JOURNAL OF SPACECRAFT AND ROCKETS
Vol. 45, No. 2, March–April 2008

Parametric Processing for Two-Dimensional Digital Sun Sensors: Algorithms, Modeling, and Testing

John Enright,[*] Albert Yam,[†] and Chris Li[†]
Ryerson University, Toronto, Ontario M5B 2K3 Canada

DOI: 10.2514/1.32537

This paper presents the modeling, algorithm development, and laboratory testing of a two-axis digital sun sensor. The basis for our modeling is the Sinclair Interplanetary SS-330, a sensor that employs a single linear array to provide two-axis measurements of the sun vector. The detector illumination simulation includes consideration of Fresnel diffraction, finite pixel size, the extended nature of the sun, and a detector-noise model to synthesize artificial images. Simulated images agree with actual sensor images to within about 3.2% (root mean square). Algorithm studies contrast the performance of existing peak-detection routines (e.g., peak detections and centroiding) with more sophisticated parametric algorithms. Laboratory studies confirm that these modified algorithms can provide a threefold performance improvement over the baseline sensor processing. These results also evaluate the simulation's ability to provide accurate estimates of sensor performance. Although the simulations qualitatively replicate many behaviors, they are not yet accurate enough to be used in place of hardware validation.

Nomenclature

$\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D}$	=	modeled image components
a_1, a_2, b_1, b_2	=	modeled image component amplitude
$C_f(\alpha), S_f(\alpha)$	=	Fresnel cosine/sine integrals
D_x, D_y	=	displacement of slits along the x–y axis, m
$\mathrm{d}\xi$	=	width of the sun-disk chord segment, rad
$\mathbf{d\lambda}$	=	update to estimated parameters
$\boldsymbol{E}, \boldsymbol{F}, \boldsymbol{G}, \boldsymbol{H}$	=	slope of modeled image components
$\boldsymbol{H}_\lambda$	=	nonlinear least-squares measurement matrix
h	=	thickness of optics, m
I	=	sensor image
I_c	=	continuous single-slit illumination pattern at the detector
K	=	rotation matrix derived from test-platform kinematics
L	=	half-spacing between peak pairs along the detector array, m
M	=	supersampling ratio
m	=	peak location, pixels
N	=	number of detector pixels
N_F	=	Fresnel number
N_i	=	number of samples in the ith annular sample ring
n	=	pixel index
n_{glass}	=	index of refraction for the sensor cover glass
p	=	chord region index on the sun disk
peaks	=	set of peak locations
q	=	centroid width, pixels
$R_i(\alpha)$	=	principal axis rotation about axis i
$\boldsymbol{r}_1$	=	vector from the test-platform origin (center of rotation) to the sun lamp, scaled so $r_{1_z} = 1$.
$\boldsymbol{r}_2$	=	vector from the sensor origin to the test-platform origin, same scale as $\boldsymbol{r}_1$
S	=	discrete Fourier transform of the sensor image
$\hat{\boldsymbol{s}}_0$	=	free-space unit vector from the sun
$\hat{\boldsymbol{s}}$	=	refracted unit vector from the sun
U	=	nondimensional distance along the detector array
V	=	chord segment area, as a fraction of the whole disk
w	=	slit half-width, m
w_G	=	additive white Gaussian noise process
$\tilde{x}$	=	estimate of x
$x[n]$	=	discrete signal
$x(t)$	=	continuous signal
α	=	pattern shift, pixels
β	=	error between the nonlinear least-squares image model and the real image
δ	=	generic error quantity
ΔT	=	pixel spacing, m
ζ	=	radial angle from the boresight, rad
η_1, η_2	=	tip/tilt parameters in the laboratory mounting model, rad
$\theta_1, \theta_2, \theta_3$	=	test-platform joint angles, rad
λ_0	=	free-space wavelength, m
λ_R	=	refracted wavelength, m
ξ	=	angle to chord segment of the sun disk, rad
ρ_x, ρ_y	=	location of pixel 0 in the image plane, m
σ	=	width of the Gaussian modeled peak, pixels
σ_G	=	standard deviation of the noise process
τ	=	displacement along the detector array from pixel 0, m
ϕ	=	clock angle between aperture slits and sensor housing, rad
Ψ	=	phase angle of the discrete Fourier transform ratio
ψ	=	angle between the x axis and the sensor detector array, rad
Ω	=	solid angle of part of the sensor field of view, sr

Subscripts

mdl	=	image produced by the estimation model
sim	=	image produced by optics simulation
x, y, z	=	vector components
x_1, x_2, y_1, y_2	=	contribution from a particular slit
0	=	zero shift

I. Introduction

TRADITIONALLY, sun sensors have been regarded as a low-accuracy, low-cost complement to a satellite's attitude

Received 31 May 2007; revision received 11 September 2007; accepted for publication 18 September 2007. Copyright © 2007 by John Enright, Albert Yam, and Chris Li.. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission. Copies of this paper may be made for personal or internal use, on condition that the copier pay the $10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 0022-4650/08 $10.00 in correspondence with the CCC.

[*] Assistant Professor, Department of Aerospace Engineering, 350 Victoria Street. Member AIAA.

[†] MA.Sc. Candidate, Department of Aerospace Engineering, 350 Victoria Street. Student Member AIAA.

determination and control system. Recent years have seen the introduction of a number of new two-axis sensor designs. This new generation of sensors has endeavored to improve the accuracy of these sensors beyond coarse attitude estimation. Despite the fact that they are limited to measuring two components of a spacecraft's attitude, high-accuracy sun sensors can help enable low-cost missions.

Approaches to fabricating these high-accuracy sun sensors vary. Some adopt analog designs and achieve the accuracy through careful calibration [1,2]. Others employ two-dimensional detector arrays and a series of pinhole apertures [3]. Another approach is to fabricate two single-axis sensors in a single monolithic device [4–7]. One novel approach is to make use of a clever aperture design to allow two-axis measurements to be made using a single detector array [8]. Another example of this latter type is the Sinclair Interplanetary SS-330 digital sun sensor, a device we shall consider at some length.

This class of sensor typically employs narrow slits to create a series of bright peaks on a linear detector array. As with all digital sun sensors, the sensor accuracy is heavily dependent on the processing of the detector image. Image processing can be divided into two stages. The initial processing step must analyze the intensity pattern from the detector array and identify and classify the contributing peaks. We refer to this task as *peak-position estimation* (PPE). Once the peak locations have been determined, a second processing step converts the displacement measurements into an estimate of the incident sun vector using a geometric model of the sensor's optics. Of these two algorithms, the former is the more interesting theoretical problem.

Our previous studies [9,10] considered how parametric algorithms could be used to produce better PPEs than traditional approaches. Because these algorithms use explicit information about the expected detector illumination, we can actually optimize an aperture pattern for better performance [11]. These early studies considered a simplified, one-axis sensor, but to be practical, our analysis must be extended to provide two-axis estimation.

This paper presents initial results of modeling and algorithm development for a linear detector, two-axis sensor. We present a theoretical model of detector illumination, discuss the PPE problem as it applies to these sensors, and compare the model predictions with laboratory results. These experiments were conducted in the Space Avionics and Instrumentation Laboratory (SAIL) at Ryerson University.

II. System Models

The contributions of this study are best understood in the context of our modeling approach. Figure 1 illustrates our testing strategy. The first step in the test procedure is image generation. We have the option of using either *synthetic* images using our sensor illumination model or *real* sensor images using the laboratory hardware. Synthetic images are used to test new sensor designs and algorithms. Images captured using the laboratory test platform can be for validation of both the simulation and the sensor processing routines.

The sun/sensor orientation is defined by $\hat{s}_0$, the unit vector from the sun to the sensor. This vector is expressed in the sensor frame of reference. Once we have a suitable image, we can apply the sensor's processing routines to generate estimates of the peak displacements (e.g., $\tilde{m}_{x_i}$ and $\tilde{m}_{y_i}$). Finally, the peak locations are used to calculate the estimated sun vector $\tilde{s}_0$. The angle between $\tilde{s}_0$ and $\hat{s}_0$ is the error in the reading and is a convenient metric for evaluating algorithm performance.

A. SS-330 Digital Sun Sensor

Our simulation is designed to represent the Sinclair Interplanetary SS-330, a modern two-axis digital sun sensor. This device measures the angular position of the sun within a 70-deg half-angle field of view (FOV). The aperture mask is made up of two orthogonal pairs of narrow slits. The slit half-width w for this particular unit is 8.8 μm (some variation is expected in the manufacturing process). The detector is a single 16-mm-long linear array of 256 pixels oriented at approximately 45 deg to the aperture slits. The peak of the detector's spectral sensitivity lies at $\lambda_0 = 700$ nm. Between the mask and the detector is a neutral-density (ND) filter with a moderate index of refraction ($n_{\text{glass}} = 1.51$). The nominal filter thickness h is 4.0 mm, plus some allowance for the detector encapsulation.

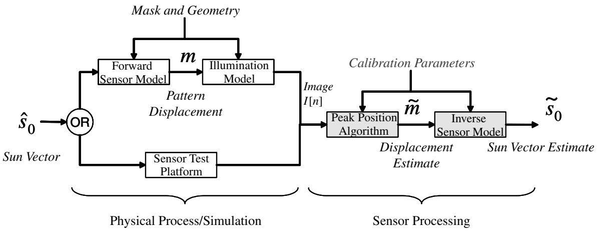


Fig. 1 A schematic representation of the sensor testing strategy.

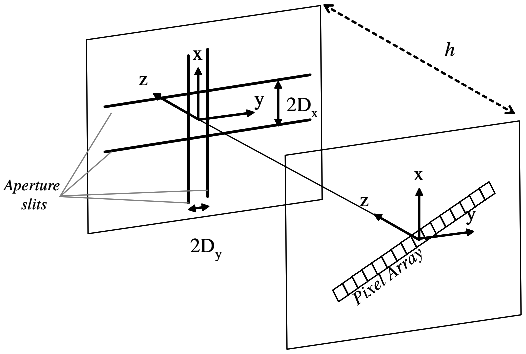


Fig. 2 Geometric model of sensor showing aperture and image planes.

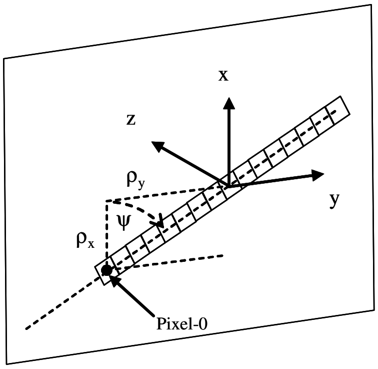


Fig. 3 The model must locate and orient the sensor array in the detector plane.

B. Sensor Geometry Model

The modeled geometry is shown in Fig. 2. Light entering the cover glass strikes the aperture mask. This mask lies in the plane $z = h$. The transmitted light then propagates through the ND filter and strikes the imaging array. Because the operation of this sensor is not dependent on the absolute detector response, we do not model extinction from the ND filter. By definition, this plane is $z = 0$, and the effective thickness h is the first calibration parameter. This parameter captures the thickness of the ND filter and the cover of the sensor. Because the indices of refraction are very close, we treat this as a single diffracting layer with index n_{glass}. The glass on the $+Z$ side of the mask does not affect the calculations. The slit spacings D_x and D_y are assumed to be known quantities.

The origin of the sensor coordinate frame lies in the image plane. The location of the origin is chosen so that the $+Z$ axis intersects the $z = h$ plane halfway between both the X- and Y-axis slits. The sensor array must also be located in the image plane (Fig. 3). The array placement is defined by three model parameters: ρ_x and ρ_y, which locate the center of pixel 0, and ψ, which describes the angle between the x axis and the sensor array. We also assume a known pixel spacing of ΔT. Sunlight passing through the slits on the sensor mask creates two pairs of bright lines on the detector plane. Because the detector cuts diagonally across the detector plane, it will intersect each of the sun lines, creating four bright peaks in the array response. The positions of these pairs of peaks (corresponding to the two pairs of slits) will change as the sun vector moves.

Simple geometric ray tracing relates the incident sun vector $\hat{s}_0$ to the peak locations m. The first important relation is between the components of the refracted sun vector and the free-space sun vector. Recognizing that the free-space vector is expressed with respect to the sensor housing and the refracted vector is expressed with respect to the sensor slits gives

$$\begin{aligned} s_x &= \frac{1}{n_{\text{glass}}}(s_{0_x}\cos\phi + s_{0_y}\sin\phi), \\ s_y &= \frac{1}{n_{\text{glass}}}(-s_{0_x}\sin\phi + s_{0_y}\cos\phi), \\ s_z &= -\sqrt{1 - s_x^2 - s_y^2} \end{aligned} \tag{1}$$

To account for boresight rotations caused by mounting or manufacturing limitations, Eq. (1) includes a z-axis rotation through an angle ϕ between the sensor frame and the platform frame. It can be shown that the peak locations m along the array are then

$$\begin{aligned} m_{x_1} &= \frac{1}{\Delta T\cos\psi}\left[-\left(\frac{s_x h + s_z D_y}{s_z}\right) - \rho_y\right] \\ m_{x_2} &= \frac{1}{\Delta T\cos\psi}\left[-\left(\frac{s_x h - s_z D_y}{s_z}\right) - \rho_y\right] \end{aligned} \tag{2}$$

$$\begin{aligned} m_{y_1} &= \frac{1}{\Delta T\sin\psi}\left[-\left(\frac{s_y h + s_z D_x}{s_z}\right) - \rho_x\right] \\ m_{y_2} &= \frac{1}{\Delta T\sin\psi}\left[-\left(\frac{s_y h - s_z D_x}{s_z}\right) - \rho_x\right] \end{aligned} \tag{3}$$

During online processing, the sensor must calculate the inverse of these functions to estimate the sun vector from the observed peaks of the illumination pattern. Defining the temporary quantities

$$A_m = \left[\Delta T\cos\psi\left(\frac{m_{x_1} + m_{x_2}}{2}\right) + \rho_y\right] \tag{4}$$

$$B_m = \left[\Delta T\sin\psi\left(\frac{m_{y_1} + m_{y_2}}{2}\right) + \rho_x\right] \tag{5}$$

It follows that

$$s_z = -\sqrt{\frac{h^2}{A_m^2 + B_m^2 + h^2}} \tag{6}$$

and hence,

$$s_x = -\frac{A_m}{h}S_z \tag{7}$$

$$s_y = -\frac{B_m}{h}S_z \tag{8}$$

C. Illumination Model

The optical path through the sensor is simple enough that ray optics can locate the illumination peaks on the detector. Accurate replication of the whole illumination patterns requires a more sophisticated model of propagation. Thus, we consider the Fresnel number N_F of one of the slits:

$$N_F = \frac{w^2}{\lambda_R h} \tag{9}$$

Taking into account the index of refraction, the Fresnel number for any particular free-space wavelength λ_0 is then

$$N_F = \frac{w^2 n_{\text{glass}}}{\lambda_0 h} \tag{10}$$

The detector sensitivity peaks at $\lambda_0 = 700$ nm. At this wavelength, $N_F = 0.084$. This value indicates that the optical characteristics near the sensor boresight lie between the Fresnel and Fraunhofer regimes. In these regimes, diffraction plays an important role in light propagation from the aperture to the detector.

1. *Plane-Wave Model*

In previous work [9], we employed a simple illumination model based on Fresnel diffraction from a single-slit aperture. Treating the aperture as an infinite narrow slit, we adapt Goodman's [12] treatment of a square aperture to derive the exact Fresnel solution. Using the nondimensional distance

$$U = \frac{\tau}{\sqrt{\lambda_R h}} \tag{11}$$

and the Fresnel number N_F of the slit from Eq. (9), the continuous intensity pattern at the image plane is given by

$$\begin{aligned} I_c(U) = \tfrac{1}{2}\{&[C_f(\sqrt{2}(\sqrt{N_F} - U)) - C_f(\sqrt{2}(\sqrt{N_F} + U))]^2 \\ &+ [S_f(\sqrt{2}(\sqrt{N_F} - U)) - S_f(\sqrt{2}(\sqrt{N_F} + U))]^2\} \end{aligned} \tag{12}$$

The center of this pattern lies at $U = 0$. If we know from our geometry models [Eqs. (2) and (3)] that we wish to shift the center of the peak by m pixels, then

$$I_{\text{shift}}(U, m) = I_c\left(U - \frac{m\Delta T}{\sqrt{\lambda_R h}}\right) \tag{13}$$

The sunlight entering the detector is incoherent. Incoherent light is additive in intensity, and so the net intensity distribution from the four peaks is then

$$\begin{aligned} I_4(U) &= I_{\text{shift}}(U, m_{x_1}) + I_{\text{shift}}(U, m_{x_2}) + I_{\text{shift}}(U, m_{y_1}) \\ &+ I_{\text{shift}}(U, m_{y_2}) = \sum_i I_{\text{shift}}(U, m_i) \end{aligned} \tag{14}$$

2. *Effect of Finite Sun Width*

The large distance between the sensor and the sun allows us to assume that the incident light arrives as plane waves. This does not mean that we can treat the sun as a point source of illumination. The finite angular radius (0.26 deg) will create a spreading of the illumination pattern. To account for the size of the sun disk, we segment the sun into a number of illuminating bands (Fig. 4) parallel to the slit axis. The sun disk is circular, and so the diagram is applicable regardless of which aperture slit we are considering. Each band lies at an angular distance ξ off-axis, is of width $\Delta\xi$, and is treated as a source of incident plane waves. The angle between the propagation vector and the inward sensor normal is ξ (Fig. 5).

Because the range of incidence angles across the solar disk is small and the illumination is incoherent, it is possible to show that the off-axis illumination pattern is a linear shift of the normally incident case. For small ξ,

$$I_\xi(U) = I_c\left(U - \xi\sqrt{\frac{h}{\lambda_R}}\right) \tag{15}$$

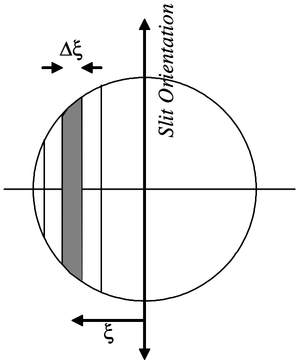


Fig. 4 Division of the sun disk into a number of illuminating bands.

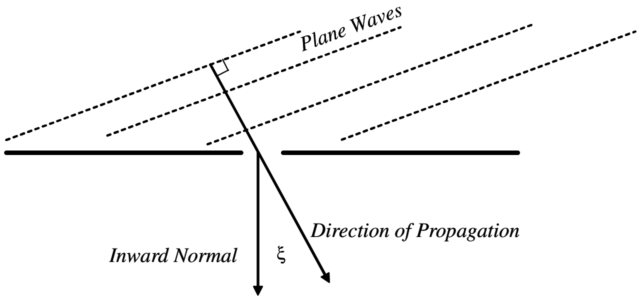


Fig. 5 Plane waves arriving from off-axis sources.

The relative area of these illumination bands is used to weight the relative intensities of their contributions to the synthetic images. If these bands are approximated as planar regions, the band centered at ξ has the area (in arbitrary units)

$$V(\xi) = \frac{1}{2}\left\{\xi(\xi_{\max}^2 - \xi^2)^{1/2} + \xi_{\max}^2 \operatorname{atan}\left(\frac{\xi}{(\xi_{\max}^2 - \xi^2)^{1/2}}\right)\right\}\Bigg|_{\xi-\frac{\Delta\xi}{2}}^{\xi+\frac{\Delta\xi}{2}} \tag{16}$$

If the sun disk is divided into $2p_{\max}$ bands, each of width $\Delta\xi = \xi_{\max}/p_{\max}$, then

$$\xi(p) = (p - \tfrac{1}{2})\Delta\xi \tag{17}$$

and the net illumination pattern is

$$I_{\text{sun}}(U) = V_S \sum_{p=-p_{\max}}^{b_{\max}} V(p) I_c\left(U - \xi(p)\sqrt{\frac{h}{\lambda_R}}\right) \tag{18}$$

The quantity V_S is used to scale the total intensity from the slit. We are more interested in the shape of the illumination pattern rather than its photometric accuracy, and so we select V_S to normalize the peak intensity.

Combining the superposition of multiple slits [Eq. (14)] with the sun-width effect gives an expression for the continuous illumination pattern at the detector:

$$I_{\text{net}}(U) = V_S \sum_i \left\{ \sum_{p=-p_{\max}}^{p_{\max}} V(p) I_c\left(U - \xi(p)\sqrt{\frac{h}{\lambda_R}} - \frac{m_i}{\sqrt{\lambda_R h}}\right)\right\} \tag{19}$$

3. *Effect of Pixel Width*

The continuous illumination pattern $I_{\text{net}}(U)$ has features that may not be resolved by the coarse spacing of the detector pixels. We adopt a simple, if inelegant, approach to capturing this effect. Essentially, we supersample $I_{\text{net}}(U)$ by a factor M and bin the resulting sequence into pixels. Thus, if the locations of the pixel centers τ_p lie at

$$\tau_p(n) = n\Delta T - \left\lfloor \frac{N}{2} \right\rfloor \le n < \left\lfloor \frac{N}{2} \right\rfloor \tag{20}$$

then the sample locations τ_s that contribute to the pixel response must lie at

$$\tau_s(i, n) = n\Delta T + \left(i - \frac{1}{2}\right)\frac{\Delta T}{M} - \left\lfloor \frac{M}{2} \right\rfloor \le i < \left\lfloor \frac{M}{2} \right\rfloor \tag{21}$$

The endpoints are common to adjacent pixels, and so the sampled contributions must be shared. To within a normalizing constant, the *discrete* response from pixel n is then

$$I_{\text{pix}}[n] = \left\{ \sum_{i=-\lfloor\frac{M}{2}\rfloor}^{\lfloor\frac{M}{2}\rfloor} I_{\text{net}}\left(\frac{\tau_s(i, n)}{\sqrt{\lambda_R h}}\right)\right\} - \frac{1}{2}\left\{ I_{\text{net}}\left(\frac{\tau_s((M/2), n)}{\sqrt{\lambda_R h}}\right) + I_{\text{net}}\left(\frac{\tau_s(-(M/2), n)}{\sqrt{\lambda_R h}}\right)\right\} \tag{22}$$

4. *Sensor Noise*

The last factor added to the model is a degradation to the signal caused by sensor noise. Because the observed noise standard deviation in this sensor is quite small (less than 1%), we consciously adopted a nonphysical additive white Gaussian noise model for the sake of simplicity. More realistic noise models are available, but are judged to be unnecessary in this study. Therefore, the net simulated image is formed by

$$I_{\text{sim}}[n] = I_{\text{pix}}[n] + w_G[n] \tag{23}$$

where w_G is an additive white Gaussian noise process with zero mean and standard deviation σ_G.

5. *Model Evaluation*

We can examine the agreement between true sensor images and the model predictions. Some preprocessing of the laboratory data is necessary before the real and synthetic images can be fairly compared. First, we make an initial correction for sensor dark current, an unmodeled noise source. We can remove most of the dark-current signal from the real images by subtracting the minimum image intensity from all of the pixels. Second, we must normalize the dark-current corrected image, effectively removing the effects of variable detector gain and integration time. Using peak displacements calculated from a known $\hat{s}_0$ and the physical design and calibration parameters from the SS-330 (Table 1), we simulate images for each sample location.

A typical measured and noise-free simulated image is shown in Fig. 6. Qualitatively, the real and synthetic images match well in both position and shape. The sharpness of each peak is quite similar as it drops from its peak value to around 5%. We can also see evidence that diffraction-induced broadening at the base of the sensor peaks was captured in the model. However, two notable discrepancies

Table 1 Simulation parameters

Parameter	Value	Rationale and units
N	256	Detector specification
λ_0	700 nm	Peak of detector sensitivity
ΔT	63.5 μm	Detector specification
n_{glass}	1.51	Material specification
$\xi_{\max}$	0.26 deg	Physical constant
w	8.8 μm	Estimated from mask images
h	4.37 mm	Calibrated
ψ	−0.790 rad	Calibrated
D_x	0.75 mm	Design specification
D_y	0.25 mm	Design specification
$p_{\max}$	20	Chosen
M	16	Chosen
σ_G	0.01	Estimated from sensor data

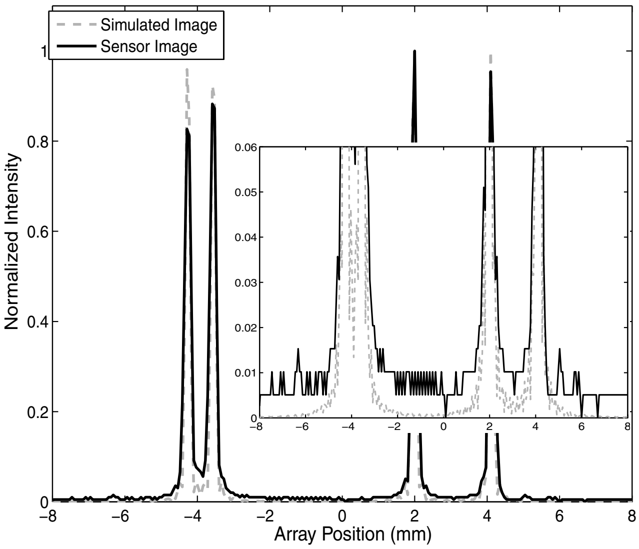


Fig. 6 Comparison of typical simulated and measured images.

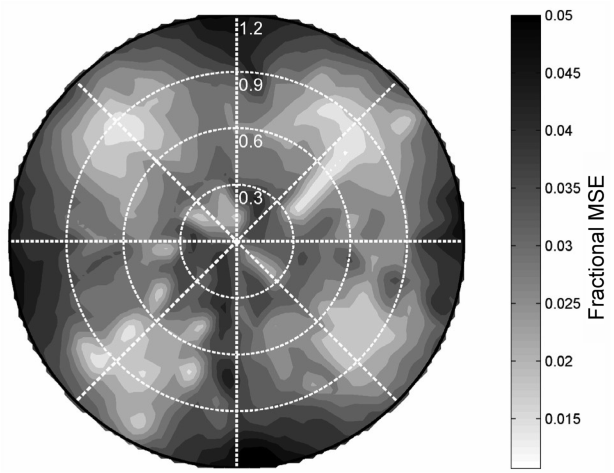


Fig. 7 Variation of MSE within FOV (radial ticks in radians).

exist. First, there are peak-amplitude variations in the sensor image that do not appear in the simulated images. From visual inspection of the sensor, we believe that nonuniformities introduced during the manufacture of the aperture may be responsible for this effect. Second, there are visible departures in the low-intensity behavior of the real and synthetic signals. The elevated intensity in the real image close to the edges of the detector array suggests an incomplete removal of the dark-current contributions. Better modeling and characterization of the dark current may alleviate this problem. The observed intensity variations in the gaps between peaks is greater still. Scattering from the edges of the aperture slits may explain the increased sensor response. More study is needed to further account for these deviations between modeled and observed performance.

A more quantitative assessment of the model fidelity can be obtained by calculating the rms error between simulated and true images for the same sun vector (Fig. 7). Although the agreement is quite good (less than 2% rms) in large regions of the sensor FOV, we can clearly see a degradation in the model fidelity in other areas (∼5% rms). Weighted by the solid angle around each sample point, the mean error over the FOV is about 3.2%. Variable extinction from the ND filter, changes to the apparent width of the slits, and even the finite thickness of the mirror coating can have an effect on the observed mean square error (MSE). We also observe a degradation in fidelity along the principal axes of the sensor, creating a four-lobed pattern in the figure. The detector is composed of two 128-pixel subarrays, rather than a single monolithic line of pixels, and this may cause distortions when the peak patterns straddle the seam.

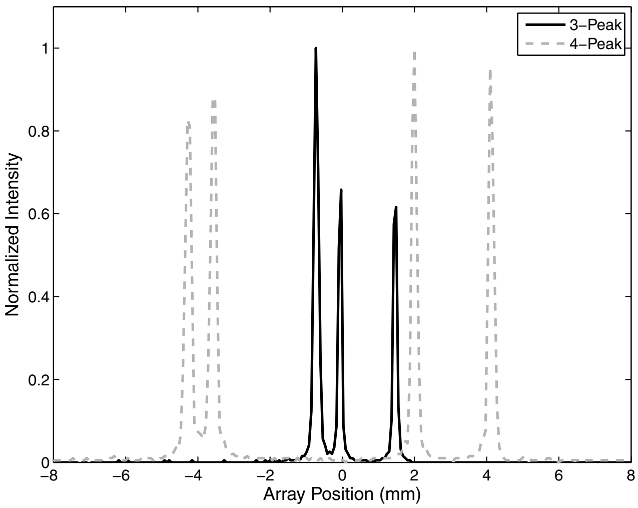


Fig. 8 Three- and four-peak images.

Two conclusions can be drawn from these results. First, the simulation fidelity may vary throughout the FOV. We expect that the sensor simulation will be a more accurate predictor of sensor performance when the sun is near the areas of good agreement. Because we are less confident of the model fidelity in other areas, we must be correspondingly more cautious about predicting trends in sensor and algorithm behavior in this regime. This bounds the utility of a sensor design based exclusively on simulation. Second, variations in peak amplitude are an obvious source of disparity between real and synthetic images. Such variations are the single largest contributor to the image MSE. A extension to the model that captures this effect will improve the utility of this simulation.

III. Peak-Position Estimation

Accurate estimation of the sun vector relies on the effectiveness of the sensor's processing routines. Of the tasks that the sensor must perform, the inverse geometry model is the most pedestrian. As long as we are careful in measuring the model parameters through calibration, the calculation of the sun vector is a straightforward endeavor. In contrast, peak-position estimation is more difficult and admits a diversity of approaches.

The PPE algorithms must identify peaks in the image, measure their positions precisely, and associate the peaks with the slits that produced them. The precise localization of a bright peak has long been studied in space-related research. Stone [13] presents a good historical discussion of several simple techniques for processing astronomical data. As the cost of computers dropped, it became worthwhile to add online processing to sensors such as star trackers [14,15] and, eventually, sun sensors [7]. PPE also has analogs in the classic registration problems in image registration and delay estimation in radar or sonar processing.

A. Peak Detection

The first step required in the image processing sequence is to locate the brightest parts of the detector image. The array locations of the brightest pixels is a simple way of estimating the d values. Past models of the SS-330 have used an algorithm very similar to the one presented here.

We expect to find three or four peaks in each image, depending on whether the light from two orthogonal slits overlaps or not. Figure 8 shows typical examples of these two cases. Peak amplitude varies both within and between the pairs of peaks. These amplitude variations are most noticeable in three-peak images. Although the superposition treatment of overlapping peaks is not strictly correct (there are not two independent light sources), affected pixels *are* exposed to more incident light and hence display a stronger response than that observed from a single peak.

Because sensor noise levels are low and the peaks are generally sharply defined, we can use a simple heuristic for the initial identification. This algorithm looks for a central maximum in a sliding five-pixel window and falls back to results from a simple three-pixel window if fewer than three peaks are detected (this can happen in some overlapping cases). In mathematical terms, we mark pixel n as a peak if

$$I[n] > I[n-2], I[n-1], I[n+2], I[n+1] \tag{24}$$

This labeling creates a set of identified peaks:

$$\{\tilde{m}_1, \tilde{m}_2, \ldots,\} \tag{25}$$

where m_i are simply the peaks detected from left to right in the image. These are *physical peaks*. We do not yet know which slits contribute to each peak. Each slit produces a *logical peak*. These must be mapped onto the physical peaks.

The expected spacings between peak pairs $2L_x$ and $2L_y$, are independent of the sun vector. From Eqs. (2) and (3),

$$\begin{aligned} L_x &= \frac{1}{2}\Delta T(m_{x_2} - m_{x_1}) = \frac{D_y}{\cos\psi}, \\ L_y &= \frac{1}{2}\Delta T(m_{y_2} - m_{y_1}) = \frac{D_x}{\sin\psi} \end{aligned} \tag{26}$$

Classifying the peaks is then a matter of finding the mapping between observed peaks (i.e., $\tilde{m}_{x_1}$, $\tilde{m}_{x_2}$, $\tilde{m}_{y_1}$, and $\tilde{m}_{y_2}$) to their originating slits (i.e., from peaks) that minimizes the resulting errors in spacing:

$$\Phi = \left(L_x - \frac{\Delta T}{2}(\tilde{m}_{x_2} - \tilde{m}_{x_1})\right)^2 + \left(L_y - \frac{\Delta T}{2}(\tilde{m}_{y_2} - \tilde{m}_{y_1})\right)^2 \tag{27}$$

These $\tilde{m}$ values are used to calculate the sun-vector estimate using the inverse geometry model [Eqs. (6–8)].

B. Centroiding

A common approach to obtaining subpixel accuracy in digital sensors is to compute centroids of the illumination instead of working with just the brightest pixels. Applying this strategy to our sensor, the algorithm first finds peaks and then computes a q-point centroid around each peak. Thus,

$$\tilde{m}_i = \frac{\sum_{l=n_i-q}^{n_i+q} l \cdot I[l]}{\sum_{l=n_i-q}^{n_i+q} I[l]} \tag{28}$$

This algorithm then calculates the optimal matching from Eq. (27) and resulting sun vector from Eqs. (6–8). Because we are working with more precise estimates of the peak locations, the geometry model generally gives better estimates of the sun vector.

C. Linear Phase

In other studies, we have presented a simple frequency-domain-estimation algorithm based on the classic phase-correlation method [16]. Briefly, given a discrete signal and its discrete Fourier transform (DFT),

$$I([n] \overset{\text{DFT}}{\Leftrightarrow} S[k]) \tag{29}$$

we can neatly express the frequency-domain effect of a space-domain shift:

$$I[n-\alpha] \overset{\text{DFT}}{\Leftrightarrow} S[k]e^{-j\frac{2\pi k\alpha}{N}} \tag{30}$$

In the second case, the illumination was shifted by α samples: a shift in the space domain is equivalent to a multiplication by a linear phase term in the frequency domain. What is significant is that this interpretation holds even if α is not an integer.

If $S_0[n]$ is the DFT of the zero-shift signal and $S_\alpha[n]$ is the DFT of the shifted version, the exponential term can be isolated:

$$\Psi[k] = \text{slope}\left(\frac{S_\alpha[n]}{S_0[n]}\right) = \text{slope}(e^{-j\frac{2\pi k\alpha}{N}}) = -\frac{2\pi\alpha k}{N} \tag{31}$$

A least-squares line fit of $\Psi[k]$ gives an estimate of the slope $\tilde{\alpha}$. This quantity is directly related to the peak positions:

$$\tilde{m} = \pm L/2 + \tilde{\alpha}\Delta T \tag{32}$$

Typical two-dimensional applications of the phase-correlation method [17] will naturally have a 2-D DFT from which to extract phase information. Because the SS-330 only provides linear samples, the image components from each axis must be isolated before applying this algorithm. Instead of resolving both axes together, the problem is viewed as two sequential 1-D problems. Each sensor image can be viewed as the superposition of three subimages:

$$I[n] = I_x[n] + I_y[n] + I_n[n] \tag{33}$$

where $I_x[n]$ and $I_y[n]$ are the contributions from the x- and y-axis slits, and $I_n[n]$ contains noise and other artifacts. Furthermore, each primary subimage is simply a shifted version of a normally illuminated pair of slits. Specifically,

$$I_x[n] = I_{x_0}[n - \alpha_x], \qquad I_y[n] = I_{y_0}[n - \alpha_y] \tag{34}$$

Thus, when presented with a sensor image, the linear phase (LP) algorithm takes an initial pass through the image and identifies the peaks to the nearest pixel ($\tilde{m}_i$). A simple heuristic state machine implements hysteresis and thresholding to help provide immunity from noise and spurious signals (e.g., Earth's albedo). This approach is conservative about partially overlapping peaks, intentionally classifying them as a single peak. The coarse peak estimates are used to perform an optimal mapping to slits. At this point, our processing diverges from the previous algorithms.

Instead of calculating the sun vector from the coarse peak-position estimates, the peak classifications are used to decompose the sensor image into the subimages of Eq. (33). Images containing four well-resolved peaks are easily separated into contributions from each axis. Conversely, if the algorithm finds only three peaks (i.e., two are overlapping), the situation is more complex. In this study, we adopt the crude but robust approach of ignoring the overlapping peaks entirely. Thus, I_x and I_y become single peak images. Because the algorithm knows which peak from each pair is usable, it can select an appropriate reference image for I_{x_0} and I_{y_0}. Using the component images and Eq. (31), the algorithm computes $\tilde{\alpha}_x$ and $\tilde{\alpha}_y$. From the definition of the sensor coordinate system, the unshifted peaks should lie at $\pm L_x/2$ and $\pm L_y/2$. Therefore, Eq. (32) becomes

$$\tilde{m}_x = \pm L_x/\Delta T + \tilde{\alpha}_x \tag{35}$$

$$\tilde{m}_y = \pm L_y/\Delta T + \tilde{\alpha}_y \tag{36}$$

and $\tilde{s}_0$ is computed from the inverse geometry model.

D. Nonlinear Least-Squares Technique

The second parametric model that shows promise for these sensors is a nonlinear least-squares (NLSQ) technique. Using a simple parameterized approximation of the illumination pattern, the algorithm identifies the peak locations. NLSQ techniques have been widely used in other satellite signal processing applications, often for star trackers, but sometimes for sun sensors as well. Typically, NLSQ can be used to register a single feature such as the centroid of a star [18] or a pattern of features such as the distribution of pinholes [3].

The following derivation is adapted from a conventional Gauss–Newton NLSQ algorithm. The algorithm relies on a simplified model of illumination that will be used by the online routines. This image model does not have to reflect the physics of image formation, it merely has to approximate the shape. Denoting the modeled continuous pattern from a single slit as $I_{\text{mdl}_1}(\tau)$, the illumination at the detector will be of the form

Table 2 Image classifications

Image type	Characteristics	$\boldsymbol{H}_\lambda$ matrix
0	Any image with four peaks	$[\boldsymbol{A}\quad \boldsymbol{B}\quad \boldsymbol{C}\quad \boldsymbol{D}\quad (a_1\boldsymbol{E}+a_2\boldsymbol{F})\quad (b_1\boldsymbol{G}+b_2\boldsymbol{H})]$
1	A and C overlap	$[(\boldsymbol{A}+\boldsymbol{C})\quad \boldsymbol{B}\quad \boldsymbol{D}\quad (a_1\boldsymbol{E}+a_2\boldsymbol{F})\quad (a_1\boldsymbol{G}+b_2\boldsymbol{H})]$
2	B and C overlap	$[\boldsymbol{A}\quad (\boldsymbol{B}+\boldsymbol{C})\quad \boldsymbol{D}\quad (a_1\boldsymbol{E}+a_2\boldsymbol{F})\quad (a_2\boldsymbol{G}+b_2\boldsymbol{H})]$
3	A and D overlap	$[(\boldsymbol{A}+\boldsymbol{D})\quad \boldsymbol{B}\quad \boldsymbol{C}\quad (a_1\boldsymbol{E}+a_2\boldsymbol{F})\quad (b_1\boldsymbol{G}+a_1\boldsymbol{H})]$
4	B and D overlap	$[\boldsymbol{A}\quad (\boldsymbol{B}+\boldsymbol{D})\quad \boldsymbol{C}\quad (a_1\boldsymbol{E}+a_2\boldsymbol{F})\quad (b_1\boldsymbol{G}+a_2\boldsymbol{H})]$

$$I_{\mathrm{mdl}}(\tau) = a_1A(\tau) + a_2B(\tau) + b_1C(\tau) + b_2D(\tau) \tag{37}$$

where

$$\begin{aligned} A &= I_{\mathrm{mdl}_1}(\tau - m_{x_1}), & B &= I_{\mathrm{mdl}_1}(\tau - m_{x_1} - D_x), \\ C &= I_{\mathrm{mdl}_1}(\tau - m_{y_1}), & D &= I_{\mathrm{mdl}_1}(\tau - m_{y_1} - D_y) \end{aligned} \tag{38}$$

This model contains six parameters that must be identified (a_1, a_2, b_1, b_2, m_{x_1}, and m_{y_1}) four amplitudes, and two displacements. Only the latter two quantities are of direct interest, but because the amplitudes do vary, they must be determined as well. A simple Gaussian model captures the zero-shift peak shape:

$$I_{\mathrm{mdl}_1}(\tau) = e^{-\frac{\tau^2}{2\sigma^2}} \tag{39}$$

where the peak width σ is determined empirically.

For notational convenience, the image components and model parameter estimates can be expressed as vectors

$$\boldsymbol{A} = \{A_n\} = [A|_{\tau_1} A|_{\tau_2}]^T \tag{40}$$

(similarly with $\boldsymbol{B}$, etc.) and

$$\boldsymbol{\lambda} = [a_1 \quad a_2 \quad b_1 \quad b_2 \quad m_{x_1} \quad m_{y_1}]^T \tag{41}$$

Following a typical NLSQ implementation, the error observed between a real image ($I_s[i]$) and a modeled image can be written as

$$d\beta_n = I[n] - I_{\mathrm{mdl}}(\tau_n, a_1, a_2, b_1, b_2, m_{x_1}, b_{y_1}) \tag{42}$$

Partial derivatives of I_{mdl} with respect to the model parameters form the measurement matrix:

$$\boldsymbol{H}_\lambda = \begin{bmatrix} \left.\frac{\partial I_{\mathrm{mdl}}}{\partial a_1}\right|_{\tau_1} & \left.\frac{\partial I_{\mathrm{mdl}}}{\partial a_2}\right|_{\tau_1} & \\ \left.\frac{\partial I_{\mathrm{mdl}}}{\partial a_1}\right|_{\tau_2} & \ddots & \\ \vdots & & \left.\frac{\partial I_{\mathrm{mdl}}}{\partial m_2}\right|_{\tau_n} \end{bmatrix} \tag{43}$$

Each element in this matrix represents a partial derivative evaluated at the current estimate of the model parameters and a particular τ_i. The partial derivatives that make up $\boldsymbol{H}_\lambda$ can be evaluated directly. The first four are quite simple:

$$\begin{aligned} \left.\frac{\partial I_{\mathrm{mdl}_1}}{\partial a_1}\right|_\tau &= A(\tau) \\ \left.\frac{\partial I_{\mathrm{mdl}_1}}{\partial a_2}\right|_\tau &= B(\tau) \\ \left.\frac{\partial I_{\mathrm{mdl}_1}}{\partial b_1}\right|_\tau &= C(\tau) \\ \left.\frac{\partial I_{\mathrm{mdl}_1}}{\partial b_2}\right|_\tau &= D(\tau) \end{aligned} \tag{44}$$

The last two are more complex:

$$\begin{aligned} \left.\frac{\partial I_{\mathrm{mdl}}}{\partial m_{x_1}}\right|_\tau &= -\left[a_1\left(\frac{\partial}{\partial u} I_{\mathrm{mdl}_1}(u)\right|_{u=\tau-m_{x_1}}\right) \\ &\quad + a_2\left(\frac{\partial}{\partial u} I_{\mathrm{mdl}_1}(u)\right|_{u=\tau-m_{x_1}-L_x/\Delta T}\left.\vphantom{\frac{\partial}{\partial u}}\right)\Bigg] \\ &= a_1E(\tau) + a_2F(\tau) \\ \left.\frac{\partial I_{\mathrm{net}}}{\partial m_{y_1}}\right|_\tau &= -\left[b_1\left(\frac{\partial}{\partial u} I_{\mathrm{mdl}_1}(u)\right|_{u=\tau-m_{y_2}}\right) \\ &\quad + b_2\left(\frac{\partial}{\partial u} I_{\mathrm{mdl}_1}(u)\right|_{u=\tau-m_{y_2}-L_y/\Delta T}\left.\vphantom{\frac{\partial}{\partial u}}\right)\Bigg] \\ &= b_1G(\tau) + b_2H(\tau) \end{aligned} \tag{45}$$

Thus,$\boldsymbol{H}_\lambda$ can be succinctly written in matrix form as a series of column vectors:

$$\boldsymbol{H}_\lambda = [\boldsymbol{A} \quad \boldsymbol{B} \quad \boldsymbol{C} \quad \boldsymbol{D} \quad (a_1\boldsymbol{E} + a_2\boldsymbol{F}) \quad (b_1\boldsymbol{G} + b_2\boldsymbol{H})] \tag{46}$$

We can show that for each iteration of the algorithm, the parameter update $\boldsymbol{d\lambda}$ is found by solving the linear system:

$$\boldsymbol{H}_\lambda^T \boldsymbol{d\beta} = (\boldsymbol{H}_\lambda^T \boldsymbol{H}_\lambda)\boldsymbol{d\lambda} \tag{47}$$

For our standard model, the system has six equations and six unknowns. The algorithm terminates after a set number of iterations or when the changes to $\boldsymbol{\lambda}$ drop below a certain tolerance. Although the latter makes more sense from a numerical standpoint, the former exit criterion is a better match to embedded applications in which temporal determinism is desired.

The system of equations (47) is well-behaved when there are four distinct physical peaks in the image. When illumination from different slits overlap, the linear system for the parameter update becomes singular. Physically, there is no way of separating the two arbitrary overlapping peak amplitudes. To resolve this degeneracy, three-peak images must be treated as a series of special cases; one parameter is removed, and the overlapping peaks are assumed to have the same amplitude. Removing a parameter from the image model ensures that the linear system remains at full rank. Table 2 enumerates the different types of images that can be encountered and the corresponding construction of $\boldsymbol{H}_\lambda$. Four permutations of peak overlaps are possible.

V. Algorithm Testing

Having developed a sensor simulation and number of algorithms for sun-vector estimation, we must synthesize these components and evaluate the results. As outlined earlier, we have two goals. First, we would like to establish that performance predictions from *simulated* sensors can be used to predict the performance of real sensors. Second, we wish to evaluate the impact of processing improvements on the performance of the SS-330 sensor.

A. SAIL Test Facility

A schematic of the experimental setup is shown in Fig. 9. Sensor illumination is provided by a xenon arc lamp. A condenser lens is used to collimate and direct the light at the sensor. An adjustable iris restricts the apparent angular diameter of the light to that of the sun (i.e., 0.52 deg). This is essentially a *critical* (or *Abbé*) illumination

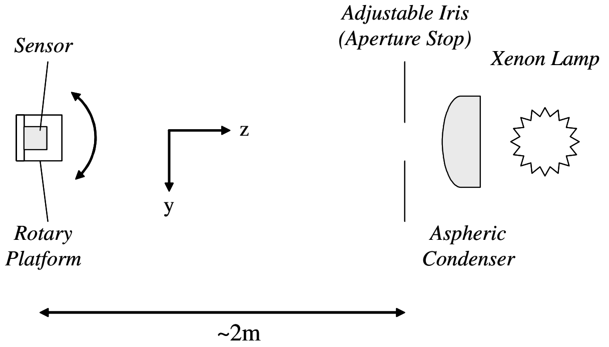


Fig. 9 Top-down schematic of experimental setup.

system, focused at infinity. Although critical systems suffer from nonuniformity across the illuminated field, this setup maximizes the brightness at the sensor using only a few simple components. Because the sensor is small, mounted close to the center of rotation, and relatively insensitive to slit-to-slit lighting variations, maintaining a large, uniform field is unnecessary. Thus, our laboratory setup is adequate for the SS-330, but other sensor designs may require a higher-fidelity solar simulator (e.g., integrating spheres, etc.).

The sensor is mounted to a three-axis rotary test platform (*X*–*Y*–*Z*). The three motion stages are computer-controlled and can be commanded in 0.001-deg increments (absolute repeatability is 0.002 deg, 1-σ). Only two motion stages are required for sun sensors, but the third joint helps to avoid singularities when taking measurements while the sensor is in motion (dynamic measurements are not within the scope of this study). Imperfect alignment of the sensor with respect to the center of rotation can be corrected during calibration. This simplifies test setup and reduces the need for high-precision alignment.

B. Calibration

We first calibrate our sensor to provide the geometric parameters needed for the forward and inverse sensor models. This is typically based on a test suite of 400 observations. Sample locations are selected so that the sun vectors form a series of evenly spaced rings around the boresight. The selected vectors span a conical FOV from the boresight out to 70 deg. For each $\hat{s}_0$, we use the motorized test platform and rotate the sensor to present the sun-simulator lamp in the correct orientation.

The number of samples in each ring is weighted by its subtended solid angle. This ensures relatively even sampling of the sensor FOV. Because this distribution is not perfect, when quantities are averaged over the FOV, each sample is weighted by the *actual* size of its corresponding annular sector. Thus, if the *i*th annular ring contains N_i samples and extends from ζ_{i_-} to ζ_{i_+}, the solid angle represented by each sample is

$$\Omega_i = \frac{2\pi}{N_i}(\cos\zeta_{i_-} - \cos\zeta_{i_+}) \tag{48}$$

The total solid angle seen by the sensor is

$$\Omega_T = 2\pi(1 - \cos\zeta_{\max}) \tag{49}$$

Accordingly, the expectation of some error quantity $\tilde{\delta}$ is found by a weighted sum over the elements in each ring:

$$\tilde{\delta} = \frac{1}{\Omega_T}\sum_i\sum_j \Omega_i\delta_{ij} \tag{50}$$

We call this the *solid-angle-weighted error*.

In addition to the five calibration parameters (h, p_x, p_y, ψ, and ϕ) used to describe the sensor construction, several other parameters are necessary to account for nonidealities of the laboratory setup. The sun vector is calculated from the platform kinematics and the mounting geometry:

$$\boldsymbol{r}_{\text{sun}_S} = R_x(\eta_1)R_y(\eta_2)(K^{-1}\boldsymbol{r}_1 + \boldsymbol{r}_2) \tag{51}$$

where $\boldsymbol{r}_1$ is a vector from the center of rotation of the test platform to the sun lamp, $\boldsymbol{r}_2$ is the offset between the sensor origin and the center of rotation, η_i capture tip-tilt mounting error, and K is a transformation matrix derived from the kinematics of the sensor test platform. The $\boldsymbol{r}_{\text{sun}}$ vector is normalized to form $\hat{s}_0$. Because of this normalization, the absolute scale of $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are unimportant: only their relative magnitudes are needed. This allows us to define r_{1_z} as equal to unity.

The calibration process produces a set of known joint angles and corresponding sensor images. Peak-position estimates are derived from these images using the NLSQ algorithm. Numerical optimization routines are then employed to minimize the rms error between estimated sun vectors from Eqs. (6–8) and the *known* sun vectors from Eq. (51) (strictly speaking, it is the joint angles that are known, not $\hat{s}_0$). The numerical minimization simultaneously solves for the 12 numerical values captured in the construction parameters (h, p_x, p_y, ψ, and ϕ) and the laboratory geometry (η_1, η_2, $\boldsymbol{r}_1$, and $\boldsymbol{r}_2$).

C. Algorithms Tests

Once the sensors have been calibrated, we take another set of measurements with the sensor and apply the estimation algorithms. These measurements are made at different sample locations from the calibration observations. At each location, we collect a sensor image and simulate a corresponding synthetic image for that location. Once the data collection is finished, we apply our four algorithms to the two data sets.

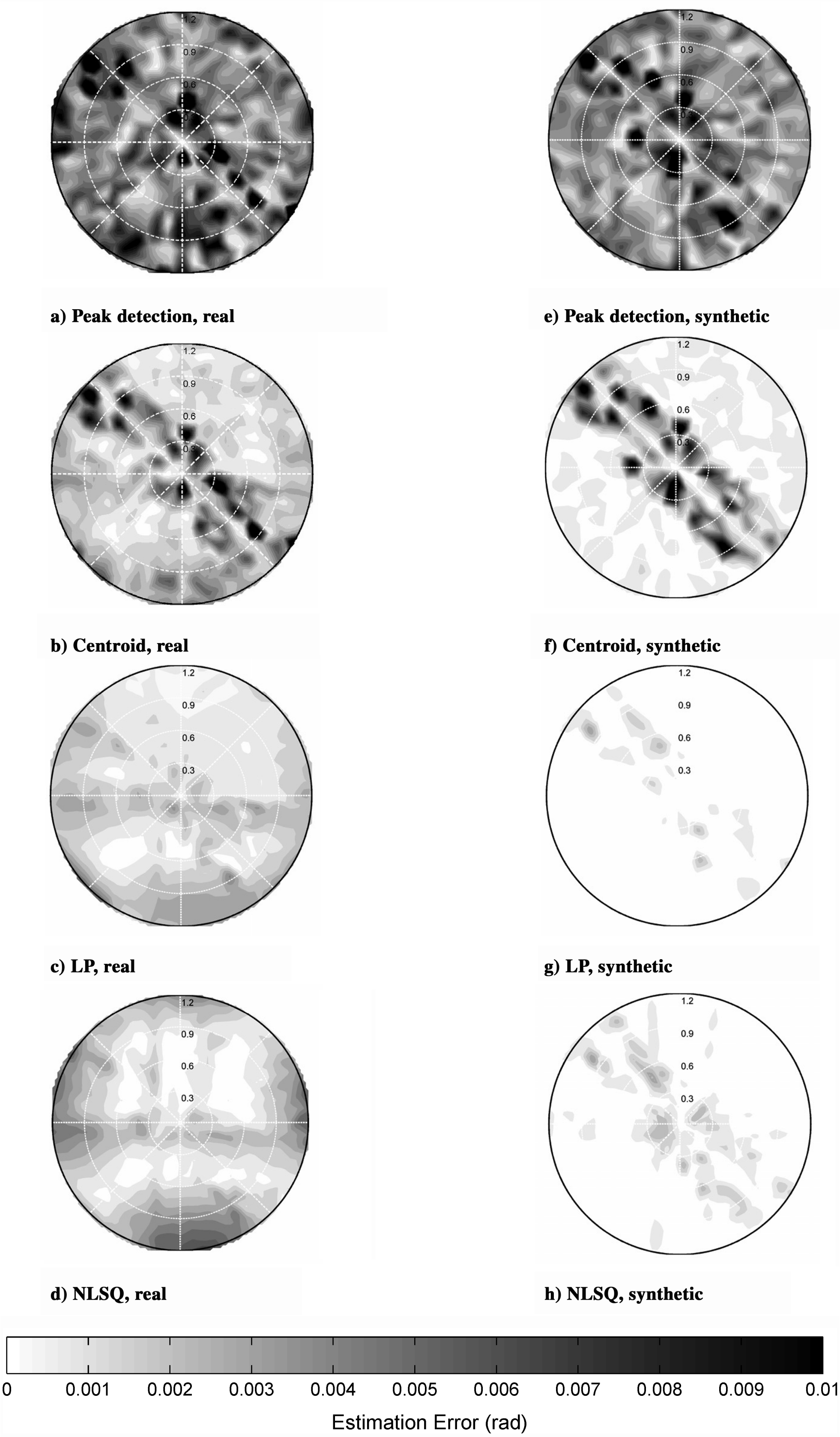
The aggregate results for the four algorithms are shown in Fig. 10. The subfigures show variations in algorithm performance across the sensor FOV. In these figures, the boresight angle was mapped to a radial distance. The size and rather noisy nature of the data present a challenge for visualization. Large features in the figures are accurate, but the routines used to spline and contour the available data may distort the precise shape of smaller features.

Several observations can be made from this data. First, the peak-detection and centroid algorithms perform similarly on both real and synthetic images. This is a consequence of the fact that these algorithms depend primarily on the locations of the brightest pixels. Second, we notice a diagonal band in the FOV in which performance is particularly poor. This is especially apparent in the peak-detection and centroiding algorithms, but can be seen in the other algorithms as well. This band represents regions in which the narrowly spaced peaks straddle one of the peaks from the second axis. The overlap in the image makes it difficult to identify and categorize the peaks. Third, the LP algorithms display greater variation in error when we consider the real images as opposed to the simulations. Although difficult to quantify in general terms, the real sensor images display a variety of artifacts that are not seen in the modeled images. In particular, partially overlapping peaks are often less well defined in the real sensor images, possibly due to scattering from the slit edges. Finally, the performance of the two parametric routines is generally a function of the distance from the boresight. Points near the center are identified accurately, but errors increase toward the edge of the FOV. This effect is only apparent in the real images; simulated images do not suffer from this degradation. This trend strongly suggests a dependence on boresight angle that is not captured in the sensor model.

D. Simulation Accuracy

Plotting the cumulative rms error from the boresight out to the margins of the FOV gives additional insight into the algorithm behavior. These results are shown in Figs. 11a and 11b. These plots are both encouraging and discouraging.

Near the boresight, the algorithm performs well in predicting the peak-detection and centroiding algorithms. The slight increase in error in the neighborhood of the boresight is also captured in the simulated images. This particular sensor exhibits overlapping peaks at zero sun angle, resulting in the observed performance degradation. As



a) Peak detection, real

e) Peak detection, synthetic

b) Centroid, real

f) Centroid, synthetic

c) LP, real

g) LP, synthetic

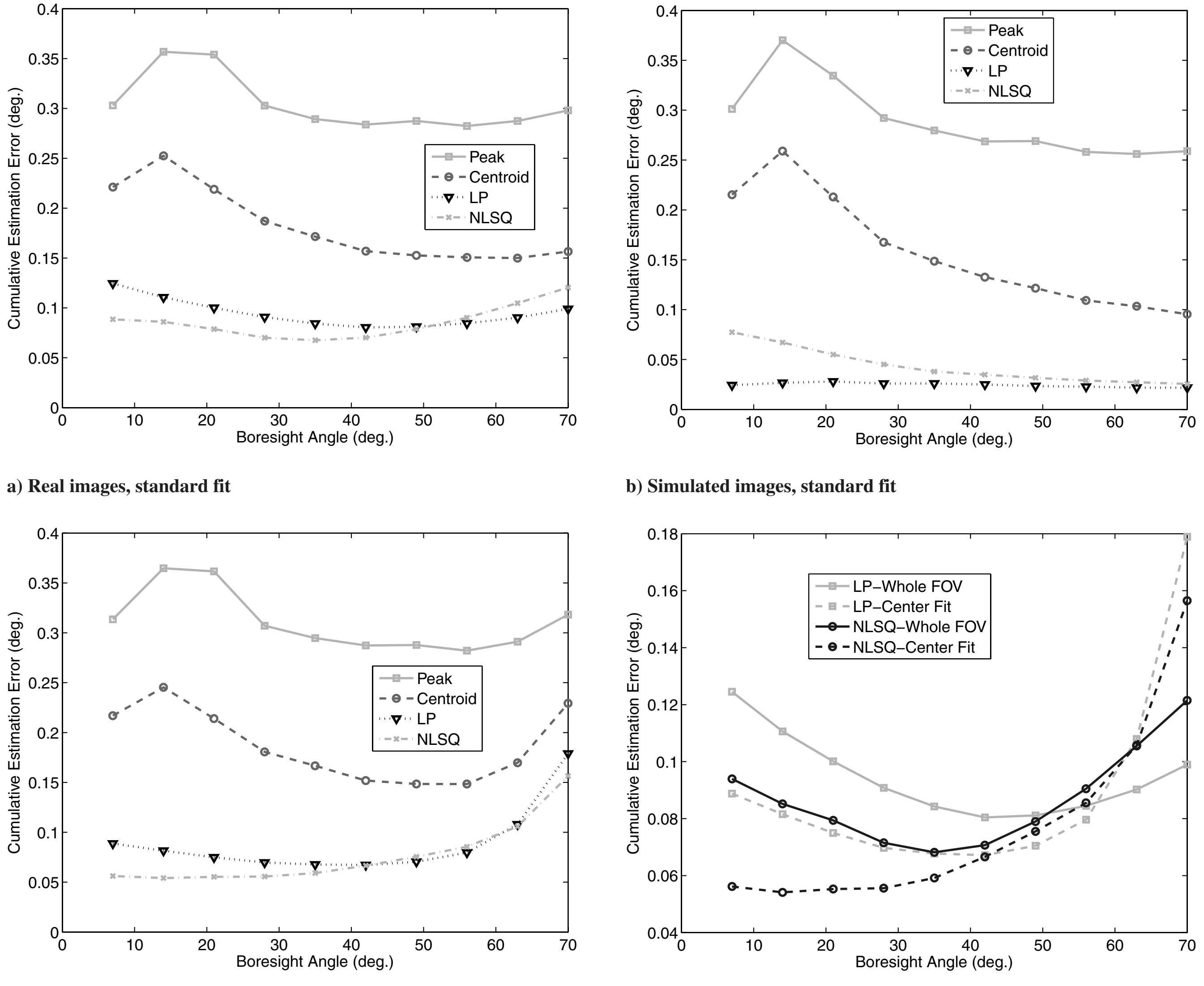
d) NLSQ, real

h) NLSQ, synthetic

Fig. 10 Algorithm performance comparison throughout the sensor's FOV (radial distances in radians).

already discussed, Figs. 10b and 10f show that the synthetic images can also reproduce the azimuthal variations seen in the real images.

Unfortunately, the correspondence between the two sets of results is not without flaws. The results from the linear phase in NLSQ algorithms show only modest correlation: trends near the center appear correct, but the results diverge near the edges of the FOV. In these regions, the simulation is not a good predictor of performance, particularly for the LP algorithm. This limits the utility of the sensor simulator as a design tool. Although it remains useful for qualitative performance predictions, any high-precision results are suspect. The calculation of simulation performance effectively ignores inaccuracies in the calibration parameters and any conceptual



a) Real images, standard fit

b) Simulated images, standard fit

c) Real images, center fit

d) Comparison, standard vs center fit

Fig. 11 Cumulative weighted error as a function of distance from the boresight.

deficiencies in the models of geometry and mounting. More study is necessary to determine augmentations to the model that may improve its fidelity.

E. Algorithm Performance

Our second stated objective in these trials is to evaluate the relative performance of the different PPE algorithms. Figure 11a shows that LP and NLSQ estimation routines outperform the peak-detection and centroiding algorithms throughout the sensor FOV. The cumulative solid-angle-weighted error shows similar performance between the two parametric routines. Over the entire FOV, the LP algorithm slightly outperforms NLSQ, but NLSQ has the advantage near the boresight. The aggregate performance over the entire trial is displayed in Table 3.

If we are willing to accept off-axis performance degradation, we can perform the sensor calibration using only samples at moderate angles. Figure 11c shows the algorithm performance when the calibration is conducted using samples collected from 0 to 45 deg. In these tests, improvements for measurements near the boresight are balanced by severe errors when the sensor moves off-axis. Figure 11d highlights the differences between the two calibration approaches. This trend gives further evidence to suggest the presence of unmodeled phenomena in the geometry and illumination models.

Overall, the parametric algorithms yield appreciable improvements in performance over conventional processing. NLSQ improves vector estimation accuracy by a factor of about two and a half and improves linear phase by more than a factor of 3. Both algorithms provide respectable increases in performance and come at the cost of a moderate increase in computational requirements.

Table 3 Comparison of PPE accuracy

Synthetic images			Real images	
	Mean error, deg	Performance index[a]	Mean error, deg	Performance index[a]
Peak Detection	0.260	1.14	0.298	1.00
Centroiding	0.085	3.49	0.156	1.90
Linear phase	0.021	13.7	0.099	3.01
Nonlinear least-squares technique	0.025	11.7	0.120	2.48

[a] Performance improvement ratio, relative to real-image peak detection.

Examining the predictions from the synthetic images, we see even more dramatic performance predictions. In simulation, LP algorithms perform more than 12 times better than the baseline algorithms. It is presently unclear whether these are merely an artifact of the simulation process or whether better geometry models would allow us to realize some of these additional performance gains.

V. Conclusions

In this study, we developed a model of digital sun sensor performance, presented several algorithms for peak-position estimation, and evaluated the performance of both our algorithms and our simulations against laboratory data. Although the simulation appears promising, some doubts remain over the simulation's ability to accurately predict quantitative performance. In contrast, we *can* conclude that the parametric algorithms evaluated in this study provide many practical benefits for digital sun sensors.

Although the initial experimental validation of the performance model left some doubts as to the precision of the model's absolute results, the approach shows promise in predicting qualitative trends and relative performance. If the quantitative accuracy of the models can be improved, the simulation can serve as a valuable design optimization tool, particularly for evaluating changes to the mask layout or physical geometry. Detailed performance modeling is certainly not a new technique, but the economics of the low-cost sensor market dictate that design optimality is rarely a technical driver. Practical, improved development tools facilitate the development of better sensors, which in turn can enable more low-cost space missions.

The proposed revisions to the SS-330 processing algorithms yielded substantial improvements over the baseline routines. Our efforts to extend parametric algorithms into two dimensions yielded good results despite the difficulties encountered reconciling laboratory results and simulation predictions. There is certainly justification for evaluating a larger range of parametric processing routines for this application. In ongoing research, we are also examining several effects that impact sensor performance: albedo rejection, motion-induced "smearing" of the images, and the precision of measuring small relative motions. The importance of these phenomena is heavily dependent on the target mission's operational concept, making it difficult to predict a sensor's effectiveness in isolation from its application. The emphasis is therefore on establishing an analysis framework appropriate to low-cost missions.

Because peak-position estimation is treated separately from sun-vector calculations, our parametric techniques should be easy to adapt to other linear array sun sensors. Star and sun sensors employing rectangular detector arrays can also benefit from these techniques. In particular, a two-dimensional detector would allow a full phase-correlation implementation: a technique that compares favorably to NLSQ estimation in our simulations.

For engineers concerned about the computational costs associated with the extra processing required by our parametric routines, we have found that integrated microcontrollers minimize any systemic impact. In a recent study with a slightly newer sensor design, the NLSQ algorithm can run on the sensor's internal processor at more than 5 Hz [19]. No external computational resources are necessary to gain this functionality. This represents an important step in establishing the viability of this technique. As low-cost, high-performance missions become more common, intelligent sensor processing may provide a convenient vehicle for increasing performance without adding to hardware costs.

Acknowledgment

The authors wish to thank Doug Sinclair from Sinclair Interplanetary for technical assistance and donation of a SS-330 sun sensor.

References

[1] Falbel, G., and Paluszek, M., "An Ultra Low Weight/Low Cost Three-Axis Attitude Readout System for Nano-Satellites," *2001 IEEE Aerospace Conference*, Vol. 5, Inst. of Electrical and Electronics Engineers, Piscataway, NJ, Mar. 2001, pp. 2469–2481. doi:10.1109/AERO.2001.931208

[2] Wu, S., and Steyn, W., "Modelling and In-Orbit Calibration Practice of a Miniature 2 Axis Analogue Sun Sensor," *Aerospace Science and Technology*, Vol. 6, No. 6, pp. 423–433. doi:10.1016/S1270-9638(02)01187-2

[3] Liebe, C., Mobasser, S., Bae, Y., Wrigley, C., Schroeder, J., and Howard, A., "Micro Sun Sensor," *2002 IEEE Aerospace Conference*, Vol. 5, Inst. of Electrical and Electronics Engineers, Piscataway, NJ, Mar. 2002, pp. 2263–2273. doi:10.1109/AERO.2002.1035393

[4] De Boom, C., and van der Heiden, N., "A Novel Digital Sun Sensor: Development and Qualification for Flight," 54th International Astronautical Congress, Bremen, Germany, International Astronautical Congress Paper 03-A.P.20, Sept.–Oct. 2003.

[5] Rufino, G., Grassi, M., and Perrotta, A., "Development and Validation of a Modern CMOS Digital Sun Sensor at UniNa," 6th Cranfield International Conference on Dynamics and Control of Systems and Structures in Space, Riomaggiore, Italy, Cranfield Univ., Paper D12-34, July 2004.

[6] Buonocore, M., Grassi, M., and Rufino, G., "APS-Based Miniature Sun Sensors for Earth Observation Nanosatellites," *Acta Astronautica*, Vol. 56, Nos. 1–2, pp. 139–145. doi:10.1016/j.actaastro.2004.09.006, Jan. 2005.

[7] Ninomiya, K., Ogawara, Y., Tsuno, K., and Akabane, S., "High Accuracy Sun Sensor Using CCDs," *AIAA Guidance, Navigation and Control Conference*, AIAA, Reston, VA, 15–17 Aug. 1988, pp. 1061–1070; also AIAA Paper 1988-4180, .

[8] Zabiyakin, A., Prasolov, V., Baklanov, A., Eltsov, A., and Shalnev, O., "Sun Sensor for Orientation and Navigation Systems of the Spacecraft," *Proceedings of SPIE: The International Society for Optical Engineering*, Vol. 3901, Oct. 1999, pp. 106–111. doi:10.1117/12.365928

[9] Enright, J., and Godard, "Advanced Sun Sensor Processing and Design for Super-Resolution Performance," *2006 IEEE Aerospace Conference* [CD-ROM], Inst. of Electrical and Electronics Engineers, Piscataway, NJ, Mar. 2006, Paper 1206.

[10] Enright, J., and Godard, "Super-Resolution Techniques for Sun Sensor Processing," *IEEE Transactions on Aerospace and Electronic Systems* (to be published).

[11] Godard, and Enright, J., "Optimization of a Sun Sensor Illumination Pattern Using Genetic Algorithms," IEEE Canadian Conference on Electrical and Computer Engineering, Inst. of Electrical and Electronics Engineers, Paper DS5-3, 7–11 May 2006.

[12] Goodman, J., *Introduction to Fourier Optics*, 2nd ed., McGraw–Hill, New York, 1996, pp. 84–85.

[13] Stone, R. C., "A Comparison of Digital Centering Algorithms," *Astronomical Journal*, Vol. 97, No. 4, 1989, pp. 1227–1237. doi:10.1086/115066

[14] Rufino, G., and Accardo, D., "Enhancement of the Centroiding Algorithm for Star Tracker Measure Refinement," *Acta Astronautica*, Vol. 53, No. 2, 2003, pp. 135–147. doi:10.1016/S0094-5765(02)00199-6

[15] Liebe, C. C., "Accuracy Performance of Star Trackers–A Tutorial," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 38, No. 2, Apr. 2002, pp. 587–599. doi:10.1109/TAES.2002.1008988

[16] Knapp, C., and Carter, G., "The Generalized Correlation Method for Estimation of Time Delay," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 24. No. 4, Aug. 1976, pp. 320–327.

[17] Foroosh, H., and Zerubia, J. B., "Extension of Phase-Correlation to Subpixel Registration," *IEEE Transactions on Image Processing*, Vol. 11, No. 3, Mar. 2002, pp. 188–200. doi:10.1109/83.988953

[18] Auer, L. H., and van Altena, W. F., "Digital Image Centering, 2," *Astronomical Journal*, Vol. 83, May 1978, pp. 531–537. doi:10.1086/112232

[19] Enright, J., and Sinclair, D., "Algorithm Enhancements for the SS-411 Digital Sun Sensor," 21st AIAA/USU Conference on Small Satellites, Utah State Univ., Paper SSC07-VIII-5, 13–16 Aug. 2007.

C. Kluever
Associate Editor