Systematic centroid error compensation for the simple Gaussian PSF in an electronic star map simulator

Wang Haiyong a, Wang Yonghai b,*, Li Zhifeng c, Song Zhenfei a

a School of Astronautics, Beihang University, Beijing 100191, China
b Beijing Institute of Space Long March Vehicle, Beijing 100076, China
c National Key Laboratory of Science and Technology on Test Physics & Numerical Mathematics, Beijing 100076, China

Received 10 July 2013; revised 20 November 2013; accepted 10 February 2014
Available online 18 March 2014

KEYWORDS
Error compensation; Image analysis; Point spread function; Star map simulation; Star sensor

1. Introduction

A star map simulator generates standard artificial star maps as inputs for testing a star sensor. Currently there are two types of star simulator, one is physical starlight simulator, and the other is electronic star map simulator (ESS). Star maps simulated by an ESS according to assigned nominal attitude can...
achieve high precision which qualifies it as an ideal reference
data source, and its capabilities of whole-celestial-sphere and
all-working-condition can fully satisfy the need of a star sensor
test, yet whose shortage is that it bypasses the optical imaging
system and goes straight to the embedded system, an incom-
plete test covering range. Therefore, it is suggested that the
complementary advantages be utilized about the two kinds of
simulator.

If an accuracy or performance test is conducted just for
some local algorithm or a single star sensor, it perhaps does
not need real-time. However, for a star sensor installed on a
satellite, ready for a system-level ground simulation, the paired
ESS should be required to match the real-time rhythm by which
the simulating system operates. Thus the completion time of a
frame of a simulated star map, starting from the receipt of a
quaternion sent by an upper PC, cannot exceed such a time
span which is equivalent to the integral time of an actual imag-
ing chip. Only by this means can the simulating similarity to the
real imaging process be guaranteed, otherwise the star sensor
will fail to receive the simulated star map in time. It is reported
that certain institution has always been executing satellite
system ground simulating tests with the lack of a star sensor,
for which the exact reason lies in the incapability of real-time
performance of the star simulator, not in the star sensor.

The loop of gray diffusion around mapped coordinates is
the most time-consuming loop of an ESS and apt to lose accu-
curacy to some extent which depends on different PSF models.
This paper will carry out the error mechanism analysis on
two typical PSFs, the integral form of PSF and the simple
PSF, and try to establish an error compensation method
for which the exact reason lies in the incapability of real-time
performance of the star simulator, not in the star sensor.

Currently, there are two typical types of PSF model, the IPSF
and the SPSF. The expression of the IPSF is as below

$$h(i, j) = \begin{cases} \frac{1}{2\pi\sigma^2} \int_{-0.5}^{0.5} \int_{-0.5}^{0.5} e^{-\left(\frac{(x-m)^2+(y-n)^2}{2\sigma^2}\right)} \, dx \, dy & \text{if } (x, y) \in D \\ 0 & \text{else} \end{cases}$$

(1)

$$g(i, j) = Ah(i, j)$$

(2)

where $(x_m, y_m)$ are the mapping coordinates, floating-point, $(i,
j)$ is any pixel within the image spot coverage, integer, $h(i, j)$ is
the energy shock response on the imaging array surface, $g(i, j)$
is the gray of pixel $(i, j)$, $D$ is the circular supporting domain of
$h(x, y)$, being centered at $(x_m, y_m)$, $\sigma$ is the Gaussian radius,
the size of the diffused image, $A$ is the energy-gray coefficient,
which is related to the total illumination, photoelectric sensi-
tivity, and integral time of the imaging spot located on the
pixel array plane.

The basic idea of the IPSF can be visualized in Fig. 3.

When the integral time has been fixed for a selected imaging
chip, in a way of the IPSF, an pixel gray is assigned by taking
the surface integral of the 2D Gaussian distributed illumina-
tion over the corresponding pixel rectangular area, which
strictly accords with the semiconductor photoelectric process of
the photo-generated charge. Just due to the perfect consist-
ency between the mathematical model and the physical pro-
cess, there is no reason not to believe that a simulated star
image spot generated by the IPSF can be regarded as an ideal
digital gray star image, which will be applied as the standard
reference data in the following theoretical analysis to the
SPSF.

2. Two PSF models: IPSF and SPSF

2.1. Star image analysis

An actual star image spot usually covers a certain pixel region,
a circle area, due to the combined effects of optics aberration
and de-focus measure. The distribution model is called PSF,
and in most cases, the energy distribution of the imaging spot
approximately accords with the two-dimensional Gaussian
distribution, as shown in Fig. 1. The following Fig. 2 shows
the space discrete effects of a digital gray star image spot on
the imaging array.

In order to obtain the simulating verisimilitude about the
space discretized digital image just like that of a CCD or
CMOS APS device, the gray diffusion also needs to be done
with each of the star image spots.

2.2. The model of IPSF

Currently, there are two typical types of PSF model, the IPSF
and the SPSF. The expression of the IPSF is as below

$$h(i, j) = \frac{1}{2\pi\sigma^2} \int_{-0.5}^{0.5} \int_{-0.5}^{0.5} e^{-\left(\frac{(x-m)^2+(y-n)^2}{2\sigma^2}\right)} \, dx \, dy$$

(1)

$$g(i, j) = Ah(i, j)$$

(2)

where $(x_m, y_m)$ are the mapping coordinates, floating-point, $(i,
j)$ is any pixel within the image spot coverage, integer, $h(i, j)$ is
the energy shock response on the imaging array surface, $g(i, j)$
is the gray of pixel $(i, j)$, $D$ is the circular supporting domain of
$h(x, y)$, being centered at $(x_m, y_m)$, $\sigma$ is the Gaussian radius,
the size of the diffused image, $A$ is the energy-gray coefficient,
which is related to the total illumination, photoelectric sensi-
tivity, and integral time of the imaging spot located on the
pixel array plane.

The basic idea of the IPSF can be visualized in Fig. 3.

When the integral time has been fixed for a selected imaging
chip, in a way of the IPSF, an pixel gray is assigned by taking
the surface integral of the 2D Gaussian distributed illumina-
tion over the corresponding pixel rectangular area, which
strictly accords with the semiconductor photoelectric process of
the photo-generated charge. Just due to the perfect consist-
ency between the mathematical model and the physical pro-
cess, there is no reason not to believe that a simulated star
image spot generated by the IPSF can be regarded as an ideal
digital gray star image, which will be applied as the standard
reference data in the following theoretical analysis to the
SPSF.
2.3. The model of SPSF

The second PSF type is the simple PSF (SPSF), whose expression is described as follows

\[ g_2(i, j) = \frac{B}{2\pi r^2} \exp \left( \frac{-(i-x_m)^2 + (j-y_m)^2}{2r^2} \right) \]

where the meanings of \((i, j), (x_m, y_m)\) and \(r\) are the same as in Eq. (1), \(B\) is the energy-gray coefficient of the SPSF, but the pixel gray assignment of the SPSF is different, derived from the directly sampled value of the Gaussian probability density function just at the pixel center, as shown in Fig. 4.

The SPSF has such advantages as simplicity and real-time due to no integration and less computing, but unfortunately its drawback is its inherent systematic error, which is related with the Gaussian radius \(r\) and the deviation of the mapping location (floating-point type) from the integer pixel center.

2.4. Real-time performance and model selection

As for real-time, it is a prerequisite to determine whether a star map simulator can be incorporated into a ground real-time simulating system to execute in-system dynamic simulation. If a process of star map simulating could finish all the computing in a period shorter than the equivalent integration time (e.g., 50 ms) of an actual imaging chip, in the end at a ready state for outputting, what the simulator achieves can be called ultra-real-time simulating, which means that it is fully capable of outputting simulated star maps with a frame frequency not lower than the actual imaging chip; similarly, the case of equal time corresponds to real-time simulating; longer computing time than the integration time corresponds to sub-real-time.

At present, an implementation of real-time or ultra-real-time simulation is not so easy because each star in a simulated star map needs to have coordinate mapping and gray diffusion done, which means a great deal of computation. According to Ref. 8, the computing time of one simulated static star map is 63 ms, already saving 1.118 s compared with the ordinary OpenGL simulating method (1.281 s), and please note that the performance level was achieved only when the new technology of GPU hardware process was adopted. But even so, the level of 63 ms is just enough for a real-time simulating requirement.

Therefore, from the perspective of time saving and based on the self-evident truth that a simple model may lead to reduction in calculation, the non-integral SPSF is usually preferred than the IPSF just due to its simplicity and can satisfy the requirement of real-time or even ultra-real-time for an ESS.

3. Error mechanism of the SPSF-simulated star image

In order to let the sets of pixel gray by the two models (IPSF and SPSF) be comparable, normalized mathematical analysis is adopted. Since the illumination and optical integration time is the same, if no model error, the gray peak value should be the same even though different models. This idea is virtually the rational basis for the normalization.

First of all, for the IPSF, a 2D Gaussian illumination function is established in such a way that it is symmetrically distributed about the center of a pixel. Let the integral value over the exact central pixel area correspond to the full-scale gray (i.e., 255 for 8 bit AD, 1023 for 10 bit AD). In this case, the central pixel integral value is defined as unit 1, namely the normalized unit 1 quantization value of the IPSF

\[ U_1 = \frac{A_{\text{max}}}{2\pi \sigma^2} \int_{-0.5}^{0.5} \int_{-0.5}^{0.5} e^{-\frac{x^2+y^2}{2\sigma^2}} \, dx \, dy \]

where \(A_{\text{max}}\) is the gray coefficient used to adjust to the full-scale gray. Next, for star imaging spots in other cases, i.e., with the Gaussian symmetric center deviating from the pixel center, or with gray not reaching the full scale, or with the combined effects of the above two, any pixel within the spot coverage can be assigned a normalized value, namely the ratio of pixel gray to \(U_1\).
g_i,j (i,j) = \frac{A}{2\pi \sigma^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{1}{2\pi \sigma^2} e^{-\frac{(x-i)^2 + (y-j)^2}{2\sigma^2}} \, dx \, dy

= \frac{A}{2\pi \sigma^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{1}{2\pi \sigma^2} e^{\frac{-r^2}{2\sigma^2}} \, dx \, dy

= \frac{A_\text{max}}{2\pi \sigma^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{1}{2\pi \sigma^2} e^{\frac{-r^2}{2\sigma^2}} \, dx \, dy

(5)

where the central integer coordinates (i_c, j_c) are acquired by a round-off of the star mapping coordinates (x_m, y_m) of floating-point type. Hence, x_m = i_c + \Delta x_m, y_m = j_c + \Delta y_m, unit: pixel. Eq. (5) can be disassembled into two parts according to the independence of x and y of a 2D Gaussian distribution function.

Secondly, to set up the normalized method of the SPSF, the quantized value of unit 1 is defined as the full-scale gray, which is transformed from the maximum value of the Gaussian probability density function of illumination. If the Gaussian distribution is symmetrical about a pixel, the normalized value corresponds to the peak value sampled exactly at the center of the central pixel

U_s = \frac{B_{\text{max}}}{2\pi \sigma^2}

(6)

where B_{\text{max}} is the gray-energy coefficient of the SPSF. Next, the normalized gray value of an arbitrary pixel within the coverage of a star image spot is

g_i,j = \frac{A}{2\pi \sigma^2} \frac{1}{2\pi \sigma^2} e^{-\frac{(x-i)^2 + (y-j)^2}{2\sigma^2}} \cdot U_s

= \frac{B_{\text{max}}}{2\pi \sigma^2} \frac{1}{2\pi \sigma^2} e^{-\frac{(x-i)^2 + (y-j)^2}{2\sigma^2}} \cdot C

(7)

Let the equation A/A_{\text{max}} = B/B_{\text{max}} be true irrespective of the model difference, even so, the normalized gray values obtained by the IPSF and the SPSF respectively are still inconsistent, for which the exact reason lies not in B/B_{\text{max}} but in the central sampling coordinates (i, j) of Eq. (7). It is indeed just like that, and some adverse consequences occur. Firstly, two sets of assigned gray values within the coverage of the star image spot are different between the two models; secondly, the ultimate calculated centroid of the SPSF-simulated star image spot will deviate from the ideal mapped central coordinates.

The error mechanism is that the SPSF-assigned pixel gray is derived from the sampled value of the 2D Gaussian probability density function just at the center of the pixel (an unreasonable sampling location), in addition that a Gaussian surface is nonlinear, resulting in the deviation from the ideal mapped central coordinates.

4. Modeling for the error compensated SPSF

According to the above analysis about Eq. (7), some compensation measures should be adopted to equate the normalization value by the SPSF to that of the IPSF, fully conforming to the assigned gray distribution of the SPSF-simulated star image spot, which possesses perfect similarity to a real shot star image because of the completeness of the IPSF model. Out of this idea, Eq. (7) is modified as follows

\begin{align*}
g_s(i,j) &= \frac{B}{B_{\text{max}}} \cdot \frac{1}{2\pi \sigma^2} e^{-\frac{(x-i)^2 + (y-j)^2}{2\sigma^2}} \cdot C
\end{align*}

(8)

where two offset items \Delta_i and \Delta_j are inserted to the object pixel (i, j), which are the central coordinates of the object pixel, integer type. As a result, the normalized gray of pixel (i, j) is evaluated according to the sampled value of the 2D Gaussian function at position (i + \Delta_i, j + \Delta_j), not the central position (i, j), just aiming to acquire equal normalization values between the two models. The task of error compensation modeling is virtually to set up a function expression of the offset items (\Delta_i, \Delta_j) with regard to the Gaussian radius \sigma and the distance (i - x_m, j - y_m) from the object pixel center to the mapping location.

Since this is a modeling process, and considering that A/A_{\text{max}} = B/B_{\text{max}}, let the corresponding parts of the two Eqs. (5) and (8) be equal to each other

\begin{align*}
&\left\{ \begin{array}{l}
e^{-\frac{(x-i)^2 + (y-j)^2}{2\sigma^2}} = \frac{1}{2\pi \sigma^2} e^{-\frac{(x-i)^2}{2\sigma^2}} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-\frac{r^2}{2\sigma^2}} \, dx \, dy \\
e^{-\frac{(x-i)^2 + (y-j)^2}{2\sigma^2}} = \frac{1}{2\pi \sigma^2} e^{-\frac{(x-i)^2}{2\sigma^2}} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-\frac{r^2}{2\sigma^2}} \, dx \, dy
\end{array} \right.
\end{align*}

(9)

The right-hand parts of Eq. (9) are written respectively as \(R_i(i - x_m, \sigma)\) and \(R_j(j - y_m, \sigma)\), and then the above Eq. (9) can be transformed into

\begin{align*}
\Delta_i &= \begin{cases} 
-(i - x_m) + \sqrt{-2\sigma^2 \ln R_i(i - x_m, \sigma)} & \text{if } i - x_m \geq 0 \\
-(i - x_m) - \sqrt{-2\sigma^2 \ln R_i(i - x_m, \sigma)} & \text{if } i - x_m < 0
\end{cases}
\end{align*}

(10)

\begin{align*}
\Delta_j &= \begin{cases} 
-(j - y_m) + \sqrt{-2\sigma^2 \ln R_j(i - x_m, \sigma)} & \text{if } j - y_m \geq 0 \\
-(j - y_m) - \sqrt{-2\sigma^2 \ln R_j(i - x_m, \sigma)} & \text{if } j - y_m < 0
\end{cases}
\end{align*}

(11)

where the above two expressions about rows and columns respectively are apparently of the same appearance, therefore the simulation test and analysis can be carried out just by selecting the row function \(\Delta_i = f(i - x_m, \sigma)\), whose parameter fitting process wholly applies to the column function \(\Delta_j\).

Set the pixel coordinates (i_c, j_c) to (0, 0), analyzing from -6 to 6 altogether 13 pixels on both sides of the origin 0 along the row direction. 10 offset values, which represent the deviation values of i - x_m, are sampled at a step of 0.1 pixel in the interval [-0.5, 0.5] pixel. Then, the function values of \(\Delta_i\) with regard to i - x_m and \(\sigma\) have such a 3D distribution figure as in Fig. 5.

A family of function curves about \(\Delta_i\) and i - x_m under different \(\sigma\) values can also be obtained as shown in Fig. 6.

For Nos. (-1) and (-2) pixels, when the mapping deviation \(\Delta_m\) equals to 0.2 pixel, the variation curves of function \(\Delta_i\) against the Gaussian radius \(\sigma\) are plotted in Fig. 7.

Based on the simulated data of Eq. (10), function \(\Delta_i\) can be surface fitted. The kernel functions about variable x (representing i - x_m) may be selected as x and x^2 because of its odd symmetry about the origin. The other kernel functions about variable \(\sigma\) can be selected as 1, \sigma, \sigma^2.

As a result of least square surface fitting with 3D data (i - x_m, \(\sigma\), \(\Delta_i\)), the coefficient matrix is calculated as follows...
The final fitting equation of $D_i$ is got as below

$$D_i = \left[ x, x^3 \right] C \begin{bmatrix} 1 \\ \sigma \\ \sigma^2 \end{bmatrix}$$

where $x$ represents the item $(i - x_m)$.

After comparison analysis of the two computation results from Eqs. (10) and (13), the global fitting accuracy about $\Delta_i$ is

$$\sigma_{\Delta_i} = \sqrt{\frac{1}{17 \times 130} \sum_{k=1}^{17} \sum_{j=1}^{130} \left[ \Delta_k \left( (i - x_m), \sigma_k \right) - \Delta_k \right]^2}$$

$$= 0.0019 \text{ pixel}$$

(14)

For the particular case of $\sigma = 0.671$ pixel, characterized by that more than 95% of star image energy is distributed in a $3 \times 3$ pixels coverage, the necessary pixels to be treated are at least $4 \times 4$ pixels. In the domain of $(i - x_m) \in [-4.4, 4.5]$, the probable location range of the effective distance from the object pixel center to the mapping $x$ coordinate, its 1D local fitting accuracy of Eq. (13) under the case of $\sigma = 0.671$ is obtained as below

$$\sigma'_{\Delta_i} = \sqrt{\frac{1}{90} \sum_{j=21}^{130} \Delta_k \left( (i - x_m), \sigma_{0.671} \right) - \Delta_k}^2 = 0.0018 \text{ pixel}$$

(15)

Under the cases of $\sigma = 0.5, 0.7, 0.9$, each of the functional curves and its corresponding fitted curve are plotted respectively in Fig. 8.

Because of the equivalence of $x$ and $y$ axes of the 2D Gaussian distribution, the coefficient matrix $C$ in Eq. (13) can be directly copied to the function expression $D_j$. Finally, the values of $\Delta_i$ and $\Delta_j$ based on Eq. (13) are plugged into Eq. (8), which fulfills the gray error compensation for the SPSF.

5. Simulating test for the compensated SPSF

In view of the self-evident truth that a simple model is bound to boost the simulating speed, so real-time performance is not the test task here, and only the item of precision is chosen and will be tested in the following procedures. As for the selection of a test tool, there are no better verifying tools than the simulating tool. For a real shot star image, there is no way to obtain its ideal centroid and no way to conduct an error analysis.

5.1. Similarity improvement test

One cannot tell tiny gray differences between image pictures by naked eyes, so it is more meaningful to give the gray data for comparison than to list the star images simulated by the three models of the SPSF, the compensated SPSF, and the IPSF.
The assigned gray data of the star image spot in Fig. 9 is shown in the 3rd row of Table 1.

The three rows of gray data respectively simulated by the three models are obtained under the same conditions that are $\sigma = 0.5$ pixel, $\Delta x_m = 0.25$ pixel, and $\Delta y_m = 0$ pixel. Their central row data distribution of the above three star images are plotted in Fig. 10.

The tendency is clearly shown that the gray data of the compensated SPSF is nearer to those of the IPSF than those of the primitive SPSF. Choosing the correlating coefficient as the index of model similarity, the test results are plotted in Fig. 11.

After error compensation processing, the lowest correlated coefficient of 0.9858 between the SPSF and the IPSF is increased to 0.9987 (also the lowest value) for the comparison of the compensated SPSF and the IPSF, even reaching such a highest correlated coefficient of 0.9993.

As far as the simulating similarity is concerned, striving for excellence is not unnecessary though the correlated coefficient of 0.9858 is also satisfying, not forgetting that the role taken by a star map simulator is virtually a standard test instrument and the rear star sensor algorithms endeavor to reach a sub-pixel centroid precision.

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Assigned gray data comparison.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model type</td>
<td>Gray data simulated by the three models</td>
</tr>
<tr>
<td>SPSF</td>
<td>0 0 0 0 0</td>
</tr>
<tr>
<td></td>
<td>0 1 24 9 0</td>
</tr>
<tr>
<td></td>
<td>0 9 180 66 0</td>
</tr>
<tr>
<td></td>
<td>0 1 24 9 0</td>
</tr>
<tr>
<td></td>
<td>0 0 0 0 0</td>
</tr>
<tr>
<td>Compensated SPSF</td>
<td>0 0 0 0 0</td>
</tr>
<tr>
<td></td>
<td>0 3 37 17 0</td>
</tr>
<tr>
<td></td>
<td>0 17 185 83 1</td>
</tr>
<tr>
<td></td>
<td>0 3 37 17 0</td>
</tr>
<tr>
<td></td>
<td>0 0 0 0 0</td>
</tr>
<tr>
<td>IPSF</td>
<td>0 0 0 0 0</td>
</tr>
<tr>
<td></td>
<td>0 5 43 21 0</td>
</tr>
<tr>
<td></td>
<td>0 20 187 90 2</td>
</tr>
<tr>
<td></td>
<td>0 5 43 21 0</td>
</tr>
<tr>
<td></td>
<td>0 0 0 0 0</td>
</tr>
</tbody>
</table>

5.2. Centroid errors of the SPSF-simulated star image spots

For the SPSF-simulated star image spots with no noise added, the curves of centroid error $\Delta x$ with the gray weighted method are plotted in Fig. 12, where the mapping positions serve as ideal values.
It can be seen that the centroid error eminently increases as the Gaussian radius $r$ decreases. When $r = 0.5$ pixel, the maximum absolute error reaches up to 0.0226 pixel at $\Delta x_m = \pm 0.25$ pixel, larger than 1/50 pixel, a non-negligible level, but when $r = 0.671$, the maximum absolute error is merely 0.0008 pixel, that is to say, if the Gaussian radius $r$ of the simulated star image spot is greater than 0.671 pixel, there is no need for the SPSF to adopt a compensation measure, because it does not matter to use it directly; if not, that turns to be necessary.

5.3. Centroid error of the IPSF-simulated star image spots

The absolute centroid error curves of the IPSF-simulated star image spots are shown in Fig. 13.

Though it is similar to that of the SPSF in Fig. 12 by outward appearance, its maximum absolute error is merely 0.0023 pixel at $\Delta x_m = \pm 0.25$ pixel when $\sigma = 0.5$ pixel, an order of error magnitude far lower than that of the SPSF, which is why the offset compensation method for the SPSF is set up through numerical fitting in reference to the gray values generated by the IPSF.

5.4. Centroid error of the compensated SPSF simulated star image spots

With no noise added to star image spots simulated by the compensated SPSF, absolute centroid error curves with the gray weighted method are plotted in Fig. 14.

In comparison with Figs. 13 and 14, all of them reach up to the same $10^{-3}$ pixel order of magnitude about their absolute maximum centroid errors, as well as with similar forms. Furthermore, compared with centroid error of the SPSF shown in Fig. 12, under the same condition of $\sigma = 0.5$ and at $\Delta x_m = \pm 0.25$ pixel, the absolute maximum centroid error of the compensated SPSF in Fig. 14 is only 0.008 pixel, a 2.83-times accuracy in comparison to the primitive SPSF, which manifests the validity of the offset compensated method for the SPSF.

5.5. Real-time performance tests

After observing qualitatively the three expressions of the PSF, namely the IPSF expressed by Eqs. (1) and (2), the SPSF by Eq. (6), and the compensated SPSF by Eq. (8), such a judgment can be made that the computational time of the IPSF has to be far more than those of the SPSFs because of its involved integral function, and that the compensated SPSF will also consume a bit more time than the primitive SPSF just for the extra computation about $D_i$ and $D_j$ expressed by Eqs. (10) and (11).

The real-time performance comparison tests were conducted under the following hardware and software simulating circumstances: Intel Duo CPU P8600@2.40 GHz, Win7 64 bit, and Matlab 2010a. A pair of mapped coordinates on the imaging array were assumed at (200.45, 200.45), around which gray diffusion was executed according to the 3 PSFs to construct a simulated star image, and 100 times of repetitive computation were operated for each PSF in order to reach an equivalent amount of computation as that of a simulated star map which contained 100 stars inside. The elapsed time was counted respectively for the three cases of PSF, as shown in Table 2.

As expected, the consumed time of the IPSF (mean value of 14.3 s) is 2 orders of magnitude more than the latter two SPSFs, which simply cannot satisfy the requirement of real-time simulating tests. The involved double integral function (dblquad() in the Matlab program) actually uses the method of recursive adaptive Simpson quadrature which is doomed to consume more computations.

<table>
<thead>
<tr>
<th>Model type</th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
<th>4th</th>
<th>5th</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPSF(s)</td>
<td>14.27</td>
<td>14.32</td>
<td>14.37</td>
<td>14.30</td>
<td>14.21</td>
<td>14.30</td>
</tr>
<tr>
<td>SPSF(ms)</td>
<td>54.3</td>
<td>53.6</td>
<td>53.5</td>
<td>53.4</td>
<td>53.1</td>
<td>53.6</td>
</tr>
<tr>
<td>Compensated SPSF(ms)</td>
<td>61.7</td>
<td>62.4</td>
<td>61.7</td>
<td>61.4</td>
<td>61.8</td>
<td>61.8</td>
</tr>
</tbody>
</table>
As for the compensated SPSF, its consumed time has a small difference of 8.2 ms by reference to the time of the primitive SPSF, just caused by the extra computation about $\Delta_i$ and $\Delta_j$, yet the final better centroiding precision is worthy of that exchange of time price. To a star sensor mounted in a satellite ground real-time simulating test system, its paired electronic star map simulator must keep up with data updating frequency, and this level of time consumed by the SPSF or the compensated one is sufficient to meet with the demand for the output rate (higher than 10 Hz in general).

6. Conclusions

(1) The SPSF has such advantages as simplicity and fewer computations, and therefore good real-time, but has inherent systematic error. Its error mechanism is that the SPSF-assigned pixel gray derives from the sampled value of the 2D Gaussian probability density function just at the center of the pixel (an unreasonable position), in addition to that the surface of a Gaussian distribution is nonlinear, which results in a deviation from the ideal predefined mapping coordinates.

(2) In reference to the IPSF-simulated star image, the offset functions $\Delta_i$ and $\Delta_j$ from the pixel center are got, which are of the two variables $i - x_m$ and $\sigma$, and subsequently the systematic error compensation for the SPSF is executed by substituting the pixel central position $(i, j)$ with the offset position $(i + \Delta_i, j + \Delta_j)$. In the simulation tests, for the big error case of $\sigma = 0.5$ pixel, the compensated SPSF achieves an improved similarity, almost wholly compensated for the centroid systematic error (the maximum error is 0.0226 pixel, larger than 1/50 pixel) of the primitive SPSF, reaching a 0.008 pixel maximum error level, a 2.83-times precision in comparison to the primitive SPSF.

(3) The lower the $\sigma$ value, the worse situation of the systematic error, so it is suggested that the compensation measure for the SPSF be adopted when $\sigma < 0.671$ pixel, in such a manner the standardability of star map outputs of an electronic or software simulator can be insured. However, when $\sigma > 0.671$ pixel (a dividing value for reference only), the compensating effects are not that prominent.

References


Wang Haiyong received his Ph.D. degree in the speciality of Precision Instruments and Machinery in 2004 from Beihang University, Beijing, China, a lecturer for the course of Celestial Navigation Basis in the School of Astronautics, Beihang University. More than 30 acadamic papers have been published by him as the 1st author, and 2 invention patents were granted as the 1st inventor. His research interests include: celestial body sensor, celestial navigation and integrated navigation system.

Wang Yonghai received his M.S. degree in 2003 in Automatic and Electronic Engineering from Beihang University, Beijing, China. He is currently a senior engineer with Beijing Institute of Space Long March Vehicle, China. His research interests include numerical modeling, design of circuits, and optoelectronic system.