A fast algorithm of simulating star map for star sensor

Chao Li, Changwen Zheng, Jiaze Wu, Liqiang Zhang
National Key Laboratory of Integrated Information System Technology
Institute of Software, Chinese Academy of Sciences
Beijing, China
happysuperlee@163.com

***Abstract*—In order to test the function and performance of star sensor on the ground, a fast method for simulating star map is presented. The algorithm adopts instantanesous coordinate of star and improves the star searching efficiency by optimizing the zone partitioning method for star catalogue. We overcome the low accuracy of the latitude and longitude's span that FOV overlays by proposing a new spherical right-angled triangle method and the searching scope is reduced highly; meanwhile, the simulation model for star brightness is also built based on adopted star catalogue. Simulation study is conducted for the demonstration of the algorithm. The proposed approach meets the requirement of wide magnitude range and short simulation period.**

Keywords-simulated star map; zone partitioning; spherical triangle; FOV

I. INTRODUCTION

Star sensor is an very important instrument of attitude determination and guiding system on satellite [1], which primarily uses star map for real-time attitude adjustment. In the testing of star sensor's function and performance, it needs the star simulator which provides the simulation of star map in the field of view (FOV) under any time and any attitude. With the development of star sensor's technology, the demand for refresh rate and response time becomes higher, efficient simulation algorithm of star map is key to implement real-time star simulator.

Computer simulation of star map first loads star catalogue file, then finds the stars that fall into the star sensor's FOV centered by the optical axis from star catalogue, then makes the transformation of stars' coordinate and magnitude, and display them in the form of two-dimensional image. In order to accelerate the simulation of star map, many methods have been proposed. Reference [2] proposes a matrix structure of celestial sphere, which better stores the guide star catalogue for star sensor; Reference [3] adopts a cone method to partition star catalogue while designing star tracker; But they just involved with the method of improving the structure of star catalogue. Reference [4] proposes a star map simulation method, but inaccurate calculation of ascension span in searching stars reduces the accuracy of the simulation result. Reference [5] addresses a new star retrieval method, but the simulation efficiency isn't satisfying. This paper proposes a fast simulation method. By optimizing star catalogue structure, the search speed improves obviously. In order to solve the inaccurate calculation of ascension span, a new spherical right-angled triangle method is introduced. Moreover, the simulation model of star brightness in FOV is also generated.

II. FAST ALGORITHM OF SIMULATION

Simulated star map consists of the stars extracted from the catalogue of stars which fall into the FOV. So star catalogue is the basis of the simulation. One direct method to search star is traversing the whole catalogue, checking whether the stars fall into the FOV. But the catalogue contains a huge numbers of stars, obviously this method is inefficient. To solve it, the catalogue structure needs partitioning into zones, a proper partition size of each zone will be discussed, which makes the searching and access of star catalogue more efficiently.

Base on star catalogue, star map simulation computes the spherical span of FOV and finds the zones it overlays according to optical axis of FOV; then searches stars which fall into the CCD image plane of FOV from star zones; finally, convert the star brightness into corresponding pixel gray of image.

A. The partition of star catalogue

So far, the partition methods of star catalogue's structure can be grouped into two categories: even division by the average block and division by latitude and longitude.

The method of even division by the average block divides the surface of celestial sphere into N blocks evenly, every block indicates a interval. When the vector of optical axis is given, the index of corresponding block the axis points is found, and the search scope is the 8 adjacent blocks centered by it. Therefore, the search scope is reduced to the $9/(N \times N)$ of initial scope. However, the complexity of computing the boundary of each block is high [6], the time-consuming will not significantly decrease when the number of stars is huge.

Our partition method is based on another method, which divides the celestial sphere by latitude and longitude interval. Similar to earth circle of latitude and longitude, the celestial sphere is divided into non-overlapping zones by ascension and declination circle, stars in each zone are stored respectively, and coded according to the interval of ascension and declination the zone stays. The representation of this method's zone boundary is consistent with star coordinate, which is computed more easily and quickly.

As to the size of each zone, it influences the efficiency of star searching and can not be too big or too small. Here we adopt different partition means according to different size of FOV. For larger FOV, the size can take slightly larger than half the FOV; for smaller FOV, if the size is too small, excessive zones will affect the efficiency of storage and search, so a proper size can be assigned to make moderate number of zones.

In order to reduce the storage files, a combined storage method is proposed here. It means that the zones of the same declination interval can be stored into one storage file according to the order of increasing right ascension. And a index file for each storage file is created that records the ascension interval, the first address and offset length of each zone. By the combining method, we needn't to create a separate file for each zone, so the number of storage files can be reduced greatly, and the efficiency of search can improve consequently.

Based on zone partition method, a test star catalogue is created. The start point is south pole, the declination ranges from 0-180 degrees, the ascension in 0-360 degrees, and the zone's size is 1.5. Totally 120 star storage files and index files are created, and the storage format is shown as follows.

TABLE I. THE STRUCTURE OF STAR CATALOGUE

Longitude range	Star storage file	Index file
0~1.5	Zone0000.cat	Zone0000.acc
1.5~3.0	Zone0015.cat	Zone0015.acc
…	…	…
177.0-178.5	Zone1770.cat	Zone1770.acc
178.5-180.0	Zone1785.cat	Zone1785.acc

The storage file contains stars of corresponding declination range and stores in ascending order of ascension. The index file provides the ascension index of each zone in the storage file by three fields<index, start address, length>.

The basic star catalogue stores star serial number, star magnitude, star ascension and declination coordinate value in J2000 epoch. However, the simulation of star map needs the star coordinate in simulation time. So the coordinate in J2000 needs star position procession, nutation and other astronomical amendments. Correction methods have been discussed a lot in spherical astronomy. The computation method offers in Reference [4].

B. *Compute the zones the FOV overlays*

As the star catalogue is divided into different star storage files of zones, the FOV centered by the optical axis needs to find the zones it covers, thus the search scope will be reduced to the zones instead of whole sphere. Noted that ascension arc on the celestial sphere gradually shortens with the increasing declination, when the optical axis points to different declination interval, same FOV covers different numbers of star zones. Actually, the number of zones the same FOV covers reduces orderly as the optical axis points from high-declination section to low-declination section. Thus we can not simply take the fixed adjacent zones around the optical axis as the search scope.

For the problem of different covered zones with optical axis in different declination interval, several methods have been proposed. Reference [7] transforms high latitudes to low latitudes by setting the convention of two coordinate systems when FOV is in high-latitude, but the transformation matrix is complex, and also needs a additional star database in spherical coordinates. Reference [8] introduces a expansion factor to compute the approximate boundary range as it varies with different latitudes, but this method isn't very precise. In this paper, we propose a new method to construct a spherical triangle, which gives the accurate spherical range of FOV.

Considering the vertical distance between observation point and the center of earth is extremely small compared to the distance with the center of star, we take the geocentric as observation center. (see Fig.1).

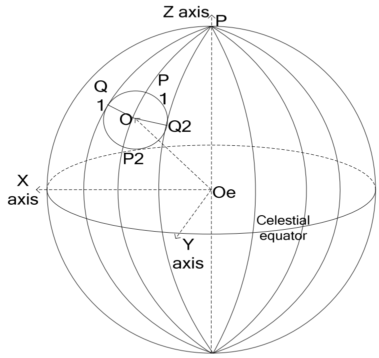

Figure 1. The range of FOV in celestial sphere.

Oe is observation center, O is intersection point between view axis and spherical surface. The spherical meridian passing point O intersects with FOV at P_1, P_2 points. FOV and two tangent meridian cut at Q_1, Q_2 points. Now the range of latitude can be determined by computing the latitude of P_1, P_2 and the range of longitude determined by computing the longitude of Q_1, Q_2. As to the range of latitude, first compute the latitude and longitude of O, then the upper latitude is solved by adding half angle of the FOV to the latitude of O and lower latitude by subtracting half angle of the FOV from the latitude of O. For the latitude and longitude of Q_1, Q_2, we compute it by solving spherical right triangle PQ_1O, PQ_2O. Take triangle PQ_2O for example, according to spherical geometry, equation can be

$$\begin{cases} \sin\alpha_2 = \cos\theta\cos\alpha_1 + \sin\theta\cos\alpha_1\sin\triangle\delta \\ \sin\triangle\delta\cos\alpha_2 = \cos\triangle\delta\sin\theta \\ \cos\triangle\delta\cos\alpha_2 = \cos\alpha_1\cos\theta - \sin\theta\sin\alpha_1\sin\triangle\delta \end{cases}$$

In the equation, θ is the half angle of FOV, α_1、α_2 is the latitude of O and Q_2, $\triangle\delta$ is the longitude difference between Q_2 and O. The value of θ and α_1 is given, and the α_2 and $\triangle\delta$ can be solved.

When the upper latitude exceeds 90 or lower latitude is less than -90, it indicates that the FOV covers the celestial pole and across the polar circle. To solve this, we segment the FOV by separating latitude range it covers, compute the range of longitude in every segment respectively, and the method is same as above. Thus, after computing the spherical range of FOV, the zone that FOV overlays can be obtained according to each zone's longitude and latitude boundary in star catalogue.

By constructing spherical right triangle, we can compute the accurate spherical range of FOV with the view axis pointing to any part of the celestial sphere. It need not the complex classification in different parts, and is easy to understand.

C. Search Stars

After computing the zones that FOV covers, the search scope is reduced to these zones instead of the whole celestial sphere.

In star sensor's FOV, the stars finally project into rectangle CCD image plane. The circular FOV is the circumcircle of CCD plane, the diagonal of the plane equals the diameter of circular FOV, and the circular FOV can be obtained by CCD's FOV.

As it is easier to seach stars in circular FOV than CCD image plane, and avoid complex computation of checking every star in zones whether projecting into image plane, we first retrieve the stars in circular FOV. So, the search process can be divided into two parts: first search stars in circular FOV by computing the vector angle between star and view axis; then, coordinate transformation is processed for each star in circular FOV, and the image plane 's (x,y) coordinate is computed, if the star's coordinate is beyond the range of the image plane, it won’t project into image plane. Based on the two steps, the star searching process is completed. During the searching, the star in celestial coordinate should be projected into CCD image plane coordinate system, the coordination transformation has been discussed a lot, Reference [4] gives detailed computation methods.

D. The simulation of star brightness

In order to simulate a star map, the star’s brightness simulation is indispensable in addition to the CCD image plane location of simulated stars. So we construct a model for computing the grey of pixel corresponding with each star in image plane. The grey of star in star map mainly correlates with its own brightness and star sensor’s exposure time. The brightness of star is reflected in the star apparent magnitude. In the star catalogue of USNO-A, every star stores two aspects infomation of magnitude: red magnitude (Rmag) and blue magnitude (Bmag). The apparent magnitude is calculated by variables Rmag and Bmag, the fomula is:

$$m = \begin{cases} Bmag/10.0-1 & Rmag = 999 \\ 0 & Rmag < 0 \\ Rmag/10.0 & others \end{cases}$$

The brightness is higher when the magnitude is smaller, Reference [9] gives the relations of magnitude and grey

$$g = 255/2.51^{m-m_s}$$

g indicates grey, range from 0-255; m represents magnitude of star, m_s represents the biggest magnitude that star sensor can detect. But the formula doesn’t consider the exposure time of star sensor.

Actually, with the exposure time becoming longer, star sensor receives more more radiation energy, thus the image brightness is larger. As the exposure time varies with different star sensor, here exposure coefficient K is used to represent the positive relationship between image gray and exposure time, the computing formula is

$$g = k * 255/2.51^{m-m_s}$$

However, in real starspace, various noise sources exists. In order to simulate more realistic star map, we reflect the noise error on the star brightness. And Gauss noise is introduced to simulate the influence of the stray light, radiation and other factors on star brightness. Then we get the formula as

$$g = k * 255/2.51^{m-m_s+m_\delta}$$

Here m_δ is the Gausss noise with average as zero and variance as $\delta_m{}^2$.

Based on CCD imaging theory, the presented grey model of star accurately simulates the star brightness. The following simulation study gives the data analyst of simulated star brightness.

III. SIMULATION STUDY

In order to verify the effectivity of the proposed algorithm, computer-based simulation evironment is set up. We adapt Intel Xeon X5450 CPU, 3.00GHZ; 16GB memory; Microsoft Visual Studio 2003 development environment. Simulated star sensor’s FOV is 2.5*2.5, the size of CCD image plane is 1024*1024, and the size of each pixel is 13 um*13 um; the biggest identified star magnitude is 15.0. Among the common star catalogue, USNO star catalogue collects more star infomation and has better distribution density [10]. The simulated star catalogue is extracted from the USNO-A V1.0 by selecting the stars with star magnitude under 15.0.

We use MonteCarlo method to randomly generate the view axis vector of star sensor which is (212.899993, -7.800001). The simulation snap time is at 0:00 AM, May 12, 2010. Totally 2066 stars that project into FOV are selected. And the simulation of star location is processed. Here three coordinates of selected star are generated: celestial coordinate in J2000 epoch, the instantaneous coordinate in simulation time and image plane coordinate(x,y). The relationship is shown as follows.

TABLE II. THE RESULT OF STAR LOCATION SIMULATION

Star magnitude	J2000 epoch		Simulation Time		x	y
	longtitude	latitude	longtitude	longtitude		
6.2	211.574016	-8.891292	211.720515	-8.943857	77	1007
7.7	211.783549	-8.439573	211.929830	-8.492009	146	819
8.4	211.783599	-8.435493	211.929878	-8.487928	146	817
7.4	212.156960	-8.862006	212.303543	-8.914248	307	977
8.4	212.264073	-8.004229	212.410187	-8.056385	323	629
8.5	212.430787	-7.819451	212.576821	-7.871509	384	550
8.5	212.520173	-8.371890	212.666537	-8.423914	436	769
7.8	213.321936	-8.669081	213.468605	-8.720659	764	865
...	...	...	...	...	...	...

We simulate the brightness of stars by setting the Gauss noise of average value as 0 and variance as 0.01, the exposure factor as 1.0. The grey histogram of the simulated star map shows as figure 2.

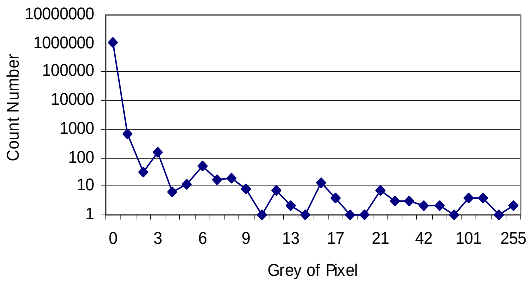

Figure 2. The distribution analysis of image grey in star brightness simulation

MonteCarlo method [11-14] is implemented to randomly generate 1000 different view axis vectors of FOV, the average time of selecting star is controlled under 5ms, which meets the requirements of high refreshing frequency of performance test for star sensor. Times of simulation also shows the high robustness of the presented approach.

IV. CONCLUSION

In this paper, we have proposed a fast simulation method of star map for star sensor. This method selects stars in FOV more quickly, and has a high real-time simulation speed. It optimizes the storage structure of star catalogue. The searching scope can be computed accurately by the new spherical right triangle method. This approach also introduces realistic star brightness simulation model. The simulation experiment proves the high robustness and effectivity of the approach.

ACKNOWLEDGMENT

This project was supported by a grant from the National High Technology Research and Development Program of China (863 Program) (No. 2009AA01Z303).

REFERENCES

[1] Carl Christian LieBe, “Star Trackers for Attitude Determination.,” IEEE AES Systems Magazine, vol. 33, 1995, pp. 10-16.

[2] Bone Jeffery W, “On-orbit star processing using multi-star trackers.” Proc.SPIE, vol. 22, 1994, pp. 6-8.

[3] Ju G, Kim H P T, Junkins J L, “DIGISTAR: a low-cost micro star tracker,” Proceedings of AIAA Space Technology Conference & Exposition, Albuquerque, 1999, pp. 3599-4603.

[4] Zhang Jun ping, Lin Tao, Zhou Jian lin et al, “A method of CCD star image simulation,” Chinese Space Science And Technology. vol. 19, 1999, pp. 46-50.

[5] Hye-Young KIM, John L. JUNKINS, “Self-organizing guide star selection algorithm for star trackers: thinning method,” Proceedings of 2002 IEEE Aerospace Conference Big Sky, Montana, USA IEEE 2002, pp. 2275-2283.

[6] Song Yijun, Yang Gelan, Tian Zunhua, “Research and implementation of space scene modeling,” computer simulation, vol. 27, 2010, pp. 40-42.

[7] Rao Caijie, Fang Jiancheng, “A way of extracting observed stars for star image simulation,” Optics and Precision Engineering, vol. 22, 2004, pp. 129-134

[8] Cai Zhiwu, Han Chunhao, Chen Jinping, “A approach of constructing high precision guide star catalogue on board,” Journal of Geomantics Science and Technology, vol. 23, 2006, pp. 29-32.

[9] Zhao Ming bo, Liu Yu, Tao Zheng yu, He Yong jun, “One method on star image simulation of airborne astronomical navigation,” Optoelectironic Technology, vol. 28, 2008, pp. 185-187.

[10] http://archive.eso.org/skycat/usno.html

[11] Lubritto, C., Petraglia, A., Vetromile, C.et al, “Simulation analysis and test study of BTS power saving techniques,” Proceedings of 31st International Telecommunications Energy Conference, Incheon, 2009, pp. 1-4.

[12] Li Zhonghua, Huang Zhipeng, Guo Lei et al, “The mesoscopic scale FEM simulation of dielectric properties of composite material based on montecarlo,” Proceedings of IEEE 9th International Conference on the Properties and Applications of Dielectric Materials, Harbin, 2009, pp. 1141-1144.

[13] Castillo, J. Bosque, J.L. Castillo et al, “Hardware accelerated montecarlo financial simulation over low cost FPGA cluster,” Proceedings of 2009 IEEE International Symposium on Parallel & Distributed Processing, Rome, 2009, pp. 1-8.

[14] Campoccia, A. Favuzza, S. Sanseverino, E.R. Zizzo et al, “Reliability analysis of a stand-alone PV system for the supply of a remote electric load,” Proceedings of 2010 International Symposium on Power Electronics Electrical Drives Automation and Motion, Pisa, 2010, pp. 158-163.