, **C** – 0_2_4_etc Shading, **D** – 0_5_10_etc Shading, **E** – 0_10_20_etc Shading, **F** – 0_60_120_etc Shading, **G** – SinX / X Shading, **H** – 1 pixel-width, no shading. All lines are 9-pixels wide except for **H**. **F** and **G** look narrower, because their outer pixels are approximately the same brightness as the background. All shading was symmetric on either side. **B-F** shading decreased linearly outward from the center. **G** shading decreased according to the function SinX / X from the center.

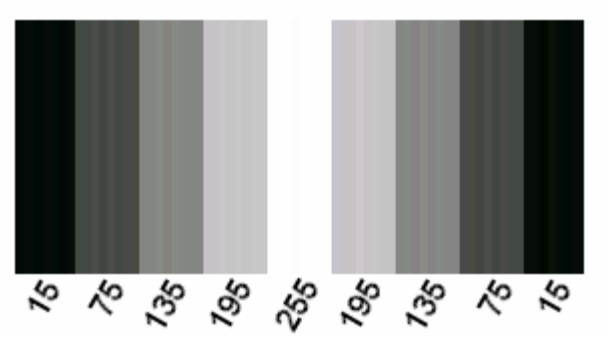


**Fig. 6. A closer look at line F on Fig. 3. Numbers indicate the pixel brightness level for the 9-pixel wide line.**

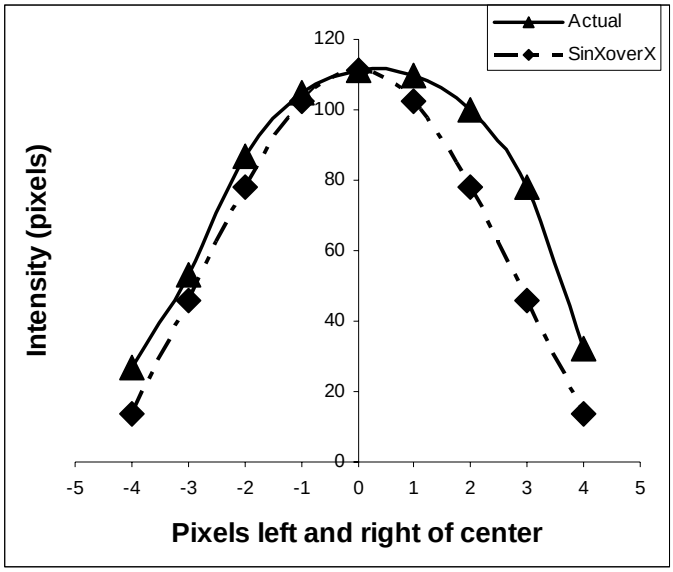


**Fig. 7. SinX / X shading compared with actual line shading.**

The Hough transform algorithm measures the position of each line separately. Each line produces a peak value corresponding to the line strength in the Hough plane as shown in the onset of Fig. 3. The locations of these peaks are determined after applying a polynomial interpolation through the peak values. Similarly, the parameters for the other line is also measured. The algorithm then numerically calculates the intersection of the two lines using the parameters for each line. Next we discuss the results from a single pixel line and the 9-pixel lines.

## 4. SIMULATION RESULTS

Single pixel line pairs with 81 different angular orientations are generated centered around (320,240) rotated at 0.05 degree intervals starting from -2 degree up to +2 degree. As stated earlier, the error is the difference between the input intersection point and that calculated by solving the parametric line equations and generated by the Hough transform algorithm. First, the error when the intersection points are two neighboring integer pixel locations are calculated as shown in Fig. 9.

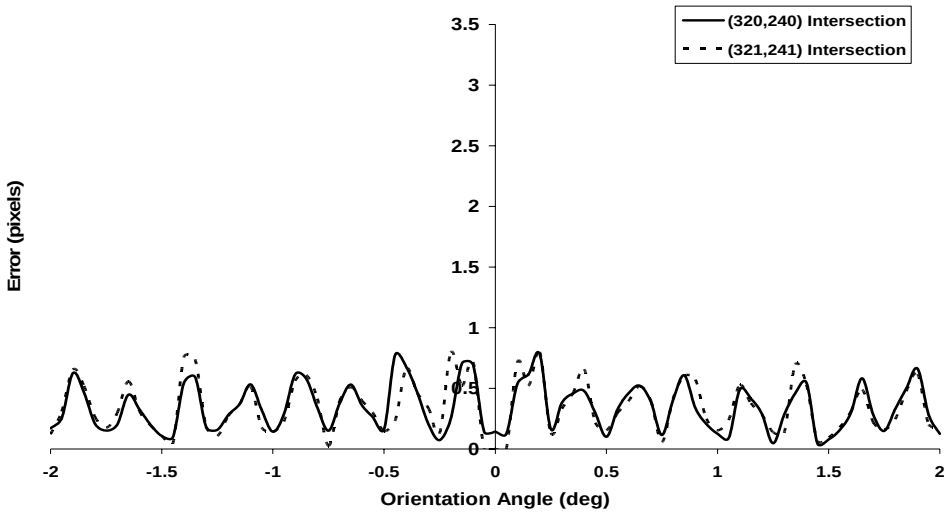


**Fig. 8: Single pixel line error vs. orientation angle at two neighboring integer intersection points.**

Fig. 8 demonstrates that the error has the same trend for both integer intersections. Next we evaluate the errors for 25 points in between these two integer pixel locations for the single pixel line. These intersections points are listed below.

(320.0 240.0) (320.2 240.0) (320.4 240.0) (320.6 240.0) (320.8 240.0)
(320.0 240.2) (320.2 240.2) (320.4 240.2) (320.6 240.2) (320.8 240.2)
(320.0 240.4) (320.2 240.4) (320.4 240.4) (320.6 240.4) (320.8 240.4)
(320.0 240.6) (320.2 240.6) (320.4 240.6) (320.6 240.6) (320.8 240.6)
(320.0 240.8) (320.2 240.8) (320.4 240.8) (320.6 240.8) (320.8 240.8)

For each angular orientation, 25 different error terms are generated. The statistics in terms of min, max and mean errors generated by these 25 pair points are depicted in Fig. 9. One surprising result is that the highest error appears to be in the 0 degree slope lines for the 25 intersecting points. We further investigated this result.

We compared the image at 0 degree angle and intersecting at an integer location with those images at 0 degree but at other intersection points. The sum of the absolute value of the difference (between the image at integer and at those at fractions) image divided by the maximum possible pixel value (255) is listed below for the zero degree angle.

Angle = 0, single pixel line

| | | | | |
|---|---|---|---|---|
| 0 (320.0 240.0) | 0 (320.2 240.0) | 0 (320.4 240.0) | 958 (320.6 240.0) | 958 (320.8 240.0) |
| 0 (320.0 240.2) | 0 (320.2 240.2) | 0 (320.4 240.2) | 958 (320.6 240.2) | 958(320.8 240.2) |
| 0 (320.0 240.4) | 0 (320.2 240.4) | 0 (320.4 240.4) | 958 (320.6 240.4) | 958 (320.8 240.4) |
| 1280 (320.0 240.6) | 1280 (320.2 240.6) | 1280 (320.4 240.6) | 2234 (320.6 240.6) | 2234 (320.8 240.6) |
| 1280 (320.0 240.8) | 1280 (320.2 240.8) | 1280 (320.4 240.8) | 2234 (320.6 240.8) | 2234 (320.8 240.8) |

The 9 zeros in the first three rows and columns indicate that the lines intersecting at 8 different locations are identical to those intersecting at the (320,240) location. Thus these 9 points, when calculating the error, had used 9 different absolute locations as the correct intersection but are getting the same result from the Hough transforms algorithm. Thus the error will be increasing as the intersecting point moves away from the integer location. This means that only one out of these 9 results are really correct from a simulation point of view. In fact for a single pixel line at 0 degree angle only 4 out of 25 results are valid. Twenty-one of the results are not valid, which results in an escalation of error at the 0 degree orientation as depicted in Fig. 9. Note that at other angles there are fewer lines that are similar to the line passing through the integer intersection as indicated by the image difference in terms of number of pixels as shown below.

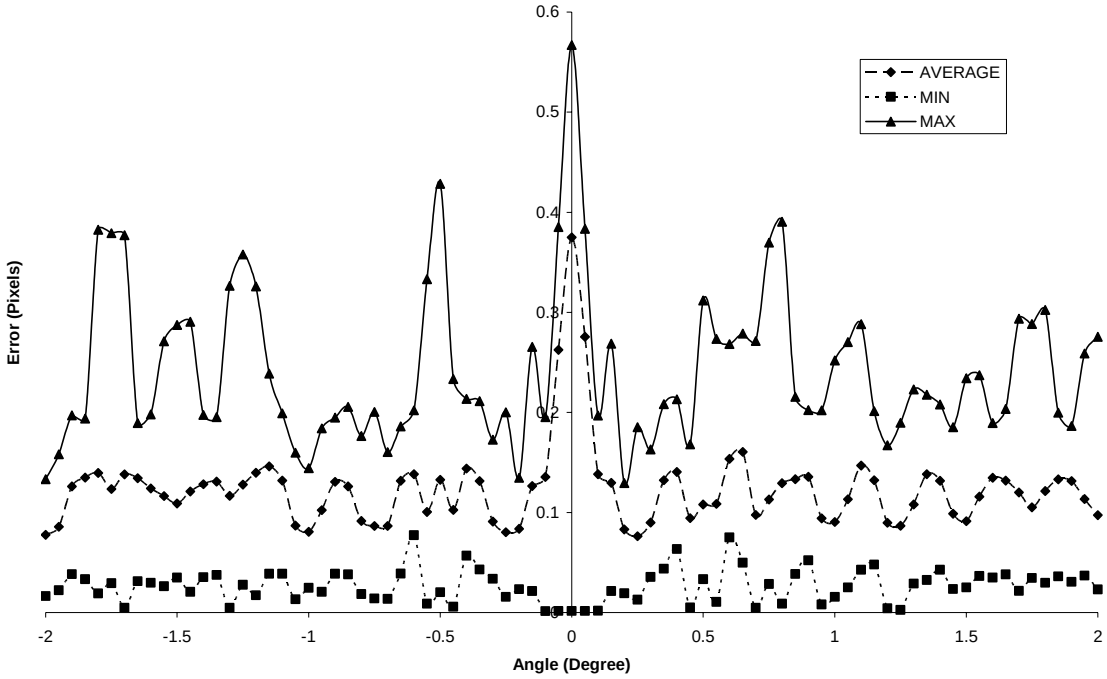


**Fig. 9: Single pixel line error vs. orientation angle at integer and fractional pixel**

Angle = 0.05, single pixel line (image difference)

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 250 | 708 | 958 |
| 0 | 0 | 250 | 708 | 958 |
| 410 | 410 | 660 | 1118 | 1366 |
| 868 | 868 | 1120 | 1574 | 1820 |
| 1280 | 1280 | 1532 | 1986 | 2234 |

… …. ….

Angle = 0.35, single pixel line (image difference)

| | | | | |
|---|---|---|---|---|
| 0 | 186 | 382 | 584 | 780 |
| 264 | 446 | 640 | 834 | 1032 |
| 526 | 708 | 904 | 1100 | 1294 |
| 758 | 942 | 1136 | 1328 | 1524 |
| 1020 | 1204 | 1402 | 1592 | 1786 |

It is possible to have similar discretization effect on the angle which may cause the error to go up for certain angles. One way the angle discretization is improved is through interpolation of the Hough domain data. In any case, it may be noted that the mean error is usually under 0.2. This number could be reduced by half (to 0.1) by using a Hough transform with higher resolution, however, at the expense of increased computational time. Currently, 720 points are used in the angular direction in the Hough domain. Surprisingly, the same discretization effect is observed in the case of 9-pixel lines as shown below.

Angle = 0, 9-pixel line (image difference)

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 958 | 958 |
| 0 | 0 | 0 | 958 | 958 |
| 0 | 0 | 0 | 958 | 958 |

| | | | | |
|---|---|---|---|---|
| 1280 | 1280 | 1280 | 2222 | 2222 |
| 1280 | 1280 | 1280 | 2222 | 2222 |

Angle = 0.35, 9-pixel line (image difference)

| | | | | |
|---|---|---|---|---|
| 0 | 186 | 382 | 584 | 780 |
| 264 | 446 | 640 | 834 | 1032 |
| 526 | 708 | 904 | 1100 | 1294 |
| 758 | 942 | 1136 | 1317 | 1513 |
| 1020 | 1204 | 1402 | 1581 | 1775 |

The actual errors at each of these points are shown below. Note that the error in the first 9 locations of the error matrix below increase by 0.2. This is because the first 9 lines are actually identical, where as the theoretical intersection points are changing by 0.2 pixels. Thus the error keeps creeping up as the difference between theoretical and actual intersection widens.

Angle = 0, 9-pixel line (error)

| | | | | |
|---|---|---|---|---|
| 0213056 | 0.201695 | 0.401675 | 0.398287 | 0.198308 |
| 0.201316 | 0.284964 | 0.449295 | 0.446269 | 0.282577 |
| 0.401310 | 0.449139 | 0.567792 | 0.5654 | 0.447628 |
| 0.398365 | 0.446523 | 0.565736 | 0.563314 | 0.444990 |
| 0.198372 | 0.282913 | 0.448011 | 0.444949 | 0.280486 |

Angle = 0.35, 9-pixel line (error)

| | | | | |
|---|---|---|---|---|
| 0.0214212 | 0.0713661 | 0.213658 | 0.352252 | 0.169911 |
| 0.0190647 | 0.0746261 | 0.215787 | 0.353544 | 0.166505 |
| 0.120485 | 0.139880 | 0.248298 | 0.370469 | 0.203927 |
| 0.225438 | 0.236561 | 0.311108 | 0.416717 | 0.275592 |
| 0.398584 | 0.403973 | 0.456035 | 0.527771 | 0.426951 |

Fig. 10 shows an example of this error variation for 9-pixel line for three intersection point locations of (320,240), (321,241) and (321,240) over the range from -2 degrees to 2 degrees. These three graphs closely follow each other for most of the range. The error in between 25 fractional points are plotted in Fig. 11.

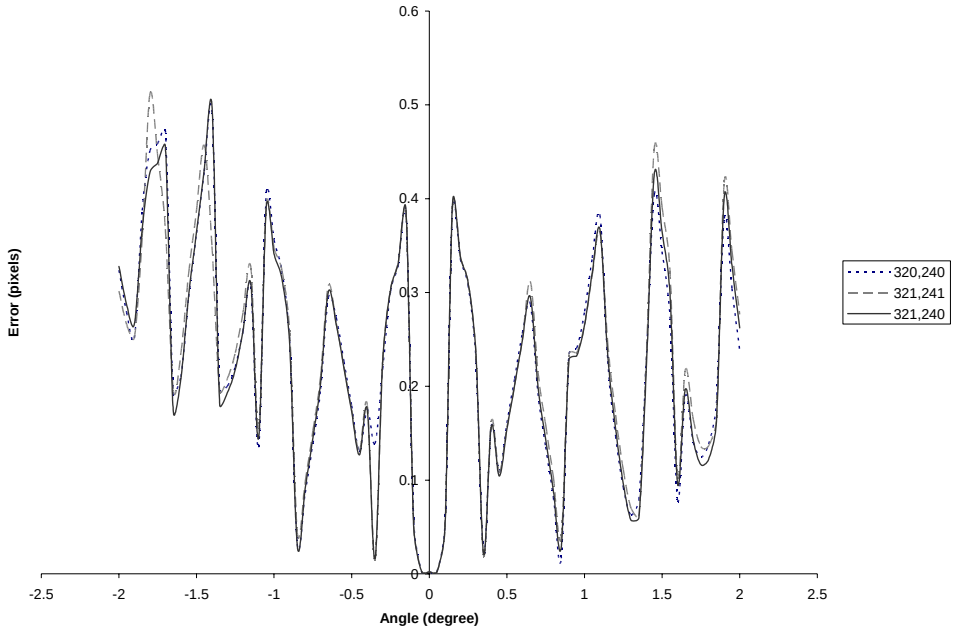


**Fig. 10. Nine-pixel wide line error vs. orientation angle at integer pixel locations**

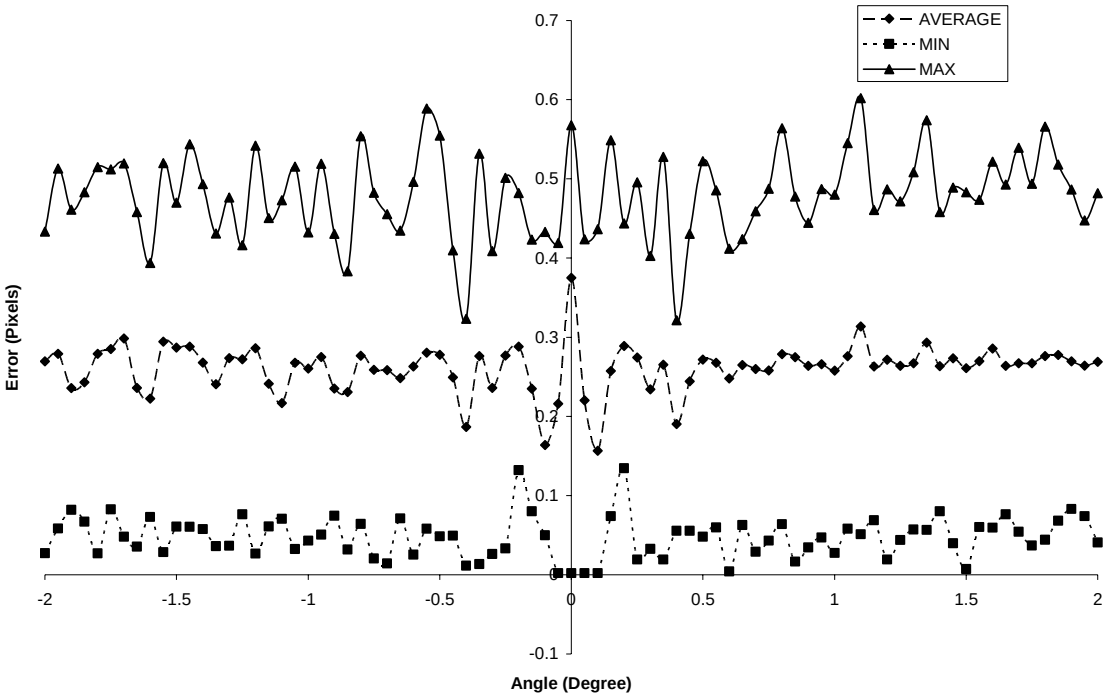


**Fig. 11. The error vs. orientation angle for a 9-pixel sinX/X line at fractional intersection point.**

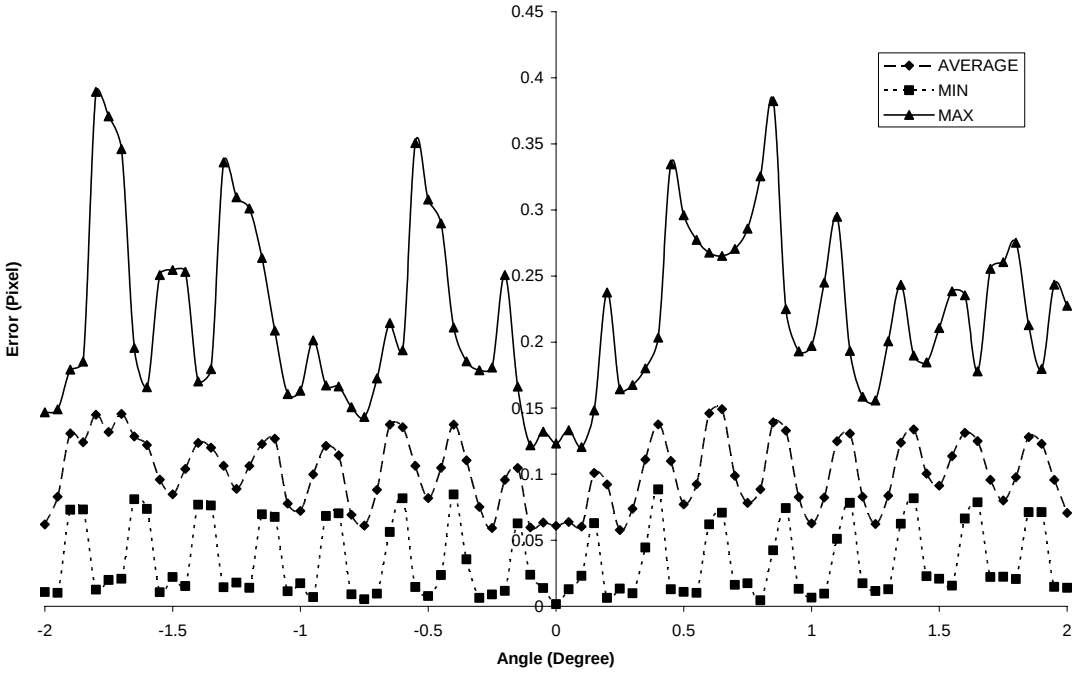


**Fig. 12. Single pixel line with weight sharing, error versus the orientation angle.**

To overcome the problem of single pixel lines representing fractional intersection points, the weight sharing scheme as explained in Fig. 4, is implemented. The error of this new line is evaluated at 36 points between (320,240) and (321,241) with a 0.2 pixel interval. The min, max and average values are shown in Fig. 12. Compared to the results in Fig. 9 for single pixel lines, the Fig. 12 shows considerable improvement near the origin. It appears that this line favors a 0 degree orientation. However, mean error remains below 0.15 level. It is possible that the high error points are problems arising from a combination of line representation and Hough domain resolution.

## 5. ANALYSIS

The Hough transform is being used to locate and update intersecting lines whose intersection point is constantly changing, therefore the simulation tested lines at various fractional pixel location between two integer points. The results show that the error is a function of three variables, position (x,y) and angle.

We have shown how the line drawing algorithm affects the accuracy of the simulation. When lines are drawn using single pixels, fewer unique lines are created when passing through fractional intersection points. As such the error increases. The simulation results are further improved by allowing two pixels to light up, depending on the distance between the line center and the neighboring pixels.

Shading applied to the 9-pixel line proved to be a more effective way to lower the error. Shading lowers the overall error for any line intersection point. This is because when two 9-pixel width unshaded lines intersect, the intersection will actually be a 9 by 9 pixel square with pixels of all the same value. It is difficult for the Hough transform to determine which of the pixels in the 9 by 9 square is the intersection point. When shading occurs, the pixels away from the center of the 9 by 9 square will be dimmer, and the Hough transform will be able to converge on the actual center much more accurately.

SinX / X shading was nearly as effective at lowering the error as the most intense linear shading. SinX / X error was also near that of the 1 pixel width line with no shading. It is unlikely that any shading scheme could produce error much less than that of a 1 pixel width line, because shading a wide line effectively narrows the line for the Hough transform. The more the shading, the thinner the line will appear until all that is left is a 1 pixel width line. Since SinX / X shading is nearly as effective as the best shading methods and because SinX / X shading closely approximates the actual line's natural shading, it seems that for this particular application on NIF, shading will have little effect on improving accuracy and reducing error.

## 6. CONCLUSIONS

In this paper, a study is conducted to observe the effect of changing the angular orientation of two perpendicularly intersecting lines. The effects of shading and location of the intersecting point are studied. Two sources of errors are expected. One resulting from the line detection algorithm and the other originating from the line drawing algorithm.

It was shown that when a line is drawn using a single pixel rounding, there are certain rotation angles that should be avoided. The recommendation for a single pixel line is to not orient the line between 0 to 0.2 degree. This is true when the intersection falls on a fractional pixel boundary. Statistically it is most likely to fall on a fractional boundary (Fig. 10). The only exception is when it falls on an integer boundary, the error is minimum near the origin (Fig. 9). The source of this error was identified to be the inability of the single pixel line to represent a unique line when fractional movement of pixel occurs. For practical application, this recommendation is only valid if the line drawn is a single pixel line and does follow a rounding based algorithm. On the other hand, if we allow the light to be distributed into two pixels based on the distance from the center of a pixel, then the origin appears to be the better choice (Fig. 13). For 9-pixel lines using rounding, the recommendation is similar. If rounding is used and the position of the line can be controlled to be on an integer pixel, then 0 degree seems to be the better choice (Fig. 11). However, if all possible positions are likely, i.e. both fractions and integer, the zero degree rotation should be avoided and fairly low average error such 0.3 pixels is observed outside this range (Fig. 12).

Another observation is that the error seems to increase periodically (Fig. 13) for certain angles. To understand the source of the higher error, several improvements to the simulation could be made. For example, a metric representing the error in line representation could be introduced. This could be the mean square difference between the theoretical line and the discretized line. The other change could be in modifying the line drawing algorithm to reflect what practically takes place when an optical image of a line is projected on the CCD camera.

An ideal simulation would be to create the line optically and use microscope alignment tools to project the line to an integer pixel, and then introduce a relative rotation and translation between the camera and the image source. The algorithm could then be used to find the intersection and compare it with the measured intersection.

The utility of this simulation is that if the most practical line representation is chosen, then the accuracy of the algorithm can be optimized by testing the variations of the algorithm and choosing the one that provides the best accuracy.

## ACKNOWLEDGEMENT

Discussion with Milt Latta are acknowledged. AASA acknowledges the help of Clement Law in drawing some of the figures in this paper. This work was performed under the auspices of the U.S. Department of Energy by the University of California, Lawrence Livermore Laboratory under contract No. W-7405-Eng-48.

## REFERENCES

1. E. Moses, et al., “The National Ignition Facility: Status and Plans for Laser Fusion and High-Energy-Density Experimental Studies”, *Fusion Science and Technology*, V. 43, p. 420, May 2003.
2. P. V. C. Hough, “Methods and means for recognizing complex patterns,” U. S. Patent 3,069,654.
3. R. O. Duda, and P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," *Comm. ACM, Vol. 15*, pp. 11–15, 1972.
4. A. A. S. Awwal, “Hough transform based corner detection for laser beam positioning” in *Photonic Devices and Algorithms for Computing VII*, edited by K. Iftekharuddin and A. A. S. Awwal,, Proc. of SPIE 5907 (SPIE Bellingham, WA, 2005), 59070I, 2005.
5. A. A. S. Awwal, Wilbert A. McClay, Walter S. Ferguson, James V. Candy, Thad Salmon, and Paul Wegner, “Detection and Tracking of the Back-Reflection of KDP Images in the presence or absence of a Phase mask,” Applied Optics, Vol. 45, pp. 3038-3048, May 2006.