345-72
114015

1996

NASA / ASEE SUMMER FACULTY FELLOWSHIP PROGRAM

MARSHALL SPACE FLIGHT CENTER
THE UNIVERSITY OF ALABAMA

A REAL-TIME POSITION-LOCATING ALGORITHM FOR CCD-BASED SUNSPOT TRACKING

Prepared By:	Jaime R. Taylor, Ph.D.
Academic Rank:	Assistant Professor
Institution and Department:	Austin Peay State University Physics Department
NASA/MSFC:	
Laboratory:	Astrionics Laboratory
Division:	Instrumentation & Control
Branch:	Instrumentation
MSFC Colleague:	Ron McIntosh

1. Introduction

NASA Marshall Space Flight Center's (MSFC) EXperimental Vector Magnetograph (EXVM) polarimeter measures the sun's vector magnetic field[1]. These measurements are taken to improve understanding of the sun's magnetic field in the hopes to better predict solar flares. Part of the procedure for the EXVM requires image motion stabilization over a period of a few minutes. A high speed tracker can be used to reduce image motion produced by wind loading on the EXVM, fluctuations in the atmosphere and other vibrations. The tracker consist of two elements, an image motion detector and a control system. The image motion detector determines the image movement from one frame to the next and sends an error signal to the control system.

For the ground based application to reduce image motion due to atmospheric fluctuations requires an error determination at the rate of at least 100 hz. It would be desirable to have an error determination rate of 1 kHz to assure that higher rate image motion is reduced and to increase the control system stability.

Two algorithms are presented that are typically used for tracking. These algorithms are examined for their applicability for tracking sunspots, specifically their accuracy if only one column and one row of CCD pixels are used. To examine the accuracy of this method two techniques are used. One involves moving a sunspot image a known distance with computer software, then applying the particular algorithm to see how accurately it determines this movement. The second technique involves using a rate table to control the object motion, then applying the algorithms to see how accurately each determines the actual motion. Results from these two techniques are presented.

2. Algorithms in use for Tracking

2.1 Centroid Tracker

A centroid tracker works by calculating the centroid of each successive image then determining the shift in the centroid location from one image to the next. The centroid is calculated using

$$X_{cen} = \frac{\Sigma\Sigma\, x \cdot f(i,j)}{\Sigma\Sigma\, f(i,j)}, \qquad Y_{cen} = \frac{\Sigma\Sigma\, y \cdot f(i,j)}{\Sigma\Sigma\, f(i,j)} \tag{2}$$

where *f(i,j)* is the image intensity for a given pixel[2]. This method involves a multiply and two adds per pixel. It also involves some post processing of data to remove linear trends in the data.

2.2 Marek's Method

In reference 3 a method was described which involved comparing the current image with a light distribution model. This algorithm has been modified to use a reference image instead of a light distribution model. This method works by taking an integral of errors over the pixels of interest. This integral is

$$E_x = \Sigma\Sigma \,\mathrm{sign}\,(x_{ref} - i)\,(\,f_{ref}(i,j) - f(i,j)\,)\,, \tag{4}$$

where $f_{ref}(i,j)$ is the reference image, $f(i,j)$ is the current image, and x_{ref} is the x location of the peak intensity value of the reference image or the minimum if using the non inverted sunspot intensity profile. The location of the peak value of the current image is given by,

$$x_o = \alpha\, E_x + x_{ref}\,, \tag{5}$$

where α is an error parameter that must be determine from the initial reference image. The parameter α is determined by shifting the reference image one pixel, then applying Equation 4 with the shifted image used as $f(i,j)$, this is shown in Figure 1 with the "dotted" line representing the shifted image. Figure 2 shows the difference between the reference image and the shifted image. In this case x_o would be equal to x_{ref} + *pixel size* , and α could be determined from Equation 5.

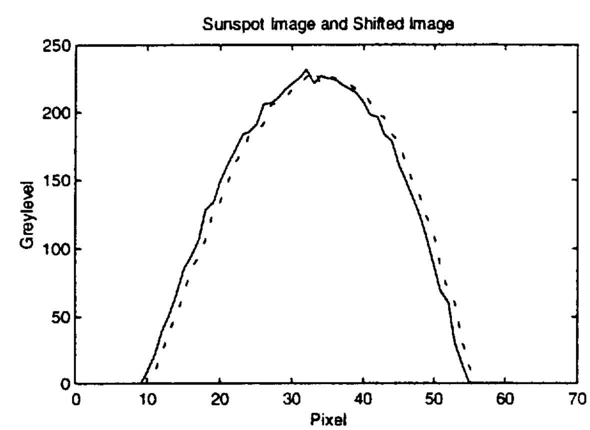

Figure 1

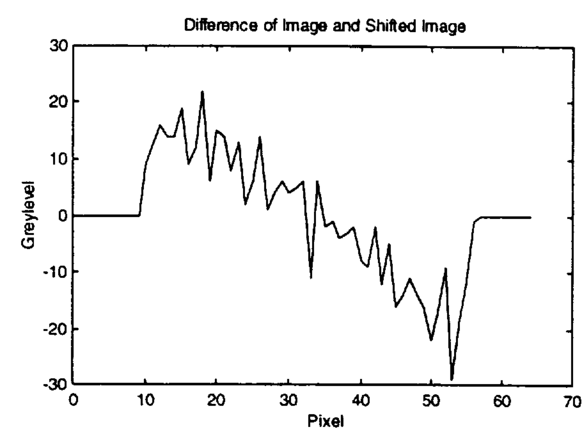

Figure 2

Marek's method requires one subtraction and one addition per pixel. This is a one addition reduction over the centroid method and it changes a multiplication to a subtraction. For high end DSP processors (for ground based applications) changing a multiplication to a subtraction does not represent a savings in time. However for a space based application a DSP might be excluded because of power, size and radiation-hardening limitations. For an 8086 Marek's method increases execution speed by nearly a factor of 20 over using the centroid algorithm[3].

A savings in time of a factor of N/2, where N is the array length, can be obtained by using one column and one row of pixels (2-axis), as opposed to using an entire NxN matrix of CCD pixels. This would save time in both the acquisition of the image data and the processing of the data. Linear CCD arrays can also be made to a higher quality. Since a sunspot has an irregular shape the question arises as to how accurate this method would be? This issue is addressed in the next two sections.

3. Algorithm Test Results with Computer Simulation

To test the algorithms an image was moved using computer software then an algorithm was applied and the calculated image motion was compared with the known image motion. The image was obtained by fitting a polynomial to a sunspot image. This polynomial was then sampled on the subpixel level and integrated over the pixel area. Once the pixel value was obtained gaussian noise was added to the image. A cross section of the sunspot image with gaussian noise added is shown in Figure 1. The image was moved randomly 100 time within a specified radius. Table 1 shows movements confined to a radius of 1 pixel, 3 pixels on up to 15 pixels. The image was divided into 65 pixels and 256 grey levels were used. The error is reported as

$$\text{error report} = |\text{ known movement - calculated movement }| \,/\, \text{pixel size} \times 100\%. \quad (6)$$

Table 1. Results of Centroid and Marek's Method using 2-axis for Tracking

	Centroid Method with 2-axis		Marek's Method with 2-axis	
Random Motion	x error %	y error %	x error %	y error %
stationary	**3.63**	3.15	**6.95**	3.94
1 pixel	**4.76**	3.22	**6.99**	4.23
3 pixel	**10.03**	5.19	**12.33**	5.76
5 pixel	**15.79**	7.33	**19.61**	7.85
10 pixel	**31.67**	12.95	**44.43**	19.97
15 pixel	**47.12**	20.13	**80.34**	59.10

If the control system works correctly the motion should be confined to a radius of less than 1 pixel. In this region both method provide similar results and it is clear that using a 2-axis tracking system would be sufficient.

4. Rate Table Experiments

A Reticon MC9128 camera and an EPIX frame grabber were used in conjunction with a rate table to uniformly rotate the Reticon camera with a simulated sunspot in the field of view (FOV). This experiment was performed twice, once for a simple compact sunspot and a second time with a more complex scene.

Figure 3 shows the results of applying Marek's method and the centroid method both using a 2-axis scheme. In Figure 3 an error pattern developed which repeated for each pixel. This pattern was produced due to a none uniform response across each individual pixel[4]. This nonlinear response produced more error than gaussian noise and is the primary limiting factor in obtaining subpixel accuracy. In Figure 3 one also notices that Marek's method tends to trail off . It eventually went to half a pixel error, see Figure 4, after moving further than four pixels, even though the reference image was update each time a pixel was crossed. It has not been determined why this occurred.

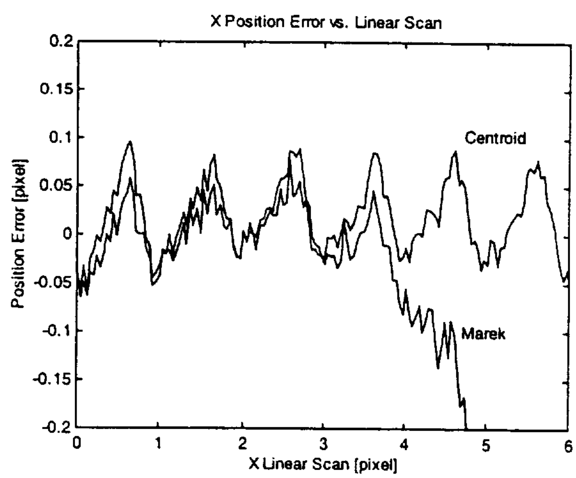

Figure 3

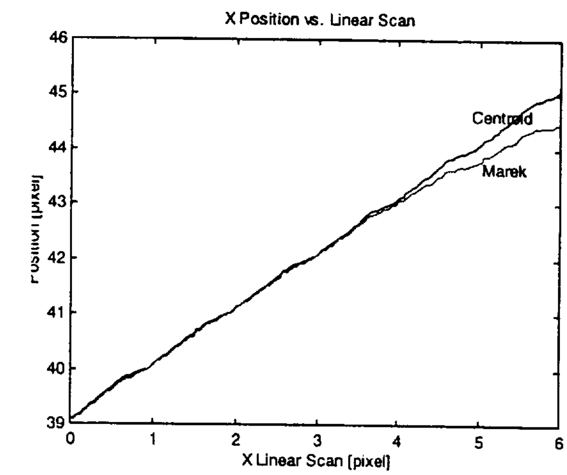

Figure 4

Next a more complex image was used to test the two algorithms. The digitized image is shown in Figure 5. The centroid algorithm was unable to track on this image. In Figure 6 the horizontal axis represents how much the image actually moved and the vertical axis is the calculated movement. Marek's method was able to track on this image, which is demonstrated in Figure 6. Figure 7 shows the error for Marek's method. Note the centroid method is not shown in Figure 7 since it would be off scale.

For a high-end DSP a subtraction, an addition, a multiply or a move between memory and register takes one clock cycle. Using the centroid method with a 64x64 array and a DSP with a clock cycle of 40ns would produce a 300 Hz rate. Using Marek's method and the 2-axis approach with two 256 linear CCD arrays would provide over a 10 kHz rate for image position error determination and increase the accuracy by nearly a factor of four over using a 64x64 array.

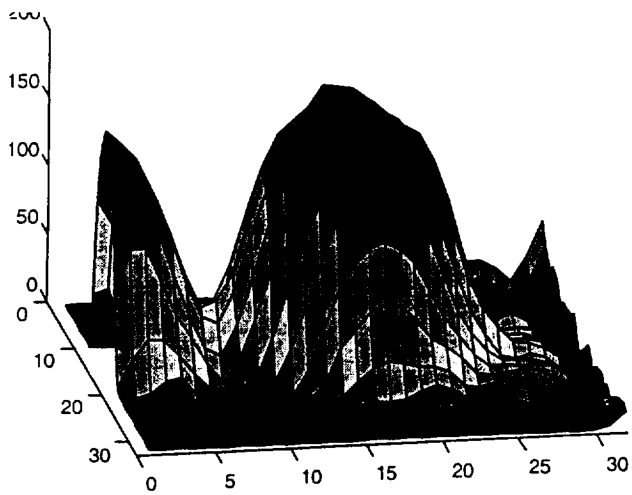

Figure 5

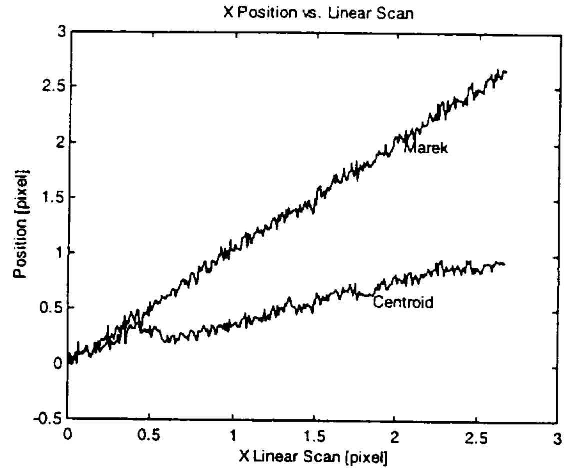

Figure 6

5. Conclusions

Using a 2-axis approach to sunspot tracking is feasible. The critical point is that the error determination and control system must operate at a rate greater than the highest frequency of image motion. Using a DSP and Marek's method would provide a position error determination system that could operate well above 1kHz, the rate of a high-end mirror based control system.

Acknowledgments

The author would like to thank Ron McIntosh, Ed West, and Don Hediger for their suggestions and support of the work presented here. He would also like to thank Marek Chmielowski for the many phone conversations they had discussing this problem. This work was supported by NASA/ASEE SFFT.

References

1 West, E. A., and Smith, M. H., "Polarization Characteristics of the MSFC Experimental Vector Magnetograph," SPIE, vol. 2265, pp. 272-283, July 1994.

2 Mansfield, P., "Machine Vision Tackles Star Tracking," Laser Focus World, pp. S21-S24, May 1996.

3 Chmielowski, M. and Klien, L., "A High-Precision, Real-Time Position-Locating Algorithm for CCD-based Sun and Star Trackers," PASP, vol. 105, pp. 114-116, January 1993.

4 Chmielowski, M., "On-Chip Image-Processing Algorithm for Real-Time CCD-Based Star Trackers and Wavefront Sensors," PASP, vol. 106, pp. 523-531, May 1994.