UCRL-PROC-223349

LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

Multi-object Feature Detection and Error Correction for NIF Automatic Optical Alignment

A. A. S. Awwal

August 3, 2006

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

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

Multi-object feature detection and error correction for NIF automatic optical alignment

Abdul A. S. Awwal
Laser Engineering division, National Ignition Facility
University of California, Lawrence Livermore National Laboratory, Livermore, CA. 94551
E-mail: awwal1@llnl.gov

ABSTRACT

Fiducials imprinted on laser beams are used to perform video image based alignment of the beams in the National Ignition Facility (NIF) of Lawrence Livermore National Laboratory. In any laser beam alignment operation, a beam needs to be aligned to a reference location. Generally, the beam and reference fiducials are composed of separate beams, as a result only a single feature of each beam needs to be identified for determining the position of the beam or reference. However, it is possible to have the same beam image contain both the beam and reference fiducials. In such instances, it is essential to separately identify these features. In the absence of wavefront correction or when image quality is poor, the features of such beams may get distorted making it difficult to distinguish between different fiducials. Error checking and correction mechanism must be implemented to avoid misidentification of one type of feature as the other. This work presents the algorithm for multi-object detection and error correction implemented for such a beam line image in the NIF facility. Additionally, we show how when the original algorithm fails a secondary algorithm takes over and provides required location outputs.

Key word: Pattern recognition, object identification, automated optical alignment, image processing, image quality detection and error detection

1. INTRODUCTION

The National Ignition Facility (NIF), under development at the Lawrence Livermore National Laboratory, is a stadium-sized facility designed to contain a 192-beam, 1.8-megajoule, 500-terawatt, ultraviolet laser system dedicated to 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 high accuracy error-free measurement [2]. If the beam alignment is not performed properly the goal of producing high energy density and pressure can not be achieved. Moreover, expensive optics could be damaged by misalignment. 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 out of 192 beams are operational. There are numerous images being processed throughout the beam path of each laser beam. One such image appears in the pre-amplifier module (PAM) as shown in Figure 1. In this image the fiducials corresponding to the reference and beams are located outside the area of the main beam but contained in the same image. It is important to identify these fiducials unambiguously. In the absence of wavefront correction or when image quality is poor, the features of such beams may get distorted, making it difficult to distinguish between different fiducials. Error checking and correction mechanism must be implemented to avoid misidentification of one type of feature as the other resulting in severe misalignment. This work presents a robust algorithm for multi-object detection and error correction implemented for the PAM automatic alignment images.

2. BACKGROUND

The basis of the detection algorithm utilized here is a classical matched filter (CMF). A CMF [3] can be defined by assuming that the Fourier transform of the object function *f*(x,y) be denoted by:

$$F(U_x,U_y) = \left|F(U_x,U_y)\right| \exp(j\Phi(U_x,U_y)) \quad \textbf{(1)}$$

Then the CMF corresponding to this function *f*(x,y), given by the complex conjugate of the input Fourier spectrum as denoted by Eq. 2,

$$H_{CMF}(U_x,U_y) = F^*(U_x,U_y) = \left|F(U_x,U_y)\right| \exp(-j\Phi(U_x,U_y)) \quad \textbf{(2)}$$

is expected to produce its autocorrelation. The performance of the matched filter can be further enhanced by performing edge detection on the image and using the edge of the to-be-detected features as the filter. This has an equivalent effect of high pass filtering the correlation plane output, thus increasing the sharpness of the peaks.

The position of the object can be found from the position of the crosscorrelation, autocorrelation, and the position of the template using the Eqs. 6-7.

$$\mathrm{x}_{\mathrm{pos}} = \mathrm{x}_{\mathrm{cross}} - \mathrm{x}_{\mathrm{auto}} + \mathrm{x}_{\mathrm{c}} \quad \textbf{(6)}$$

$$\mathrm{y}_{\mathrm{pos}} = \mathrm{y}_{\mathrm{cross}} - \mathrm{y}_{\mathrm{auto}} + \mathrm{y}_{\mathrm{c}} \quad \textbf{(7)}$$

where ($x_{pos,}y_{pos}$) is the to-be-determined position of the pattern in the image plane, ($x_{auto,}y_{auto}$) is the position of the template autocorrelation peaks and (x_{cross},y_{cross}) is the position of the cross correlation peak. The position of the cross-correlation peak was estimated using a polynomial fit of second order to the correlation peak. The center of the template, ($x_{c,}y_{c}$), and ($x_{auto,}y_{auto}$) are calculated off-line.

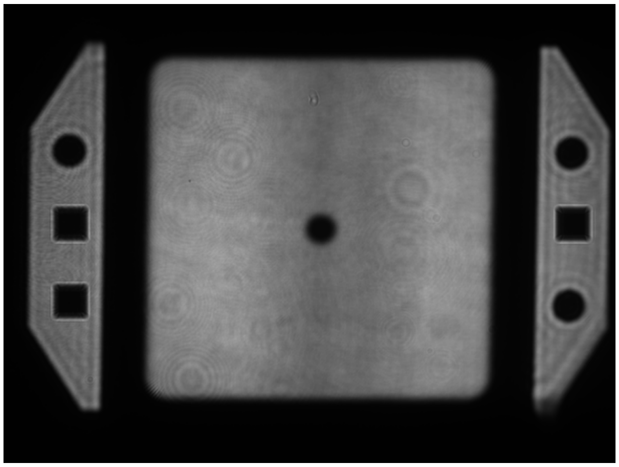

Fig. 1: The PAM output beam showing six fiducials and the square main beam.

3. ALGORITHM FOR DETECTING MULTIPLE BEAM FIDUCIALS

The video image of the preamplifier input/output laser beam depicted in Fig. 1 consists of three square fiducials corresponding to the reference position and three circles representing the beam position. The beam is essentially the inner illuminated square. When the beam moves, the three fiducial circles will move but the square will remain fixed. When the beam is aligned, the midpoint of the line joining the two diagonal circles must coincide with the center of the two square fiducials at the middle. The difference between these two positions serves as the error signal, which is used to drive the mirrors to co-align the two beams.

The overall algorithm for detecting multiple features of the PAM image is shown in the block diagram of Fig. 2. All NIF algorithms must perform a preprocessing in order to eliminate any false, black, white or truncated images. This is done in the off-normal detection stage [4]. When an image is found to be normal, it is passed to the matched-filter based algorithm, which processes the image and finds the location of the fiducials. If the position of each type of fiducial is accepted after the error checking and correction step, the algorithm outputs the positions of all the fiducials with their corresponding uncertainty [5]. If all the fiducials could not be identified or they are found but fail the error checking steps, then an alternate algorithm is automatically invoked [6]. This algorithm could be a centroid based technique. Again those fiducial positions obtained using the alternate algorithm undergo the same type of error checking and correction stage and if passed then they are sent to the control system. If the positions fail the error check, then a high uncertainty is flagged and the automatic alignment is stopped. Next we describe each of these blocks in more details.

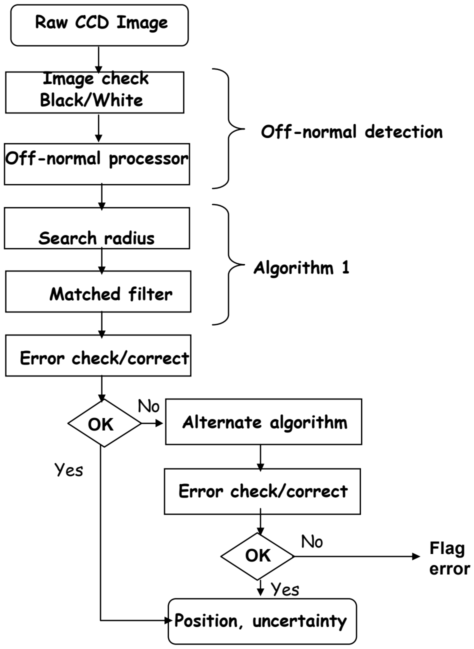

Fig. 2: The PAM image processing algorithm

The off-normal processor uses statistical methods to classify and detect simple black, white or dim images [4]. In addition, the off-normal processor checks for the size of the beam using average intensity profiles along the x- and y-direction. A general rule of thumb used is a 10% variation from nominal sizes as normal. However, historic image sets which are saved everyday from running of the NIF beams are analyzed to come up with the nominal values of all the beam parameters such as mean and max of intensity levels, min and max of sizes, etc. [7]. To keep the operation of off-normal processor up-to-date, off-normal images of various kinds are periodically added to a data base. These images are used as test images for the algorithm before it is released to the NIF facility. A typical output from the off-normal processor is shown below. Note that a normal image may have a size between 320 and 450 pixel. The measured sizes are 353 and 359 in the two orthogonal directions, consequently, this image in Fig. 1 passes the off-normal test. Next we describe the matched-filter based algorithm.

```
*******************Printing expected normal size**************************
LOOPNAME :mpai_cl_beam
Expected_size of the inner square beam is between     320 and     450
**********************************************************************
 Approving Off_Normal Method
NODE A IN OPERATION
IMAGE STANDARD DEVIATION       56.913021
NORMAL/OFF-Normal IMAGE---check FWHM
WORKING IN FUNCTION CHECK K
X-FWHM width =         359
Y-FWHM width =         353
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Image PASS PreProcessor
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

One of the prerequisites of using a matched filter is the existence of a template which faithfully represents the to-be-detected pattern. The first step of the PAM image processing algorithm is to determine the appropriate template corresponding to the circles and squares. It may be tempting to use our knowledge of the exact size of beam features from design specification. However, due to noise and other varying imaging conditions, the circles and squares could vary in size. Thus every beam line may have a slightly different size of the fiducials. Therefore, the search radius algorithm is used to estimate the exact sizes of each template for every beam line.

In the search radius algorithm, the image is smoothed, inverted and then thresholded to create an inverted binary image as shown in Fig. 3. We assume that sufficient smoothing has been done to ensure that each region that is normally connected remains connected. Next, the sizes of the different isolated regions are estimated and categorized based on their pixel count. Assume that the radii of the circles could vary between range $r_1 < r < r_2$. By searching each connected region based on its size between πr_1^2 and πr_2^2 , we can isolate each region of interest to include only one type of the pattern, namely the circles. By design, the diameter of the circles should be equal to the side of the squares, ideally the squares would vary between $4r_1^2$ and $4r_2^2$, and consequently, the areas of the squares will be larger than those of the circles. However, since the lenses have a tolerance of 10%, and the circles and the squares travel different path lengths to arrive at the camera plane, it is possible to have this criterion violated. Thus, in addition to the relative-size criterion, a position-based criterion is also applied. For a specific sized object, if it appears to violate the position criterion, then the object in the expected position is taken as the desired pattern. As shown below, the blob sizes 990, 948 and 937 are expected to be those of the circles and 1149, 1172 and 1146 belong to the square.

```
****************Beginning  a parameter search....................
Running search rad version 2.10 .................    updated 1/25/06
Looking for features in the range      250.000 to      1400.00
Feature sizes of interest         990         948        1149        1172
         778      1146           937
Found           7 spots in the valid range
Searching for big circles at      68.0000      153.000
Searching  for square around      69.5000      229.500
```

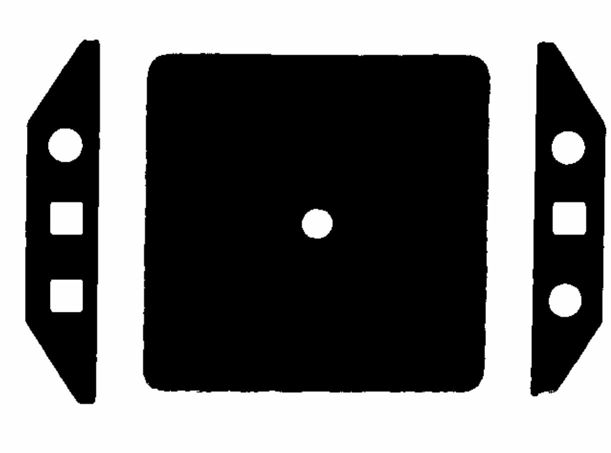

Fig. 3: The PAM output beam showing six fiducials and the square main beam.

Next, a search is executed to find the best match template for the circles (squares) from the small region containing the circle (square). For example, a search in this region with the side (radius) of the square (circle) varying between $r_1 - \Delta < r < r_2 + \Delta$ with an increment of half pixels provides the correlation peak for each r value as shown in the table below from which we can find the best dimension (radius) of the square (circle) that offers the maximum match. This size, 16.5 (17.5) pixels is chosen as the dimension of the template for the square (circular) object as shown below.

```
Searching for the square..........................
Max corr is      1.11739 at radius      14.0000
Max corr is      1.14551 at radius      14.5000
Max corr is      1.12663 at radius      15.0000
Max corr is      1.17022 at radius      15.5000
Max corr is      1.70921 at radius      16.0000
Max corr is      2.16980 at radius      16.5000
Max corr is      1.60087 at radius      17.0000
Max corr is      1.05316 at radius      17.5000
Max corr is      1.02470 at radius      18.0000
Max corr is      1.18370 at radius      18.5000
Max corr is     0.861071 at radius      19.0000
Max corr is     0.689603 at radius      19.5000
Max corr is     0.601807 at radius      20.0000
Returning from search_rad......................
*******Search of parameter (circle and square dimensions) completed....
Chosen Square side (half) =      16.5 Chosen Circle radius =      17.5
```

Note that the correlation peak drops almost by 50% for a 1 pixel difference from the chosen dimension of 16.5. In addition, the correlation peak will be broader at those locations, resulting in higher uncertainty about the result. This “search radius” program is also designed to detect more sophisticated off-normal conditions, such as having fewer than 6 fiducials. Such a condition causes an exit from the algorithm and will generate a high uncertainty.

Once the template size for each type of object is selected, the whole image is searched utilizing each template to find the occurrence of that type of object. Next we attempt to increase the detection accuracy by increasing the discrimination between the two to-be-detected objects. This is achieved by using the features of the objects such as edges. The edge image shown in Fig. 4 is obtained by performing a smoothing followed by a sobel edge operator and a subsequent thresholding.

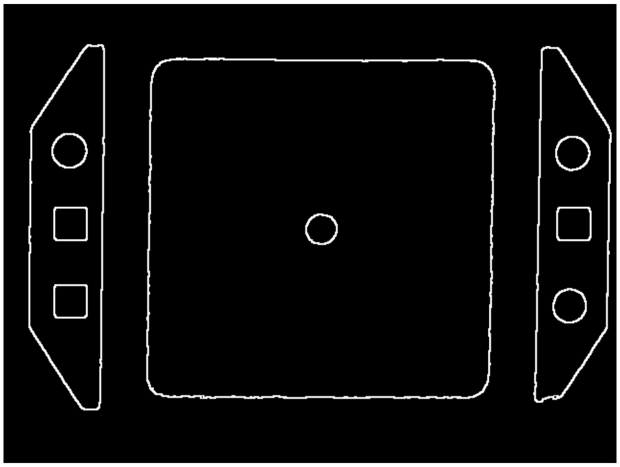

Fig. 4: The edge image generated from image of figure 1

Now using the edge of the circles as filters, the resulting correlation output as shown in Fig. 5 is obtained. The numerical values of the peaks are shown in the output below. Note that the autocorrelation of the circle is higher than the cross-correlation with the squares. Now based on the normalized autocorrelation value, we can set a threshold (as a percentage of the maximum peak, quality factor) to reject the correlation due to non-circles. The correlation plane output depicted in Fig. 5 demonstrates that both the auto- and cross-correlation can identify the two beam features, if the threshold is lowered. After selecting the circles, we correlate the image from Fig. 4 with a second template consisting of a square mask. Now using Eqs. 6-7, the position of the objects can be found from the position of the crosscorrelation peak, the autocorrelation peak, and the template. Again, using a threshold criterion based on the normalized correlation we can select the best matches while rejecting the other objects. A typical search of both circle and square objects produces the output shown below:

```
****************Finding the positions.................................
Looking for the outer squares with dimension      16.5000
xpos=      69.5186 ypos=      229.492 Quality=      1.00000 spots=       1
xpos=      69.5164 ypos=      310.480 Quality=     0.991139 spots=       2
xpos=      595.000 ypos=      229.500 Quality=     0.891139 spots=       3
*********Minimum quality factor accepted =      0.650000 and present 0.417722
Looking for the outer circles with radius      17.5000
xpos=      68.3298 ypos=      152.998 Quality=      1.00000 spots=       4
xpos=      593.716 ypos=      155.668 Quality=     0.960000 spots=       5
xpos=      590.555 ypos=      315.245 Quality=     0.915000 spots=       6
*********Minimum quality factor accepted =      0.600000 and present  0.448333
```

The three circles are detected using a normalized correlation and a threshold of 0.60 of the maximum peak. However, it is possible to have more than 3 peaks detected when the threshold is lowered. All the peaks detected above the threshold are passed on to the error handler to select the right spots. The error handler performs three different tests to validate any detected pattern as circles. These tests are the spacing test on individual positions, absolute position test for each of these points and spacing and position test taking a pair of points at a time. The spacing test dictates that each circle must have two other circle at a certain x and y position interval. Absolute position test states that the top left circle should be within the top one third of the area of the image. The pair test ensures that two squares should have the same y and a certain x distance. The user feedback from these tests are shown below:

```
Possible       3  squares and       3 circles  found...........
Looking for circular spots with x_spacing 536, y_spacing 164 and tolerance 50
Found       3 circle  spots after spacing test
Looking for square spots with x_spacing 536, y_spacing 82 and tolerance 50
```

```
Found        3 square  spots after spacing test
Passing manual position check ......
Passing redundant check for circles (2 at a time)...
Passing redundant check for squares (2 at a time)...
----------------------------------------------------------------------
Found   6 viable spots after triple  tests ...

mpa_ref_squares_beam_circles_eh:
      68.3298      152.998     0.167982
      590.555      315.245     0.167982
      69.5186      229.992     0.134385
      595.000      230.000     0.134385
      69.5164      310.980     0.134385
      593.716      155.668     0.167982

      …            …           …
      0.00000      0.00000      1.50000
JEDI MPAO reference square and beam circles  positions done
```

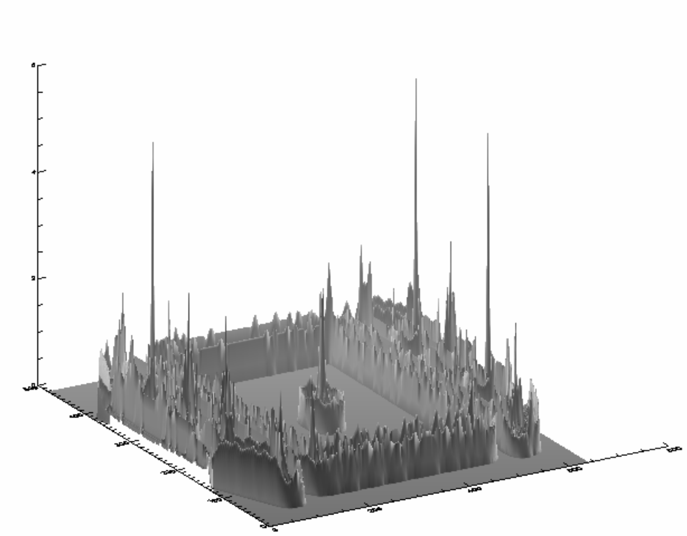

Figure 5. The correlation with circle with the image in Fig. 4

The above output shows the locations of the six fiducials (excluding the center spot). The first two entries are the two diagonal circles, the next two are the two squares in the middle. The last two are the remaining circle and square. To perform alignment, only the first four entries are required.

6. AUTO-SWITCH OF ALTERNATE ALGORITHM

There are cases when the matched filter based algorithm fails to identify the six fiducials because it often finds the circles to be of different sizes. As such, the maximum correlation peak for those different sized circles falls below the threshold. This happens for the image shown in Fig. 6. Note that the corners are detected by Hough transform [8] which was described in an earlier paper.

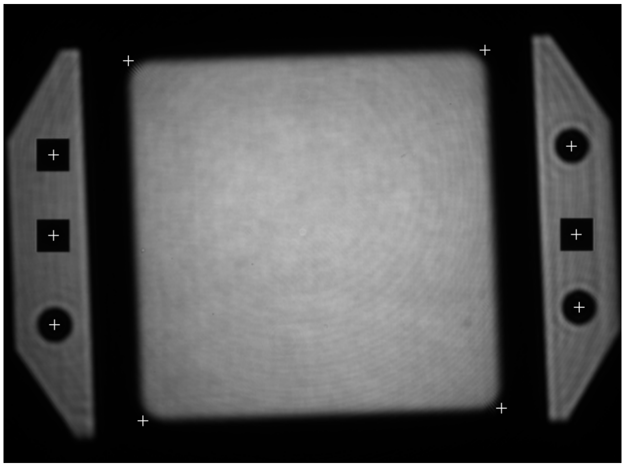

Figure 6. The PAM image processed by a centroid-based algorithm

```
****************Beginning  a parameter search...................
Running search rad version 2.10 .................   updated 1/25/06
Looking for features in the range      250.000 to      1400.00
Feature sizes of interest        1222        1046        1203        1120
        1209        1019
Found           6 spots in the valid range
Searching  for square around      47.5000      237.500
Searching for big circles at      52.0000      145.500
```

From the initial search it can be seen that the circle sizes vary from 1019 to 1120 and the square sizes vary from 1203 to 1222. The circles vary by more than 100 pixels. This could result in a difference of more than 1 pixel in terms of the dimension of the circle. As such, comparing the output of search radius, the difference of 1 pixel can cause more than 50% reduction in the normalized cross-correlation peak.

```
*******Search of parameter (circle and square dimensions) completed....
Chosen Circle radius =      19.5000 Chosen Square side(half) =      17.0000
Looking for the outer circles with radius      19.5000
xpos=      53.5898 ypos=      145.146 Quality=      1.00000 spots=       4
xpos=      593.259 ypos=      162.894 Quality=     0.561489 spots=       5
*********Minimum quality factor accepted =      0.5 and present  0.388350
****************Finding the positions.................................
Looking for the outer squares with dimension      17.0000
xpos=      51.9663 ypos=      320.821 Quality=      1.00000 spots=       1
xpos=      52.0436 ypos=      237.688 Quality=     0.982456 spots=       2
xpos=      591.500 ypos=      238.766 Quality=     0.708502 spots=       3
*********Minimum quality factor accepted =  0.65 and present 0.481781

Possible       3  squares and       2 circles  found..........
```

Note that the algorithm could not find three spots above 0.5 quality factor. Next, the algorithm executes the error checks.

```
Looking for circular spots with x_spacing 536, y_spacing 164 and tolerance 50
Did not find 3 spots but found   2 circle  spots after spacing test
       2 circles found, can not resolve them
Looking for the square spots with x_spacing 536, y_spacing 82 and tolerance 50
Found        3 square  spots after spacing test
Upper right circle failed position test
Failing redundant check for circles,x_spacing and y-spacing of 536 and 164
Passing redundant check for squares (2 at a time)...
----------------------------------------------------------------------
Found        3 viable spots after triple  tests ...

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Failing mpa_ref_squares_beam_circles_eh
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

Since the algorithm could not locate 6 spots, it declared it a failure. This triggered an automatic execution of the centroid based algorithm.

```
Invoking centroid based algorithm
           6 blobs found
Found        6 spots after spacing test
Passing redundant check for circles (2 at a time)...
Passing redundant check for squares (2 at a time)...
----------------------------------------------------------------------
Found   6 viable spots after triple  tests ...
Centroid results for mpa_ref_squares_beam_circles_eh:
      54.2612      145.760      1.12752
      586.393      330.367      1.12752
      52.9433      237.709      1.07752
      591.008      238.969      1.07752
      52.8169      320.701      1.17752
      593.991      163.791      1.17752
      …            …            …
-------------End of image
```

The locations of the 6 spots and their corresponding uncertainty are showed above. The example demonstrates an auto algorithm switching mechanism, allowing the PAM algorithm to work in a robust fashion when image quality changes..

6. CONCLUSIONS

In this paper, a method of detection of different fiducials from the same beam image is presented. One of the challenges of this real-time algorithm is the illumination gradient. This tends to change the strength of the correlation peaks, which may fall below a certain threshold and not be recognized. The other effect comes from difference of sizes of the circular objects. Even a one pixel change in radius could change the peak value by more than 50%, which may result in a match of a circle with a square higher than those of a circle with a circle of different radius. This in turn could introduce one of the squares misidentified as the circle. To overcome this problem, we have instituted three error checking mechanism. This error checking can filter out the incorrectly detected spots. On the other hand, a lower correlation will mean that all three fiducials could not be detected because it is less than the detection threshold. This may in fact leave the algorithm with insufficient spots detected. This results in declaring a failure for the original algorithm, which in turn triggers a centroid based algorithm to take over. Another side effect of the differing circle size, when they are still above the threshold is a rise in uncertainty of the location of the circle. In other words, increase in the error of the fiducial location. One possible solution to all these problems is to have a different template for each circle. This can be achieved by simultaneously tracking the correlation peak over each circle location while the radius of the template is changed. Then the best match template can be selected for each circle location. The square fiducials are usually more uniform, therefore one template for all the squares is adequate.

ACKNOWLEDGEMENT

This work was performed under the auspices of the U.S. Department of Energy by the University of California, Lawrence Livermore National 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. J. V. Candy, W. A. McClay, A. A. S. Awwal, and S. W. Ferguson, “Optimal position estimation for the automatic alignment of a high-energy laser,” Journal of Optical Society of America A, Vol. 22, pp. 1348-1356, 2005.
3. M. A. Karim and A. A. S. Awwal, Optical Computing: An Introduction, John Wiley, New York, NY, 1992.
4. J.V. Candy, A. A. S. Awwal, W. McClay, S. C. Burkhart, “Detection of off-normal images for NIF automatic alignment,” in *Photonic Devices and Algorithms for Computing VII*, edited by K. Iftekharuddin and A. A. S. Awwal,, Proc. of SPIE Vol. 5907, pp. 59070B-1 - 59070B-12, 2005.
5. W. A. McClay III, A. A. S. Awwal, H. E. Jones, K. C. Wilhelmsen, W. Ferguson, M. McGee, M. G. Miller, “Evaluation of laser-based alignment algorithms under additive random and diffraction noise,” in *Photonic Devices and Algorithms for Computing VI*, edited by K. Iftekharuddin and A. A. S. Awwal,, Proc. of SPIE 5556 (SPIE Bellingham, WA, 2004), pp. 243-248.
6. 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.
7. A. A. S. Awwal, C. Law, J.V. Candy, W. McClay, W. Ferguson, and S. C. Burkhart, “Commissioning of the off-normal processor for NIF alignment images under varying imaging conditions,” in *Proceedings of the Sixth Tri-laboratory Engineering Conference*, p. 16, Monterey, CA, September 12-15, 2005.
8. 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.