

Article

## A Novel Multi-Aperture Based Sun Sensor Based on a Fast Multi-Point MEANSHIFT (FMMS) Algorithm

Zheng You <sup>2</sup>, Jian Sun <sup>1,2,\*</sup>, Fei Xing <sup>2</sup> and Gao-Fei Zhang <sup>2</sup>

<sup>1</sup> MOE Key Laboratory for Strength & Vibration, School of Aerospace, Xi'an Jiaotong University, Xi'an 710049, China

<sup>2</sup> State Key Laboratory of Precision Measurement Technology and Instruments, Department of Precision Instruments and Mechanology, Tsinghua University, Beijing 100084, China; E-Mails: yz-dpi@mail.tsinghua.edu.cn (Z.Y.); xingf02@mails.tsinghua.edu.cn (F.X.); zgf@mail.tsinghua.edu.cn (G.-F.Z.)

\* Author to whom correspondence should be addressed; E-Mail: sjxa@163.com; Tel.: +086-010-62776000.

Received: 23 January 2011; in revised form: 20 February 2011 / Accepted: 28 February 2011 / Published: 3 March 2011

---

**Abstract:** With the current increased widespread interest in the development and applications of micro/nanosatellites, it was found that we needed to design a small high accuracy satellite attitude determination system, because the star trackers widely used in large satellites are large and heavy, and therefore not suitable for installation on micro/nanosatellites. A Sun sensor + magnetometer is proven to be a better alternative, but the conventional sun sensor has low accuracy, and cannot meet the requirements of the attitude determination systems of micro/nanosatellites, so the development of a small high accuracy sun sensor with high reliability is very significant. This paper presents a multi-aperture based sun sensor, which is composed of a micro-electro-mechanical system (MEMS) mask with 36 apertures and an active pixels sensor (APS) CMOS placed below the mask at a certain distance. A novel fast multi-point MEANSHIFT (FMMS) algorithm is proposed to improve the accuracy and reliability, the two key performance features, of an APS sun sensor. When the sunlight illuminates the sensor, a sun spot array image is formed on the APS detector. Then the sun angles can be derived by analyzing the aperture image location on the detector via the FMMS algorithm. With this system, the centroid accuracy of the sun image can reach 0.01 pixels, without increasing the weight and power consumption, even when some missing apertures and bad pixels appear on the detector due

to aging of the devices and operation in a harsh space environment, while the pointing accuracy of the single-aperture sun sensor using the conventional correlation algorithm is only 0.05 pixels.

**Keywords:** APS sun sensor; multi-point MEANSHIFT algorithm; multi-aperture

---

## 1. Introduction

The sun sensor is one of the most important devices for satellites. The early sun sensors were mainly sun appearance sensors and analog sun sensors [1]. With the development of space technology in recent years, requirements for sun sensors has been extended to a large field of view and high accuracy levels. The traditional design and implementation methods cannot meet these the requirements, so the single-axis digital sun sensor [2] appeared, with an accuracy of better than  $0.1^\circ$ . Two of the same single-axis digital sun sensors mounted vertically can constitute a conventional two-axis sun sensor. Such digital sun sensors are widely used currently.

Array image sensors such as CCD or CMOS and so on have been widely applied in space science. The array image sensor-based sun sensor is currently a research focus [3]. This kind of equipment is of high accuracy, high anti-interference. The aperture of sun sensors can be divided into single-aperture [as shown in Figure 1(a)] and multi-aperture [as shown in Figure 1(b)]. Compared with the traditional linear array sun sensor, the accuracy of a single-aperture sun sensor is increased, however, since only one spot is used in the system, image utilization rate is low, so it is difficult to further enhance its accuracy. The Italian Galileo Avionica Company has developed a single-aperture sun sensor, named Smart Sun Sensor [4].

Compared with the single-aperture sun sensor, a multi-aperture sun sensor has the following three advantages:

1. Accuracy: as the single-aperture sun sensor accuracy depends mainly on the positioning accuracy of the sun spot, and its pointing accuracy is only about  $0.03^\circ$ , but for the multi-aperture sun sensor the accuracy of the sun spot location centroid is proportional to the square root of the number of apertures. Assuming that the pointing accuracy of a single hole  $\delta$ , and hole arrays of  $N \times N$ , then the final pointing accuracy is  $\delta/N$ . This shows that multi-aperture can greatly improve the accuracy of the sun sensor.

2. Reliability: for the single-aperture sun sensor there is a very deadly danger: lack of reliability! When the hole is blocked or partially blocked, the accuracy will be greatly affected or even cause the system to lose function. For the multi-aperture sun sensor, the distribution of the image sensor area is relatively large, so even when part of the holes are completely blocked, using a reasonable method, the system still be able to guarantee the normal operation.

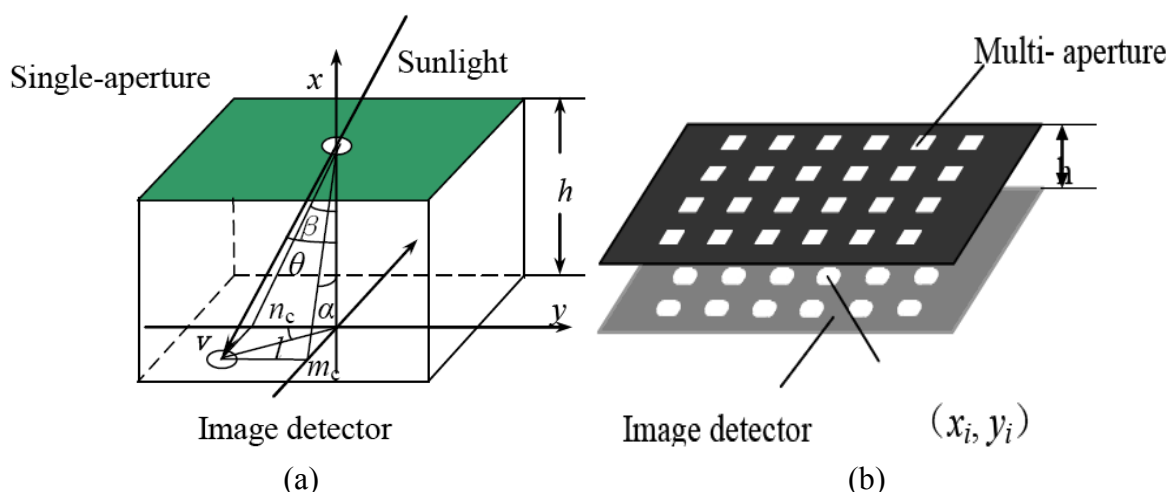
3. The image sensor requirements: the requirements of single-aperture sun sensor for image sensors are relatively high. The existence of bad pixels on the image sensor where the sun spot may fall will affect the accuracy and even function, while for the multi-aperture sun sensor, with a  $N \times N$  imaging point, the distribution of the image sensor area is relatively large, the damage to individual pixels has almost no effect on the system accuracy.

From the above analysis we can find that multi-aperture sun sensors are superior to single-aperture sun sensors as far as the accuracy, reliability, and the requirements of the image sensor are concerned, so multi-aperture sun sensors must be the future direction of the technology.

## 2. Modeling of the Sun Sensor [5]

A multi-aperture sun sensor is commonly composed of a piece of mask and a CCD or APS (Active pixel sensor) CMOS detector below at a distance of about several millimeters. The sun spot is formed at the detector through the aperture, as shown in Figure 1(a). In the Figure 1(a),  $m_c$  and  $n_c$  denote the coordinates of the sun spot in the sun sensor body coordinates,  $l$  denotes the distance between the sun spot and hole sight point,  $h$  denotes the distance between the mask and the focal plane,  $\theta$  denotes the incidence angle and  $\alpha$ ,  $\beta$  denote the sunray horizontal and azimuth orientation in the sun sensor body coordinates, respectively.

**Figure 1.** (a) single-aperture sun sensor (b) multi-aperture sun sensor.



The relationships are as follows:

$$\theta = \arctan\left(\frac{l}{h}\right), \alpha = \arctan\left(\frac{m_c}{h}\right), \beta = \arctan\left(\frac{n_c}{h}\right) \quad (1)$$

$$l = \sqrt{m_c^2 + n_c^2} \quad (2)$$

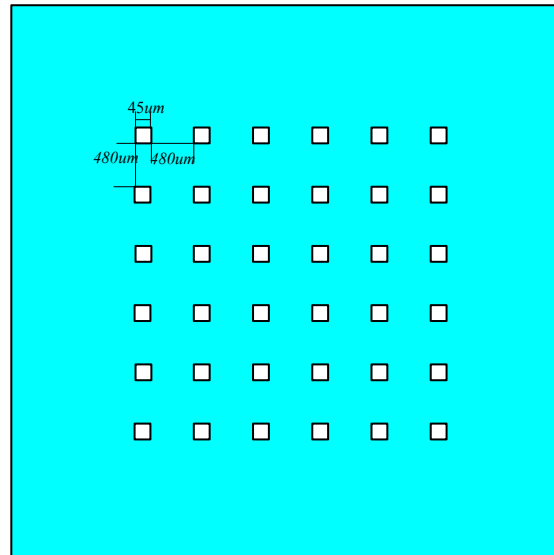
$$\tan \theta = \sqrt{(\tan \alpha)^2 + (\tan \beta)^2} \quad (3)$$

For a given distance  $h$ , the greater the number of apertures the higher the accuracy. However, the assembly and alignment errors will significantly impact the system accuracy when the location accuracy is of 1/100 pixels, about  $0.005^\circ$ . Generally, the sub-pixel accuracy of a single sun spot is at the level of 0.05–0.1 pixels, to obtain the whole sun spot location centroid accuracy of 1/100 pixels, the number of apertures should be more than 25 ( $1/100 = 0.05/251/2$ ), so the aperture pattern is designed as a  $6 \times 6$  array in this paper, as shown in Figure 2.

Assuming the sun angles are  $\alpha, \beta$  respectively and the sun spot centroid is  $m_c, n_c$  their relationship can be expressed by Equation (1). In fact, the Equation becomes much more complicated because of

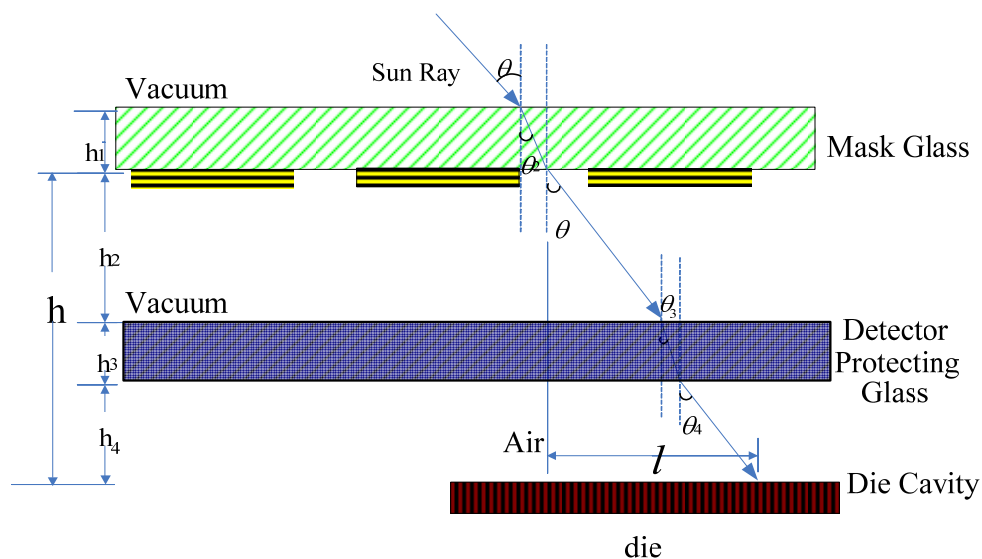
the refraction which occurs when a sun ray goes through different media sequentially (vacuum-mask glass-vacuum-detector protecting glass-air) before it arrives at the detector, as shown in Figure 3.

**Figure 2.**  $6 \times 6$  multi-aperture mask.



From Figure 3 we can conclude that the greater  $h$  the higher accuracy, however, the  $h$  must ensure that the image of sunlight in the field of view (FOV) is still within the photosensitive surface of the detector. So according to the effective detection area of the Star1000 and the sun sensor FOV design  $h$ , from the Figure 3, we can find  $h = h_2 + h_3 + h_4$ ; for the specific Star1000  $h_3$  and  $h_4$  is fixed, when the image of sunlight in the FOV is within the photosensitive surface of the detector, the greater  $h_2$  the higher accuracy. When the incident angle is large,  $h_3$  and  $h_4$  introduce nonlinear factors which may affect the sun sensor accuracy.

**Figure 3.** illustration of the sunray tracking route in the sun sensor.



From Figure 3, we can conclude that the relationship between  $l$  and  $\theta$  is:

$$l = h_2 \tan \theta + h_3 \tan \theta_3 + h_4 \tan \theta_4 \quad (4)$$

$$\frac{n_{Glass}}{n_{Vacuum}} = \frac{\sin \theta}{\sin \theta_3} \quad (5)$$

$$\frac{n_{Air}}{n_{Glass}} = \frac{\sin \theta_3}{\sin \theta_4} \quad (6)$$

Where  $n_{glass}$ ,  $n_{vacuum}$ ,  $n_{air}$  denote the light refractive index in the detector protecting glass, vacuum and air, respectively, and  $l$  is the displacement of the sun spot centroid in the case of normal incidence.

Substitution of  $\theta_3$  and  $\theta_4$  from Equations (5) and (6) into Equation (4) yields:

$$l = h_2 \tan \theta + h_3 \tan\left(\arcsin\left(\frac{\sin \theta}{n_{Glass}}\right)\right) + h_4 \tan\left(\arcsin\left(\frac{\sin \theta}{n_{Air}}\right)\right) \quad (7)$$

Since  $h_2, h_3, h_4, n_{glass}, n_{air}$  are all constants,  $l$  is a function of the incidence angle  $\theta$ . Conversely,  $\theta$  could be expressed as:

$$\theta = f(l) = f(\sqrt{m_c^2 + n_c^2}) \quad (8)$$

Combining Equations (1–3) and (8) yields:

$$\alpha = \arctan \frac{m_c \tan(f(\sqrt{m_c^2 + n_c^2}))}{\sqrt{m_c^2 + n_c^2}} \quad (9)$$

$$\beta = \arctan \frac{n_c \tan(f(\sqrt{m_c^2 + n_c^2}))}{\sqrt{m_c^2 + n_c^2}}$$

Equations (7–9) illustrate the process to calculate the sun angles ( $\alpha$ ,  $\beta$ ) from the sun spot centroid location ( $m_c$ ,  $n_c$ ) in the sun sensor body coordinates.

### 3. APS CMOS Detector Characteristics [6,15]

The STAR1000 is a CMOS image sensor with 1,024 by 1,024 pixels on a 15 mm pitch. It features on-chip Fixed Pattern Noise (FPN) correction, a programmable gain amplifier, and a 10-bit Analog-to-Digital Converter (ADC). All circuits are designed using the radiation tolerant design rules for CMOS image sensors, to allow high tolerance against to total dose effects, for more details about this one can refer to [5]. It is different from the CCD read out mechanism, for the APS based image sensors, in that registers that are directly accessed by the external controller contain the X- and Y-addresses of the pixels to be read. This architecture provides flexible operation and allows different operation modes such as (multiple) windowing, subsampling, and so on. Compared with the CCD detector, APS CMOS is more suitable as a detector for a sun sensor.

#### 4. MEANSHIFT-Based High-Precision Positioning and Tracking Algorithm

The traditional spot positioning method is the correlation algorithm [5], but this approach has two shortcomings: (1) positioning accuracy is not high, and typically only  $0.03^\circ$  accuracy is achieved (2). It cannot track the movement of the sun spot. If the sun spot location changes with the satellite attitude, the correlation algorithm cannot track the movement of the sun spot and needs to renew the location of the sun spot, which is very complicated and time consuming. In this paper we employ FMMS which not only offers high positioning accuracy but also can be adaptive to track the location of spot movement.

##### 4.1. The Spot Features Description and Similarity Function

In this paper we locate the coordinates of the centroid by identifying the spot, so we need to define the feature of the spot before identification.  $\{x_i\}_{i=1\dots n}$  is a region of the normalized data points set (which could be viewed as pixels) with 0 as the center. An isotropic, convex profile with a monotonically decreasing kernel function  $k(x)$  weighted for all points, farther away from the center, the smaller the weight.

According to the MEANSHIFT [7] algorithm, calculating the spot features is divided into the following three steps:

1. Characterization of the pixel value (Figure 4). Defined function  $b: R^2 \rightarrow \{1\dots m\}$ , for each pixel,  $0 \leq x \leq 255$ ,  $b(x)$  is the quantified number of the pixels value in the quantitative feature space:

$$b(x) = \begin{cases} 1 & x < t_1 \\ 2 & t_1 \leq x < t_2 \\ \vdots & \vdots \\ m & t_n \leq x \end{cases} \quad (10)$$

2. Select kernel function (Figures 5 and 6). There are many kinds of kernel function mentioned in [7], such as Epanechnikov kernel:

$$K_E(x) = \begin{cases} C_k(1 - \|x\|^2) & \|x\| < 1 \\ 0 & \|x\| \geq 1 \end{cases} \quad (11)$$

normal kernel:

$$K_N(x) = (2\pi)^{-d/2} \exp(-\frac{1}{2} \|x\|^2) \quad (12)$$

In this paper we employed the Epanechnikov kernel. For the definition of the kernel function one can refer to [7], as no details are provided here.

3. Using the kernel function weighted all the quantified pixel, Calculated the spot feature vector as follows:

$$\hat{q}_u = C \sum_{i=1}^n k_E(\|x_i\|^2) \delta[b(x_i) - u], \quad u = 1 \dots m$$

In which constant  $C$  is a normalization function, makes the  $\sum_{u=1}^m \hat{q}_u = 1$ , then  $C = \frac{1}{\sum_{i=1}^n k_E(\|x_i^*\|^2)}$ .

For a simulated sun spot such as Figure 7. The quantified spot template is shown as Figure 5. The kernel function weight distribution is shown as Figure 6 and a 3-dimensional map of the kernel function is shown as Figure 7. The feature vector is:

$$q_1 = 0.08999 \quad q_2 = 0.28798 \quad q_3 = 0.23038 \quad q_4 = 0.14399 \quad q_5 = 0.21886 \quad q_6 = 0.02880$$

$$\hat{q} = [0.08999, 0.28798, 0.23038, 0.14399, 0.21886, 0.02880]$$

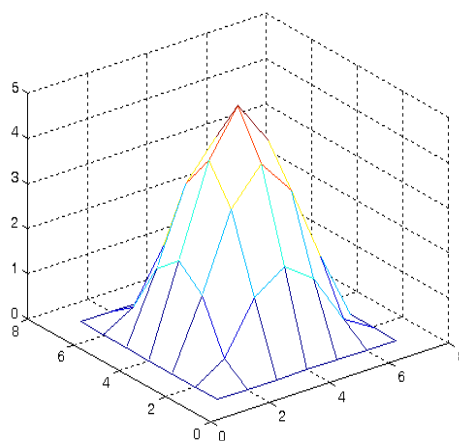
**Figure 4.** The quantified spot template.

1	6	6	6	6	6	6	6
2	6	6	5	4	5	6	6
3	6	5	3	2	3	5	6
4	6	4	2	1	2	4	6
5	6	5	3	2	3	5	6
6	6	6	5	4	5	6	6
7	6	6	6	6	6	6	6
	1	2	3	4	5	6	7

**Figure 5.** Kernel function weight distribution.

1	0	0	0	0	0	0	0
2	0	0.4	1.52	2	1.52	0.4	0
3	0	1.52	3.2	4	3.2	1.52	0
4	0	2	4	5	4	2	0
5	0	1.52	3.2	4	3.2	1.52	0
6	0	0.4	1.52	2	1.52	0.4	0
7	0	0	0	0	0	0	0
	1	2	3	4	5	6	7

**Figure 6.** 3-dimensional map of kernel function.



**Figure 7.** The simulated spot gray distribution.

1	0	0	0	0	0	0	0
2	0	30	70	120	70	30	0
3	0	70	150	200	150	70	0
4	0	120	200	225	200	120	0
5	0	70	150	200	150	70	0
6	0	30	70	120	70	30	0
7	0	0	0	0	0	0	0
	1	2	3	4	5	6	7

We set the candidate target model  $\{x_i\}_{i=1 \dots n_h}$ , with  $y$  as the center. Using the same kernel profile, the radius  $h$ , features  $u = 1 \dots m$  probability of occurrence in the candidate target model is:

$$\hat{p}_u(y) = C_h \sum_{i=1}^{n_h} k_E \left( \left\| \frac{y - x_i}{h} \right\|^2 \right) \delta[b(x_i) - u] \quad (13)$$

If  $C_h$  is a normalization factor, making  $\sum_{u=1}^m \hat{p}_u = 1$ , then:

$$C_h = \frac{1}{\sum_{i=1}^n k \left( \left\| \frac{y - x_i}{h} \right\|^2 \right)}$$

The term  $h$  determines the scale of the candidate target.  $\hat{q}_u$  and  $\hat{p}_u(y)$  is the description of the feature of the goal and the candidate. Using the Bhattacharyya coefficient:

$$\hat{\rho}(y) \equiv \rho[\hat{p}(y), \hat{q}] = \sum_{u=1}^m \sqrt{\hat{p}_u(y) \hat{q}_u} \quad (14)$$

we describe the degree of similarity between the feature of the goal and the candidate; the greater the coefficient, the more similarity between the goal and the candidate. The distance between them is defined as follows:

$$d(y) = \sqrt{1 - \rho[\hat{p}(y), \hat{q}]} \quad (15)$$

#### 4.2. The Sun Spot Pointing and Tracking

With a description of the goal and the candidate target, and the criteria which can gauge their similarity degree, pointing and tracking the spot becomes a problem of searching for a new spot location in the current frame which made it possible that the function with  $y$  as independent variables could obtain the minimum distance. Searching begins from the previous location of a spot, and looking around in the neighborhood. Minimizing Equation (15) is equivalent to maximizing Equation (14). Assuming  $y_1$  is the spot location of the previous frame, we first calculate  $\{\hat{p}_u(y_1)\}_{u=1 \dots m}$  in the current

frame, if we expand the  $\rho[\hat{p}(y), \hat{q}]$  in the predicted location of  $y_1$ , we then get the linear approximation of  $\rho[\hat{p}(y), \hat{q}]$ :

$$\rho[\hat{p}(y), \hat{q}] \approx \frac{1}{2} \sum_{u=1}^m \sqrt{\hat{p}_u(\hat{y}_1) \hat{q}_u} + \frac{1}{2} \sum_{u=1}^m \hat{p}_u(y) \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{y}_1)}} \quad (16)$$

This kind of approximation is satisfactory when there is little difference between the candidate  $\{\hat{p}_u(y)\}_{u=1, \dots, m}$  and the initial  $\{\hat{p}_u(\hat{y}_1)\}_{u=1, \dots, m}$ . In general, this assumption for the adjacent two frames is reasonable.

Combining Equation (13) with (16) yields:

$$\rho[\hat{p}(y), \hat{q}] \approx \frac{1}{2} \sum_{u=1}^m \sqrt{\hat{p}_u(\hat{y}_1) \hat{q}_u} + \frac{C_h}{2} \sum_{i=1}^{n_h} w_i k'(\|\frac{y - x_i}{h}\|^2) \quad (17)$$

in which:

$$w_i = \sum_{u=1}^m \delta[b(x_i) - u] \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{y}_1)}} \quad (18)$$

In this way, minimizing  $d(y)$  becomes to maximize the second half of Equation (17), which denotes the kernel density estimation calculated by using  $k(x)$  at the  $y_i$  in current frame, so that we can use the MEANSHIFT procedure to find the great density estimation value (mode) in the neighborhood. In this process, the kernel shift from the current location  $y$  to the new location  $y_1$ :

$$y_1 = \frac{\sum_{i=1}^{n_h} x_i w_i g(\|\frac{y - x_i}{h}\|^2)}{\sum_{i=1}^{n_h} w_i g(\|\frac{y - x_i}{h}\|^2)} \quad (19)$$

in which  $g(x) = -k'(x)$

If we set the spot with a feature vector  $\{q_u\}_{u=1, \dots, m}$  at  $y_0$  in the previous frame,  $y_1$  is the new spot location. Then the flow of the MEANSHIFT algorithm is:

Set the spot with a feature vector  $\{q_u\}_{u=1, \dots, m}$ , at  $y_0$  in the previous frame.

1. Calculate the feature vector of candidate spot  $\{\hat{p}_u(\hat{y}_0)\}_{u=1, \dots, m}$  of the first spot in current frame, and Bhattacharyya coefficient:

$$\rho[\hat{p}(\hat{y}_0), \hat{q}] = \sum_{u=1}^m \sqrt{\hat{p}_u(\hat{y}_0) \hat{q}_u}$$

2. Calculate  $\{w_i\}_{i=1, \dots, m}$  with Equation (18).

$$w_i = \sum_{u=1}^m \delta[b(x_i) - u] \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{y}_1)}}$$

3. Calculate the new location of spot with Equation (19)

$$\hat{y}_1 = \frac{\sum_{i=1}^{n_h} x_i w_i g(\|\frac{\hat{y} - x_i}{h}\|^2)}{\sum_{i=1}^{n_h} w_i g(\|\frac{\hat{y} - x_i}{h}\|^2)}$$

4.  $\rho[\hat{p}(\hat{y}_1), \hat{q}] = \sum_{u=1}^m \sqrt{\hat{p}_u(y_1) \hat{q}_u}$ .
5. If  $\rho[\hat{p}(\hat{y}_1), \hat{q}] < \rho[\hat{p}(\hat{y}_0), \hat{q}]$ , then  $\hat{y}_1 \leftarrow \frac{1}{2}(\hat{y}_0 + \hat{y}_1)$  until  $\rho[\hat{p}(\hat{y}_1), \hat{q}] > \rho[\hat{p}(\hat{y}_0), \hat{q}]$
6. If  $\|\hat{y}_1 - \hat{y}_0\| < \varepsilon$  then stop iteration to get the spot new location.  
else  $\hat{y}_0 \leftarrow \hat{y}_1$  jump to 2
7. Repeat steps 1–6 to get the coordinates of other sun spots.
8. Calculate the sun incident angle  $(\alpha_i, \beta_i)_{i=1 \dots 36}$ .
9. Then the final sun incident angle  $(\alpha, \beta)$  are

$$\alpha = \frac{\sum_{i=1}^n \alpha_i}{n}, \beta = \frac{\sum_{i=1}^n \beta_i}{n}$$

#### 4.3. Over-Relax Technology Based Multi-Point Fast MEAN Shift Algorithm

For a single spot using MEANSHIFT algorithm positioning usually reaches sub-pixel accuracy, and in order to achieve high-precision positioning we need to use spot arrays. For each spot in the array we can get a pair of sun incident angles  $\alpha_i, \beta_i$ , then the final sun incident angle is:

$$\alpha = \frac{\sum_{i=1}^n \alpha_i}{n}, \beta = \frac{\sum_{i=1}^n \beta_i}{n} \quad (20)$$

in which  $n$  is the number of the aperture.

As the MEANSHIFT computing process is complex, a multiple point MEANSHIFT procedure requires a lot of time, so in order to improve the real-time system, we need to study the fast MEANSHIFT algorithm. In this paper we use the over-relaxing method [8] to adjust the step size, and acceleration can be achieved. The over-relaxed bound optimization iteration is:

$$y^{(k+1)} = y^{(k)} + \beta \cdot T_{step} \quad (21)$$

In which  $T_{step}$  is the MEANSHIFT vector. Apparently when the learning rate  $\beta = 1$ , over-relaxed optimization reduces to the standard MEANSHIFT algorithm. It is easily seen that when  $\beta > 1$  acceleration is realized. However by simply assigning a fixed value to  $\beta$ , no convergence is secured and it seems quite difficult to get the optimal value for  $\beta$ . Xu *et al.* [9] proves that in the case of the Gaussian Mixture Model (GMM) parameter estimation, convergence can be guaranteed using this method when we are close to a local maximum and  $0 < \beta < 2$ . This conclusion is generalized to the case of general bound optimization methods in [10,11]. Based on this important proposition, a simple adaptive over-relaxed bound optimization is readily available: the learning rate can be adjusted by evaluating the cost function. When the cost function becomes worse for some  $\beta > 1$ , then  $\beta$  has been set too large and needs to be reduced. By simply setting  $\beta = 1$  immediately, convergence can be achieved. In this paper by regarding MEANSHIFT as a special case of bound optimization we presented the accelerated MEANSHIFT algorithm obtained in this way:

##### 1. Initialization:

Set the iteration index  $k = 1$ , and the step parameter  $\alpha > 1$ .

2. Iterate until convergence condition is met:

(1) Calculate  $\hat{y}_{i+1}$  with Equation (19). And the MEANSHIFT vector

$$m_G(\hat{y}_{i+1}) = \hat{y}_{i+1} - \hat{y}_i.$$

(2)  $y_{i+1} = y_i + \beta m_G(\hat{y}_{i+1})$

(3) If  $\rho(y_{i+1}) > \rho(y_i)$ ,

Accept  $y_{i+1}$  and  $\beta = 1$ ,  $\beta = \alpha \cdot \beta$

Else reject  $y_{i+1}$ , and  $y_{i+1} = \hat{y}_{i+1}$ ,  $\beta = 1$ .

(4) Set  $k = k + 1$ , start a new iteration

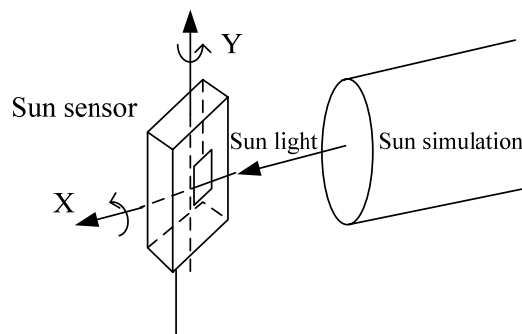
(5) If  $m_G(y_{i+1}) < \mathcal{E}$  stop iteration.

## 5. The Test and Experiment

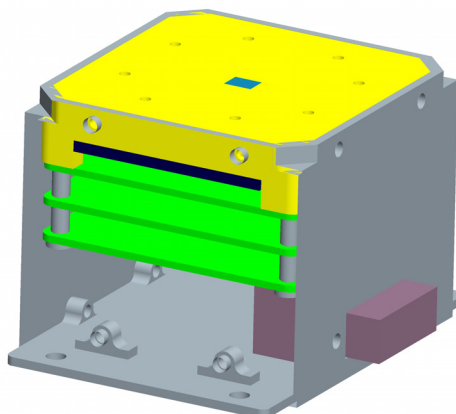
### 5.1. The Lab Test Equipment for an APS Sun Sensor

The equipment used to test and calibrate the sun sensor consists of a sun simulator which can send a bundle of sunlight, a two-axis gimbals rotary table and a theodolite goniometer. The sun simulator has an angular diameter of 32 arcmin and a brightness of 0.1 sun constant. The sun sensor was fixed on the two-axis rotary table. The theodolite goniometer guarantees that the axis of sun sensor is collinear with the sun simulator's. Arbitrary sun incidence angles can be established by rotating the table. The lab test system is shown in Figure 8. The diagram of an APS sun sensor is shown in Figure 9.

**Figure 8.** The lab test system for a sun sensor.

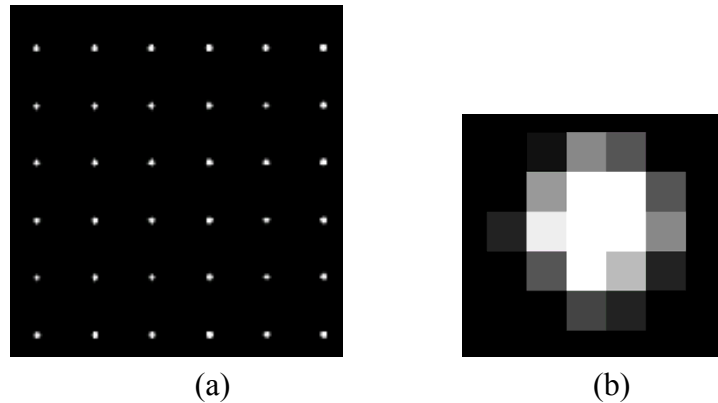


**Figure 9.** The diagram of an APS sun sensor.



The sun spot templates extracted from the sun image when the sun incident angles are both equal to 0. This work is carried out on the lab test equipment. The sun spots array image and a typical single sun spot template are shown in Figure 10(a,b). For each sun spot, we can get a template with  $5 \times 5$  pixels.

**Figure 10. (a)** Sun spots array image. **(b)** Single sun spot image.



### 5.2. Random Noise [12]

The sun sensor was placed on the two-axis turntable for about 1,000 seconds. A single aperture in the mask was chosen from the image with coordinates  $(x,y)$ . The  $x$  and  $y$  values of the centroid is a function of time. The standard deviation on the  $x$  coordinate is 0.02 pixels and the standard deviation on the  $y$  coordinate is 0.02 pixels. Assuming a stable sun bundle, the coordinates of this aperture are supposed to be constant. The actual distance between the mask and detector is  $4,000 \mu\text{m}$ , the size of APS CMOS pixel is  $15 \mu\text{m} \times 15 \mu\text{m}$ . One pixel therefore has an angle equivalent to  $\arctan(15/4,000) = 0.21^\circ$ . The random noise on a single aperture is then  $0.02 \times 0.21^\circ = 0.0042^\circ$  in each axis. This random noise is for a single aperture. Assuming that the measurements are independent, and the mask consists of  $6 \times 6 = 36$  apertures, then the accuracy will be improved by  $\sqrt{36} = \text{six}$  times. A noise estimate for this kind of sun sensor is then  $0.0042^\circ / 6 = 0.0007^\circ$ .

### 5.3. Experimental Parameter Setting

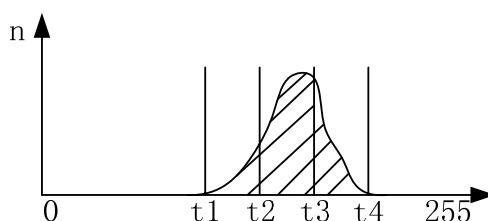
The choice of kernel function for the experimental results has a significant impact; this paper uses the following kernel function profile:

$$K_E(x) = \begin{cases} C_k(1 - \|\frac{x}{z}\|^2) & \|\frac{x}{z}\| < 1 \\ 0 & \|\frac{x}{z}\| \geq 1 \end{cases} \quad (19)$$

where  $z$  denotes effective distance which is decided by the size of the sun spot. The term  $x$  actually represents the distance between the effective pixels and the center spot. If  $z$  is set too large or too small it will affect the pointing and tracking performance. If  $z$  is set too large it increases the computation and affect the system's real-time; when  $z$  is set too small, making the effective pointing and tracking of the region smaller, not only is operational efficiency reduced, but it is difficult to ensure positioning accuracy. In this paper  $z$  is set at 6.

The election of the gray value of the pixel as a feature in general gray images; in this paper the gray value will be classified as follows (Figure 11): we choose four threshold values  $t_1, t_2, t_3, t_4$ . Five features are set:  $q_1 = [0, t_1)$ ,  $q_2 = [t_1, t_2)$ ,  $q_3 = [t_2, t_3)$ ,  $q_4 = [t_3, t_4)$ ,  $q_5 = [t_4, 255)$ , in which  $t_1, t_2, t_3, t_4$  were set as 60, 110, 160, 210, respectively.

**Figure 11.** Choice of the feature value.



According to the general considerations, the number of holes was designed to be 36 in this paper which is explained in Section 2, so that the the point accuracy of the sun spot array image location could reach a 0.01 ~ 0.02 pixel level.

#### 5.4. Calibration and Analysis

##### 5.4.1. Calibration

First of all, the sun sensor was fixed on the two-axis turntable, ensuring that the optical axis of the sun sensor was parallel with the sun simulator's, controlling the turntable rotation around the x axis in steps of  $1^\circ$  from  $-64^\circ$  to  $+64^\circ$ , while keeping the y axis unrotated, then the turntable angle and the sun sensor angle can be recorded:

Turntable x axis angle:  $X[x_0 \ x_1 \cdots x_{128}]$

Sun sensor angle:  $\alpha[\alpha_0 \ \alpha_1 \cdots \alpha_{128}]$

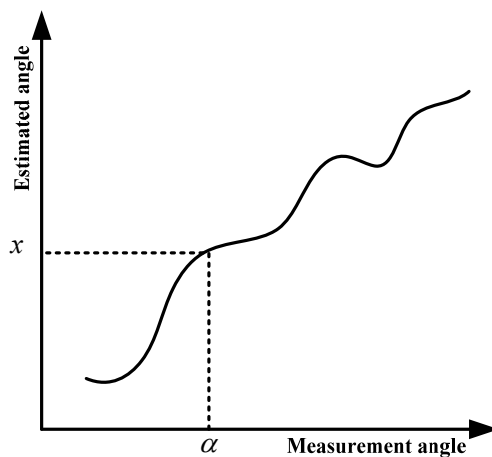
Then we control the turntable rotation around the y axis and keep the x axis unrotated to get the turntable y axis angle  $Y[y_0 \ y_1 \cdots y_{128}]$  and sun sensor angle  $\beta[\beta_0 \ \beta_1 \cdots \beta_{128}]$ .

Using MATLAB fitplot function  $(x, \alpha, 6)$ , the fitplot function  $(y, \beta, 6)$  is used to fit the two sets of data:

$$\begin{aligned}
 x_0 &= A_0\alpha_0^6 + A_1\alpha_0^5 + A_2\alpha_0^4 + A_3\alpha_0^3 + A_4\alpha_0^2 + A_5\alpha_0^1 + A_6 \\
 x_1 &= A_0\alpha_1^6 + A_1\alpha_1^5 + A_2\alpha_1^4 + A_3\alpha_1^3 + A_4\alpha_1^2 + A_5\alpha_1^1 + A_6 \\
 &\vdots \\
 x_{128} &= A_0\alpha_{128}^6 + A_1\alpha_{128}^5 + A_2\alpha_{128}^4 + A_3\alpha_{128}^3 + A_4\alpha_{128}^2 + A_5\alpha_{128}^1 + A_6 \\
 y_0 &= B_0\beta_0^6 + B_1\beta_0^5 + B_2\beta_0^4 + B_3\beta_0^3 + B_4\beta_0^2 + B_5\beta_0^1 + B_6 \\
 y_1 &= B_0\beta_1^6 + B_1\beta_1^5 + B_2\beta_1^4 + B_3\beta_1^3 + B_4\beta_1^2 + B_5\beta_1^1 + B_6 \\
 &\vdots \\
 y_{128} &= B_0\beta_{128}^6 + B_1\beta_{128}^5 + B_2\beta_{128}^4 + B_3\beta_{128}^3 + B_4\beta_{128}^2 + B_5\beta_{128}^1 + B_6
 \end{aligned}$$

to get the two vectors:  $A(A_0, A_1 \cdots A_6)$ ,  $B(B_0, B_1 \cdots B_6)$ , then we can get the relationship between the measurement angle and the estimated angle (Figure 12).

**Figure 12.** Relationship between the measurement angle and estimated angle.



A high performance Fixed-Point Digital Signal Processor TMS320C6203 which was developed by the Texas Instruments Company was employed in the sun sensor for signal processing. The data update rate is 4 Hz. The final sun sensor accuracy is: X axis:  $0.0063^\circ$ , Y axis:  $0.0068^\circ$  (Figure 13).

**Figure 13.** The accuracy of a sun sensor with incident angle in the range of  $-64^\circ$  to  $+64^\circ$ .  
(a) x axis result, (b) y axis result.

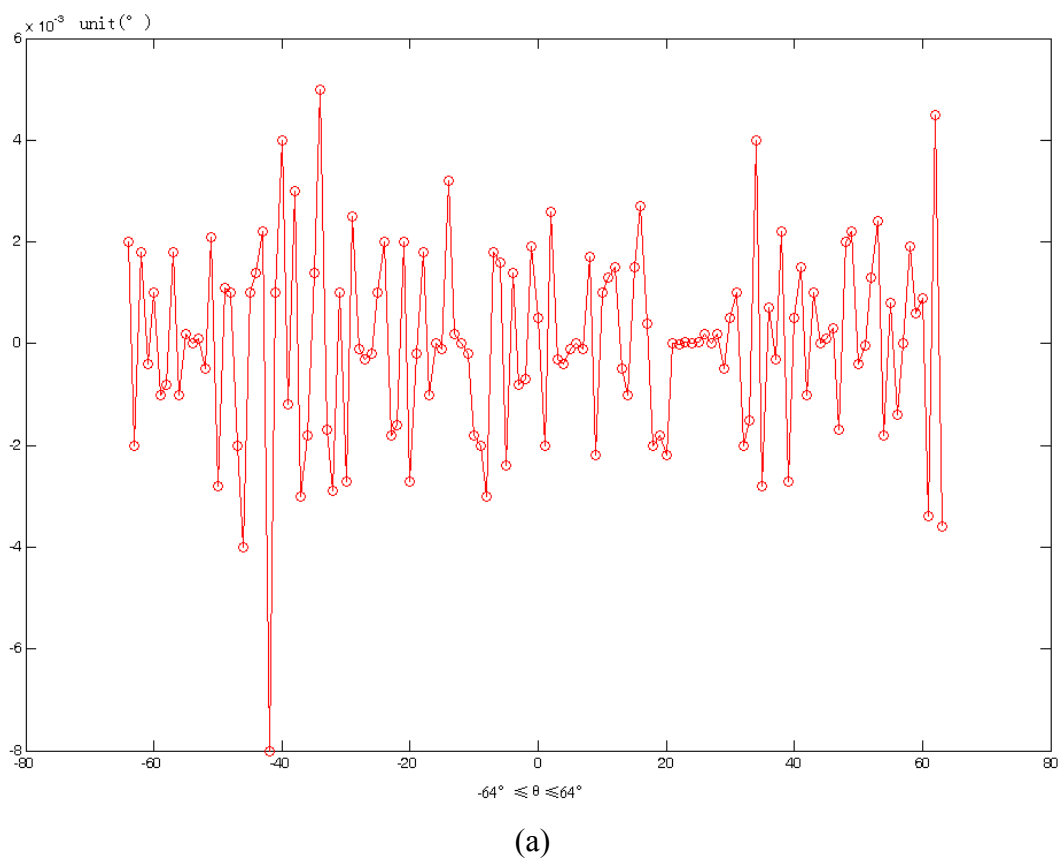
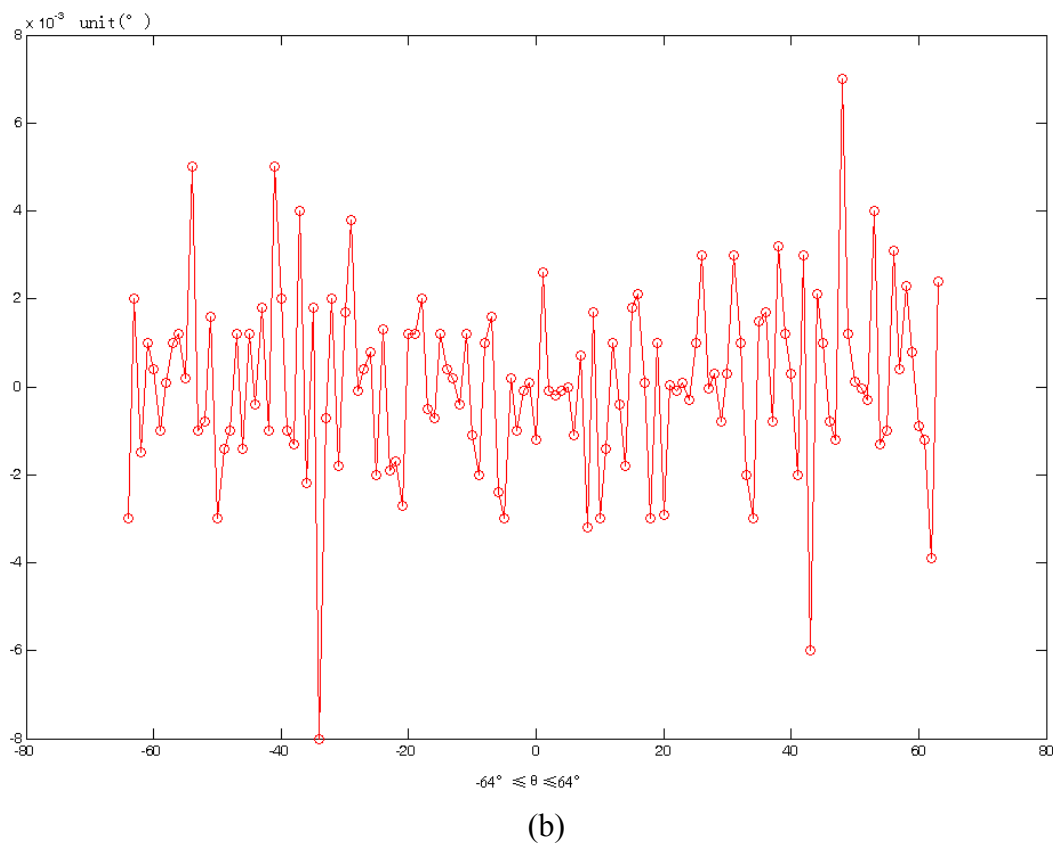


Figure 13. Cont.

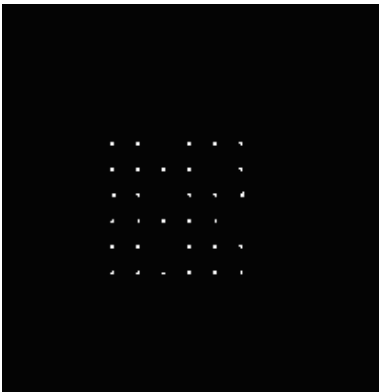


#### 5.4.2. The Hole Block Detection Logic and Its Impact on the Accuracy

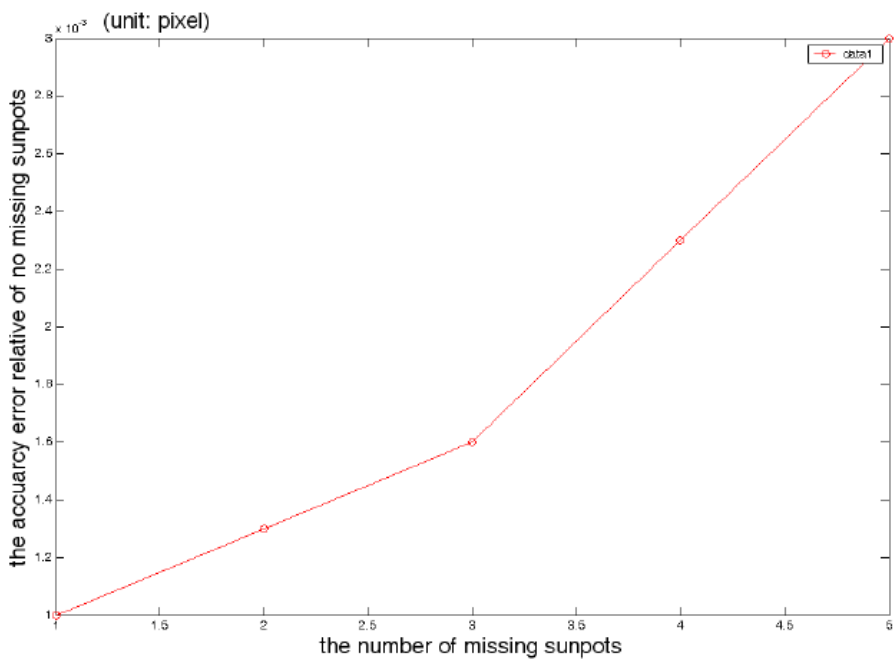
In space, some sun spots may be missing due to space debris, mask pollution or detector aging, *etc.* The multi-aperture sun sensor based on the FMMS algorithm has the advantage of maintaining a high accuracy even if some sunspots are missing. As the distance between the spots is relatively fixed, it can predict the location of holes in a certain effective area. When a similarity coefficient is less than a certain threshold in MEANSHIFT algorithm iteration process, the hole blocking. This method not only has high accuracy and the computation load is small, but it also does not affect system real-time performance.

A typical sun spot array image with five spots missing is shown in Figure 14. An experiment is carried out to compare the accuracy deterioration under the conditions that one-five sun spots are missing relative to that of no sunspot missed. The experimental data are shown in Figure 15. From Figure 15, we concluded that the average accuracy error is no more than 0.003 pixels, which has almost no impact on the system accuracy. This immunity from missing sun spots significantly improves the adaptability for a harsh space environment, and this immunity cannot be achieved by a conventional single-aperture sun sensor. The conventional single-aperture sun sensor cannot distinguish between the sun spot and noise or bad pixels, while a multi-aperture sun sensor can overcome this drawback and extend the lifetime, since both the FMMS algorithms extract and identify the sun image based on the apertures array pattern, and the immunity to the partial sun spots missing or deterioration keeps the sun sensor away from a fatal fault. Figure 16 shows a simulated deteriorated image.

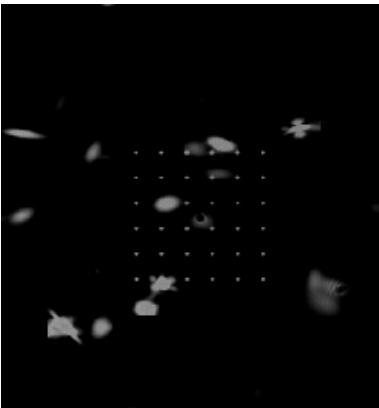
**Figure 14.** The image with one–five missing sun spots.



**Figure 15.** Accuracy analyses with 1–5 sun spots missing.



**Figure 16.** The simulated deteriorated image.



## 6. Conclusions and Future Work

In this paper, based on multi-aperture, APS detector flexible imaging and read out characters and a novel fast multi-point MEANSHIFT (FMMS) algorithm are proposed to improve the accuracy and reliability, the two key performance features of an APS sun sensor. The experimental results show that based on FMMS algorithm the multi-aperture the APS sun sensor presents many advantages compared with the traditional single-aperture sun sensor, particularly in high accuracy and reliability. With this system, the centroid accuracy of the sun image can reach to 0.01 pixels, even when some missing apertures and bad pixels appear on the detector because of aging of the devices and a harsh space environment, while the pointing accuracy of the single-aperture sun sensor using the conventional correlation algorithm is only 0.05 pixels.

When the sun incidence angle is greater than  $45^\circ$ , the sun spot is larger, the shape becomes elliptical, and sun sensor accuracy declined. In the next phase of our work we will study the reasons for this phenomenon and to find solutions to improve accuracy under wide sun angle conditions.

## Acknowledgements

This paper funded by the National Natural Science Foundation (No. 60904089). This research work was mainly carried out in the State Key Laboratory of Precision Instrument Measurement, Tsinghua University, while some of the analysis and simulation was completed at the School of Aeronautics & Aerospace Xi'an Jiaotong University. The authors wish to acknowledge the important contributions made by Yu Li, Wei Li, Dong Xu and Junshen Mo to our APS sun sensor development and Qi Wang for his efforts regarding writing and amending the paper.

## References

1. Milton, C.P.; Jenifer L.K. Using the Sun Analog Sensor (SAS) data to investigate solar array yoke motion on the GOES-8 and -9 spacecraft. *Proc. SPIE* **1996**, *2812*, 753-763.
2. Chen, F.F.; Feng, J.; Hong, Z.W. Digital sun sensor based on the optical vernier measuring principle. *Meas. Sci. Technol.* **2006**, *172*, 2494-2498.
3. Hales, J.H.; Pedersen, M. Two-Axis MOEMS sun sensor for pico-satellites. In *Proceedings of 16th Annual AIAA/USU Conference on Small Satellites*, Logan, UT, USA, August 12–15, 2004.
4. Liebe, C.C.; Mobasser, S.; Bee, Y.; Wrigley, C.J.; Schroeder, J.R.; Howard, A.M. Micro sun sensor. In *Proceedings of IEEE Aerospace Conference*, Big Sky, MT, USA, March 9–16, 2002.
5. Fei, X.; Zheng, Y.; Gao, F.Z. Jian, S. A novel active pixels sensor(APS) based Sun sensor basedon a feature extraction and image correlation(FEIC) technique *Meas. Sci. Technol.* **2008**, *19*, 125203.
6. STAR1000 datasheet. Available online: <http://www.fillfactory.com> (accessed on 20 December 2010).
7. Comaniciu, D.; Meer, P. MEANSHIFT: A robust approach toward feature space analysis. *IEEE Trans. Patt. Anal. Mach. Intell.* **2002**, *24*, 603-619.

8. Shen, C.; Brooks, M.J.; Hengel, A. van den. Fast Global kernel density mode seeking with application to localization and tracking. In *Proceedings of the 10th IEEE International Conference on Computer Vision*, Beijing, China, October 17–21, 2005.
9. Xu, L.; Jordan, M.I. On convergence properties of the EM algorithm for Gaussian mixtures, *Neural Comput.* **1996**, *8*, 129-151.
10. Ortiz, L.E.; Kaelbling, L.P. Accelerating EM: An empirical study. In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence*, Stockholm, Sweden, July 30–August 1, 1999; pp. 512-521..
11. Fashing, M.; Tomasi, C. Mean shift is a bound optimization. *IEEE Trans. Patt. Anal. Mach. Intell.* **2005**, *27*, 471-474.
12. Liebe, C.; Mobasser, S. MEMS based Sun sensor. Aerospace Conference. In *Proceedings of 2001 IEEE Aerospace Conference*, Big Sky, MT, USA, March 10–17, 2001.

© 2011 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).