LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-CONF-405484

Higher Accuracy Template for Corner Cube Reflected Image

A. A. S. Awwal, K. L. Rice, R. R. Leach, T. M. Taha

July 17, 2008

SPIE Optics and Photonics
San Diego, CA, United States
August 10, 2008 through August 14, 2008

Disclaimer

Higher Accuracy Template for Corner Cube Reflected Image

Abdul A. S. Awwal, Kenneth L. Rice**, Richard R. Leach and Tarek M. Taha**
National Ignition Facility
Lawrence Livermore National Laboratory, Livermore, CA 94551
***Electrical and Computer Engineering Department, Clemson University, Clemson, SC 29631*
E-mail: awwal1@llnl.gov

ABSTRACT

Video images of laser beams are analyzed to determine the position of the laser beams for alignment purpose in the National Ignition Facility (NIF). Algorithms process beam images to facilitate automated laser alignment. One such beam image, known as the corner cube reflected pinhole image, exhibits wide beam quality variations that are processed by a matched-filter-based algorithm. The challenge is to design a representative template that captures these variations while at the same time assuring accurate position determination. This paper describes the development of a new analytical template to accurately estimate the center of a beam with good image quality. The templates are constructed to exploit several key recurring features observed in the beam images. When the beam image quality is low, the algorithm chooses a template that contains fewer features. The algorithm was implemented using a Xilinx Virtex II Pro FPGA implementation that provides a speedup of about 6.4 times over a baseline 3GHz Pentium 4 processor.

Key word: pattern recognition, matched filtering, automated optical alignment.

1. INTRODUCTION

The National Ignition Facility, 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 temperatures and pressures [1]. Automatic alignment based on computer analysis of video images allows one to align the laser beams quickly and accurately enough to meet system requirements [2]. At the heart of this technique is the beam position detection algorithm, which determines the position of beam features from sensor images taken along the laser beam path. Varieties of alignment fiducials may be utilized to designate various beams, such as reference beams, main beams etc. For many of the beam images, centroiding is an acceptable technique for determining the beam position. However, other beam images may exhibit intensity variation or other distortions which makes such an approach susceptible to high position uncertainty; in these cases, matched filtering results in excellent stable position measurement [3]. Simple templates have the advantage of allowing detection over a wide range of intensity and beam quality variation. However, it may not lead to very accurate results in all cases. The objective of this work is to design a template that will yield more accurate result when the beam quality is better than could be obtained using a simple template, although at the expense of extra processing time required for template creation. The possibility of using field programmable logic array (FPGA) to speed up these computations is also explored.

2. BACKGROUND

The matched filtering technique utilizes a given object with known position as a template to find the position of a second object by detecting its position in the correlation domain. The *classical matched filter* (CMF) [4] and its variation *phase only filter* (POF) [5] are popular methods for detecting the presence of an object in the presence of noise and distortions. The *amplitude modulated phase only filter* (AMPOF) [6,7] was designed to further enhance the performance of the POF by modulating the POF by an inverse type of amplitude.

Matched filter can be elegantly described in the Fourier domain. Let the Fourier transform of the template function f (x, y) be denoted by:

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

and that of the input scene *g(x, y)* containing a replica of the template be represented by

$$G(U_x, U_y) = \left|G(U_x, U_y)\right| \exp(j\Psi(U_x, U_y)) \tag{2}$$

A CMF corresponding to this function f (x, y) which produces its autocorrelation function. is given by the complex conjugate of the input Fourier spectrum as denoted by Eq. 3.

$$H_{CMF}(U_x, U_y) = F^*(U_x, U_y) = \left|F(U_x, U_y)\right| \exp(-j\Phi(U_x, U_y)) \tag{3}$$

From the Fourier transform theory of correlation one can show that the inverse Fourier transformation of the product of F (U_x, U_y) and H_{CMF} (U_x, U_y) results in the convolution of f (x, y) and f (-x, -y), which is the equivalent of the autocorrelation of f (x, y). The cross-correlation of input image and the target is simply:

$$C_{CMF}(\Delta x, \Delta y) = F^{-1}\left\{G\left(U_x, U_y\right) H_{CMF}\left(U_x, U_y\right)\right\} \tag{4}$$

The position of the template in the input scene can be found from the position of the cross-correlation, auto-correlation, and the reference position of the template using Eqs. 5-6.

$$x_{pos} = x_{cross} - x_{auto} + x_c \tag{5}$$

$$y_{pos} = y_{cross} - y_{auto} + y_c \tag{6}$$

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 crosscorrelation peak. The position of the cross-correlation peak was estimated using a polynomial fit to the correlation peak. The center of the template, (x_c, y_c), and (x_{auto}, y_{auto}) may be calculated off-line and are normally constant, while the cross-correlation peaks move with the object position. If the template is aligned at the center of the image then the image center (x_c, y_c) cancels the auto-correlation position (x_{auto}, y_{auto}) in Eqs. (5-6). The performance of matched filter is further enhanced when the edges of the image are used instead of the image itself.

3. CORRELATION FILTER DESIGN

One of the most interesting NIF autoalignment images appears in the transport spatial filter (TSF) Pass 1 [8] as shown in Fig. 1. The automatic alignment algorithm correlates a given template with these images to determine the centers of the beams. One of the biggest challenges is to find a single template that can accurately locate the center of a set of images irrespective of the beam quality.

As can be seen from the sample images shown in Fig. 1, there are a variety of different distortions present such as non-uniformity of illumination, shape, size, and other distortions. These different factors make it harder for a single template to match perfectly under such beam conditions. It is possible that a single template may introduce deviations in the beam center estimations. By studying thousands of images from different beam lines in the NIF facility, it was concluded that the most prominent beam feature is the outer circular edge. Even when the beam exhibits a fuzzy edge in some specific segment of the beam, the other part of the beam is intact and strong. Consequently, we chose a template which could discriminate the outer edge of the beam and ignore the distortion in the rest of the beam, or even in part of the outer edge as long as more than half of the edge is intact. This gives us a template as shown in Fig. 2. The purpose of this template is to match the perimeter edge of the beams to find the center. In other words, the edge of the template is matched to the edge of the pinhole image. Since the size of the beam varies, therefore one must search over a range of radii to determine the best match circle. The position of the center of the circle that yields the highest correlation is chosen as the position of the correlated-beam image.

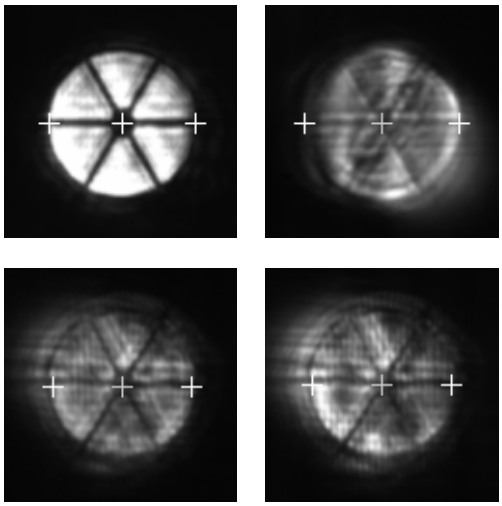

Figure 1. A sample set of corner cube reflected pinhole images

This can be problematic when the shape of the beam is significantly distorted similar to the bottom left image of Fig. 1. In this particular image a simple template will find match in two different locations, one corresponding to the edge of the actual pinhole image and one corresponding to the shadow of the image. While the crossed lines only appear at the location of the real image, they don't show at the location of the shadow. Thus if we also match the fiducial lines in addition to the edge of the circle, the algorithm may provide unambiguous match.

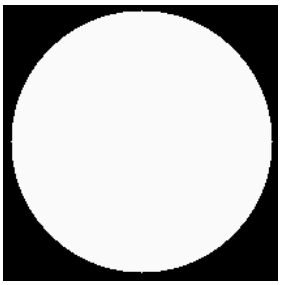

Figure 2. Original binary template

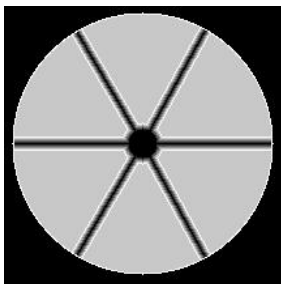

Figure 3. New gray level template

To improve the current centroid finding algorithm for pinhole images, we altered the current template (shown in Fig. 2). We enhanced the current template by adding more features as shown in sample images of Fig. 1. What we observed from the sample images is that the beams usually have a center, a horizontal line across the center, and left and right diagonals crossing at the center. From an average of the sample images, we measured that the left diagonals have an average slope of 1.689 and the right diagonals have an average slope of 1.806. All observed features have an intensity gradient. While the original template is binary, we found that the algorithm performance could be improved by constructing a template to have the above features with more than 2 gray levels. Several combinations of features were tried to determine optimal performance, including center only, horizontal line only, left diagonal line only and right diagonal only. We determined to include all the features and then tried varying the number of gray levels in the template from 2 to 7. Performance improved to a point by increasing the number of gray levels with the best performance obtained by generating the template using six gray levels.

After generating the modified template (shown in Fig. 3), we compared the performance of the original template against the modified template to assess the significance of adding template features. For the comparison between the original and modified templates, we created a set of filtered images. This image set was composed of 192 different images (two images from each of the 96 available beams as of summer 2007). We did this to get a good variance for the kinds of images that the templates would be tested against. We processed the filtered image set using the original template and the modified template. We found a 4.96% improvement over the original template when used with the filtered image set. Ground truth was determined by visual examination of the images. However the new template may give worse results with low quality beams than using the simple template. In that case, we prefer to use a simple template.

4. FPGA ACCELERATION OF IMAGE CORRELATION

The most computationally intense portion of the application is the image correlation (represented by Eqs. 1 to 4). These computations contain a significant amount of parallelism, thus enabling hardware acceleration to provide a speedup. We evaluated the potential of hardware acceleration by implementing the correlation computations on an FPGA. The test system utilized was a Cray XD1 blade consisting of 2 GHz AMD Opteron processing cores and Xilinx Virtex II Pro FPGAs. In this system an FPGA would receive the input image to process from a processor and would return the location and value of the peak in the correlation output to the processor.

4.1 Hardware design

Fig. 4 presents a system overview of the FPGA implementation. Input data and intermediate values are stored in Block RAMs (shown as the gray boxes with letters A to G). These are on-chip memories on the FPGA. The two inputs to the system, A and B, represent the source image and filter in Eqs. 1 and 2 respectively. The two-dimensional FFTs in Eqs. 1 and 2 are performed using two consecutive one-dimensional FFTs. Similarly, the inverse FFT in Eq.4 is implemented with two one-dimensional forward FFTs. The FFT units were built using Xilinx supplied library components.

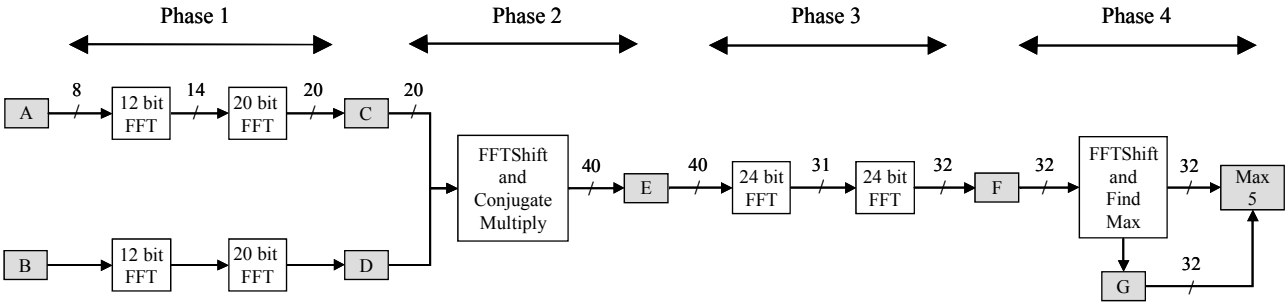

Figure 4: The block diagram of the FPGA operations.

The architecture implemented on the FPGA can be split into four phases as shown in Fig. 4:

Phase 1: Evaluate Eqs. 1 and 2. These two evaluations can be carried out in parallel. The inputs to this phase are unsigned 8-bit values. Since an 8-bit FFT unit would treat the inputs as signed values, a larger bit width FFT unit is needed. Therefore a 12-bit FFT unit is used for the first FFT stage. Following the 12-bit FFT of the first stage, a 20-bit FFT unit is used for the second stage. The outputs of the second stage FFTs utilize 20-bits and are stored in block RAM labeled C and D.

Phase 2: Eq. 3 is evaluated. Here the outputs of Eqs. 1 and 2 are multiplied. A FFT shift operation is in parallel with the multiply in order to center the image. The output at this stage is 40-bits and is stored in block RAM E.

Phase 3: Eq. 4 is evaluated. Since the inverse FFT is implemented with two 24-bit forward FFT units, the 40-bit values in E have to be scaled down by 16-bits to use the high-order 24-bits. The output at this stage is stored in block RAM F.

Phase 4: After another FFT shift operation, the location of the peak in F is determined. The intermediate peak values are stored in block RAM G. The coordinates and amplitude of the peak along with the amplitude of the surrounding four locations are returned to the processor.

The data transfer between the FPGA and its host processor is optimized. Since the same filter is used each time, it is sent once and reused for each image. Also, since the input image consists of 8-bit values, while the data bus to the FPGA is 64-bits wide, eight pixel values are packed into each data transfer.

4.2 Hardware Performance

The system above was implemented on a Xilinx Virtex II Pro FPGA (part number XCVP50) on a Cray XD1. The FPGA synthesized system ran at 159 MHz. The logic utilization was 74% while the block RAM utilization was 60%. The algorithm was also run on a 3GHz Pentium 4 processor based desktop computer using Matlab version 6. We tested both systems with a 64x64 image. The overall runtime of the FPGA system was about 0.978 ms, while the desktop system required 6.29 ms. This is a speedup of approximately 6.4 times for the FPGA implementation over the Matlab implementation.

Table 1 shows the overall output from Matlab and the FPGA for a cross-correlation between a sample beam image similar to the ones shown in Fig. 1 (top left) and the filter template shown in Fig. 3. There is a slight error in the FPGA output as it computes in the integer domain as opposed to fixed-point domain. Tables 1 show the output error between the Matlab and the FPGA implementations for the cross-correlation example. Only the peak and it neighboring four locations are shown. The average absolute error for these values 0.10%.

Table 1: Output comparison between Matlab and FPGA implementations for the peak and surrounding four locations. Absolute values for each location is shown (values are to be multiplied by 10^{10}).

Matlab	FPGA	Error (%)
8.678	8.639	0.45
8.554	8.554	0.00
8.944	8.946	-0.02
8.710	8.713	-0.02
8.247	8.245	0.03

5. SUMMARY

This paper describes the design of a template for use with corner cube reflected pinhole images of good quality. When the image quality is not so good, the simple template shown in Fig. 2 is used because when the image quality deteriorates, the line features within the pinhole image also degrade. Thus the template in Fig. 3 may match to any shifted line features such as those shown in top right corner of Fig. 1. We also demonstrate the possibility of implementing the proposed template in parallel computing hardware that resulted in speed up by more than 6 times. A more recent implementation shows a speed up of over 20 times [9] may be possible.

ACKNOWLEDGEMENT

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. Abdul Awwal acknowledges useful suggestions made by Paul Van Arsdall. Kenneth Rice acknowledges the summer student support at Lawrence Livermore National Laboratory. Kenneth Rice and Tarek Taha acknowledge grants from the Air Force Research Laboratory (including the AFRL Information Directorate) and a National Science Foundation CAREER award. This work was also supported in part by a grant of computer time from the DOD High Performance Computing Modernization Program at the Naval Research Laboratory.

REFERENCES

1. E. I. Moses, and C. R. Wuest, "The National Ignition Facility: Laser performance and first experiments." *Fusion Science and Technology* Vol. 47, p. 314-322, 2005.
2. K.C. Wilhelmsen, A. A. S. Awwal, S. W. Ferguson, B. Horowitz, V. J. Miller Kamm, C. A. Reynolds, "Automatic Alignment System for the National Ignition Facility" presented in the international Conference on Accelerator and Large Experimental Physics Control Systems, Knoxville, TN Oct 14-20, 2007.
3. 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, 2006.
4. VanderLugt, "Signal Detection by Complex Spatial Filtering," IEEE Trans. Inf. Theory IT-10, 139-145, 1964.
5. J. L. Horner and J. Leger, "Pattern recognition with binary phase-only filters," *Applied Optics*, Vol. 24, pp. 609-611, 1985.
6. A. A. S. Awwal, M. A. Karim, and S. R. Jahan, "Improved Correlation Discrimination Using an Amplitude-modulated Phase-only Filter," *Applied Optics*, Vol. 29, pp. 233-236, 1990.
7. M. A. Karim and A. A. S. Awwal, Optical Computing: An Introduction, John Wiley, New York, NY, 1992.
8. R. A. Zacharias, N. R. Beer, E. S. Bliss, et al., "Alignment of wavefront control systems of the National Ignition Facility," Optical Engineering, Vol. 43, pp. 2873-2884, 2004
9. A. A. S. Awwal, K. L. Rice and T. M. Taha, "Fast implementation of matched filter-based automatic alignment image processing," In press, Optics and Laser Technology, Vol. 40, 2008.