Rapid Star Tracking Algorithm for Star Sensor

Jie Jiang, Guangjun Zhang, Xinguo Wei & Xiao Li
Beijing University of Aeronautics & Astronautics

ABSTRACT

A rapid star tracking algorithm is proposed. In order to speed up the tracking, three techniques including parallel star centroiding, sorting, and star catalogue partition are designed for three time-consuming portions in tracking algorithms. The parallel star centroiding is implemented with Field Programmable Gate Array (FPGA) which avoid image storage and transmission. Update rate of star sensor is improved. Sorting star coordinates in star image plane, and then matching, which avoid matching between stars with a long distance in image plane. Star catalogue partition divides the celestial sphere into many small partitions. In star mapping, guide stars are searched in the partitions near the direction of star sensor's boresight is not in the whole celestial sphere and therefore reduced the total number of searched guide stars. The software and hardware performance of tracking algorithms are simulated. Tracking robustness and the tracking speed comparison are tested in the software simulation. In hardware tests, the tracking time in every step is obtained.

INTRODUCTION

Star sensor is a high-precision, high-reliability attitude measurement component used by most spacecraft [1-2]. A star sensor typically operates in two modes: initial attitude acquisition and tracking mode. The difference between the modes is whether approximate attitude knowledge is available. Initial attitude acquisition mode is also called Lost-In-Space mode and occurs when a star sensor starts to work or when a system fails. In initial attitude acquisition, the task is to perform pattern recognition of the star pattern in the Field of View (FOV). Because no prior attitude

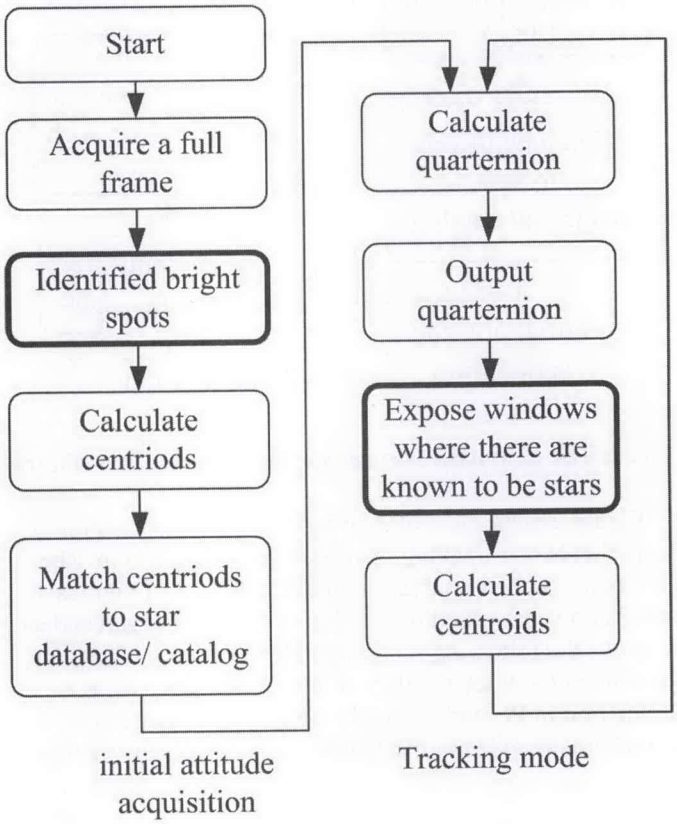

Fig. 1. Star Sensor Algorithmic Flow Diagram of JPL

information is available, a full-sky star identification is needed to establish an initial attitude. Once an initial attitude is established, the star sensor reverts to the tracking mode. Many improved Lost-In-Space algorithms are presented [3-13]. Typically, the identification can be accomplished in a few seconds. The tracking mode track previously identified stars at known positions. The tracking mode is the normal operation mode of the star sensor. As a key operation mode, tracking mode directly affects the star sensor's performance. But, at present, few papers are published for star tracking algorithms [14-15]. Figure 1 is the flow diagram of the Jet Propulsion Laboratory (JPL). The tracking mode of the JPL is the typical star tracking method. The tracking mode sets multiple expose windows simultaneously, outputs the raw pixel data in pre-defined small windows, and calculate

Author's Current Addresses:
J. Jiang, G. Zhang, X. Wei & X. Li, Key Laboratory of Precision Opto-mechatronics Techonology – Ministry of Education, School of Instrument Science & Opto-electronics Engineering, Beijing University of Aeronautics & Astronautics, Beijing, 100191, China.

Manuscript received June 2009. Released to production July 2, 2009.
Review of this manuscript was handled by D. Dobson

0885/8985/09/ $26.00 USA © 2009 IEEE

centroids of the pixels. Attitude and rate information from previous exposures is used in the tracking mode to predict star positions in the next exposure.

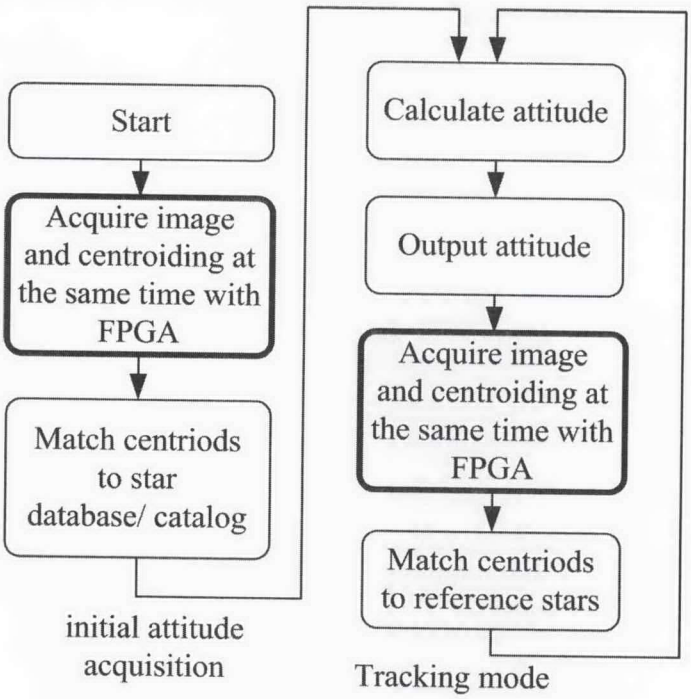

Fig. 2. The proposed star sensor algorithmic flow diagram

Figure 2 is the flow diagram of our star sensor. A rapid and reliable star tracking method is presented herein. The difference of the two tracking mode is shown in Figures 1 and 2 and is described in detail herein. This is organized as follows: the following section entitled *Tracking Algorithm* introduces the tracking algorithm in detail. The software simulation in PC and the hardware test for the tracking algorithm are given in the *Experiment and Results* section. In the *Discussion* section, some advantages of the proposed tracking algorithms are discussed. Finally, some concluding remarks are drawn in the final section.

TRACKING ALGORITHM

Definitions

To facilitate analysis and understanding of the implementation details given in the following sections, some terms are introduced for stars in different reference frames or have special uses:

- ***Reference stars (R):***
  a set of stars previously identified or a subset of guide stars mapping from the star catalog.
- ***Reference star image:***
  The image
- ***Observation stars(O):***
  a set of detected stars (presented with centroid of stars)on the sensor image plane.
- ***Observation star image:***
  Star Mapping:
  generates virtual star image based on the stars information in the star catalog. The star coordinate in the star catalog is the celestial sphere coordinate, so the star mapping is to transform the star coordinate from the celestial sphere coordinate to the sensor image plane coordinate.

Principle

In this algorithm, the (k-1)th star images and before are supposed to have been identified, and the kth reference star

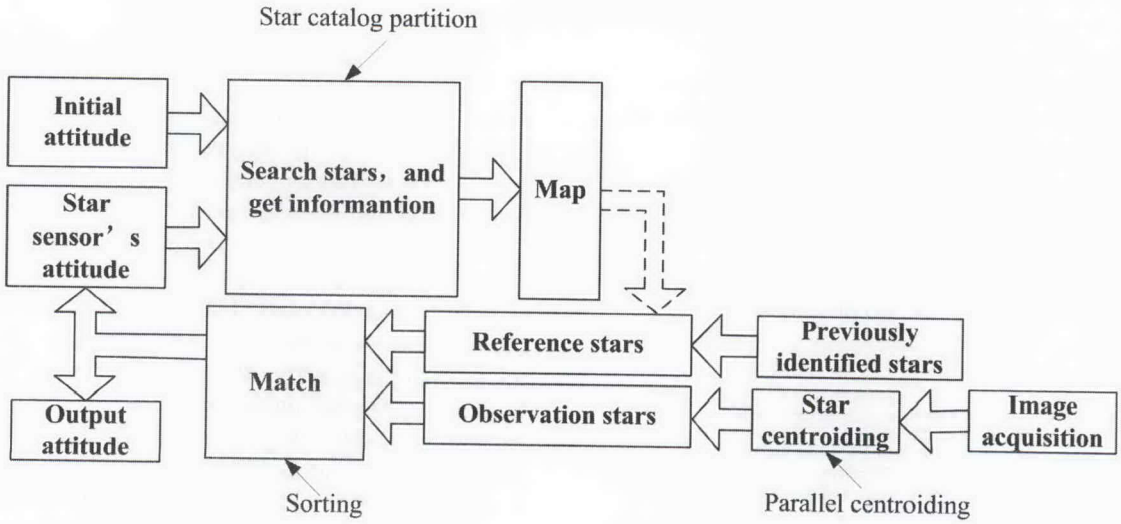

Fig. 3. The principle of star tracking

image is generated. The kth observation star image will be obtained from star centroiding and be tracked. The procedure is shown in Figure 1.

Stars in the kth reference star image match with the stars in the kth observation star image as follows. If a star's neighborhood in observation star image include only one star in reference star image, the star in observation star image is matched with the star in reference star image, the star in the observation star image is tracked successfully. Otherwise, if a star's neighborhood in observation star image includes no star or more than one star in reference star image, the star is not matched and tracked. The distance between the star Oi (i ¡∈[1,n]) (n is the number of stars in observation star image); and the star Rj (j∈[1,m]) (m is the number of stars in the reference star image) is calculated.

$$d = \sqrt{\left(x_{Oi} - x_{Rj}\right)^2 + \left(y_{Oi} - y_{Rj}\right)^2} \tag{1}$$

If $d < r$ $N_{Oi} = N_{Oi} + 1$;

- r is the matching radius shown in Figure 4, N_{Oi} is the matched number
- If $N_{Oi} = 1$ so the star Oi is matched with the star Rj, and is tracked.
- Or the star Oi is not tracked.

In Figure 4, there is only star R7 in the neighborhood of star O6. So, star O6 can be matched and tracked successfully. But there are stars R7 and R11 in the neighborhood of star O9 and no star in the neighborhood of star O10. So star O9 and O10 cannot be matched and tracked. In fact, neighborhood radius relates to star sensor's rate information. According to stars transformation between inertial-based coordinate system and image plane, the star sensor's attitude can be calculated [16].

The (k+1)th reference star image can be generated from the tracking results and used to match with the next observation star image in the same way; then stars can be tracked continuously.

The stars in the reference star image will be moved away from the FOV one-by-one. When the stars left in reference star image is less than 3, the new star identification is necessary. In this case, the star sensor's boresight can be calculated according to the result of the previous tracking, and the stars around the boresight in the star catalogue can be obtained. The information includes right ascension and declination of stars in celestial sphere coordinates, magnitudes, star numbers, and so on. Through star mapping, coordinate of stars in this range transform from celestial sphere coordinate to image plane pixel coordinate. The information of all stars in this reference star image is known. So the new star identification is implemented. The reference star image is composed of newly identified stars for next tracking.

To improve the speed of star tracking, three techniques including: parallel star centroiding for acquiring observation star image quickly; sorting before matching for avoiding unnecessary comparison; and star catalogue partition for speeding up new star identification are introduced for three time-consuming portions in tracking algorithm.

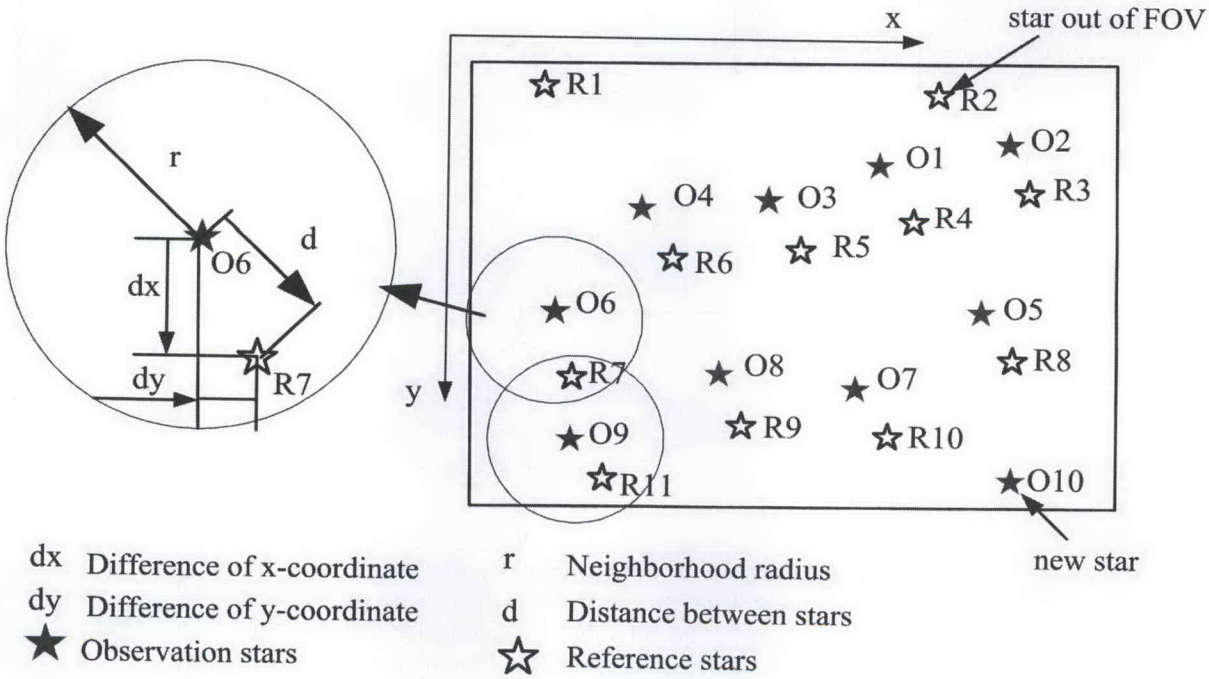

Fig. 4. Star match for tracking

Parallel Star Centroiding

The optics of star sensor is slightly defocused so star sensors will achieve subpixel centroiding accuracy. In the tracking mode based on windows which output the raw pixel data in pre-defined small windows and calculate the centroids with a processor, one window includes only one star, so the centroid calculation is based on the window.

Herein, a parallel star segmentation and centroiding is implemented for full frame image with FPGA. Many stars are included in full frame image, so the star segmentation is necessary. The star segmentation is to mark all connected pixels belonging to the same star region with a common label [17], where the connectivity actually used in practice is 4-connectivity. The result of the labeling is shown in Figure 6; so the centroid is calculated in the star region based on Equation (2).

Here m, n is the size of stars. The star image is raster-scanned and the pixel data is outputted from top left to bottom right. When the pixel data is outputted one-by-one, the pixel data is processed with FPGA as follows and is shown in Figure 5.

$$x_0 = \frac{\sum_{x=1}^{m}\sum_{y=1}^{n} image(x,y)x}{\sum_{x=1}^{m}\sum_{y=1}^{n} image(x,y)}, \quad y_0 = \frac{\sum_{x=1}^{m}\sum_{y=1}^{n} image(x,y)y}{\sum_{x=1}^{m}\sum_{y=1}^{n} image(x,y)} \tag{2}$$

- If the gray value of the pixel is less than a preset threshold T (here, T is 15 on the base of the average background noise), the pixel is considered as background and is labeled zero.

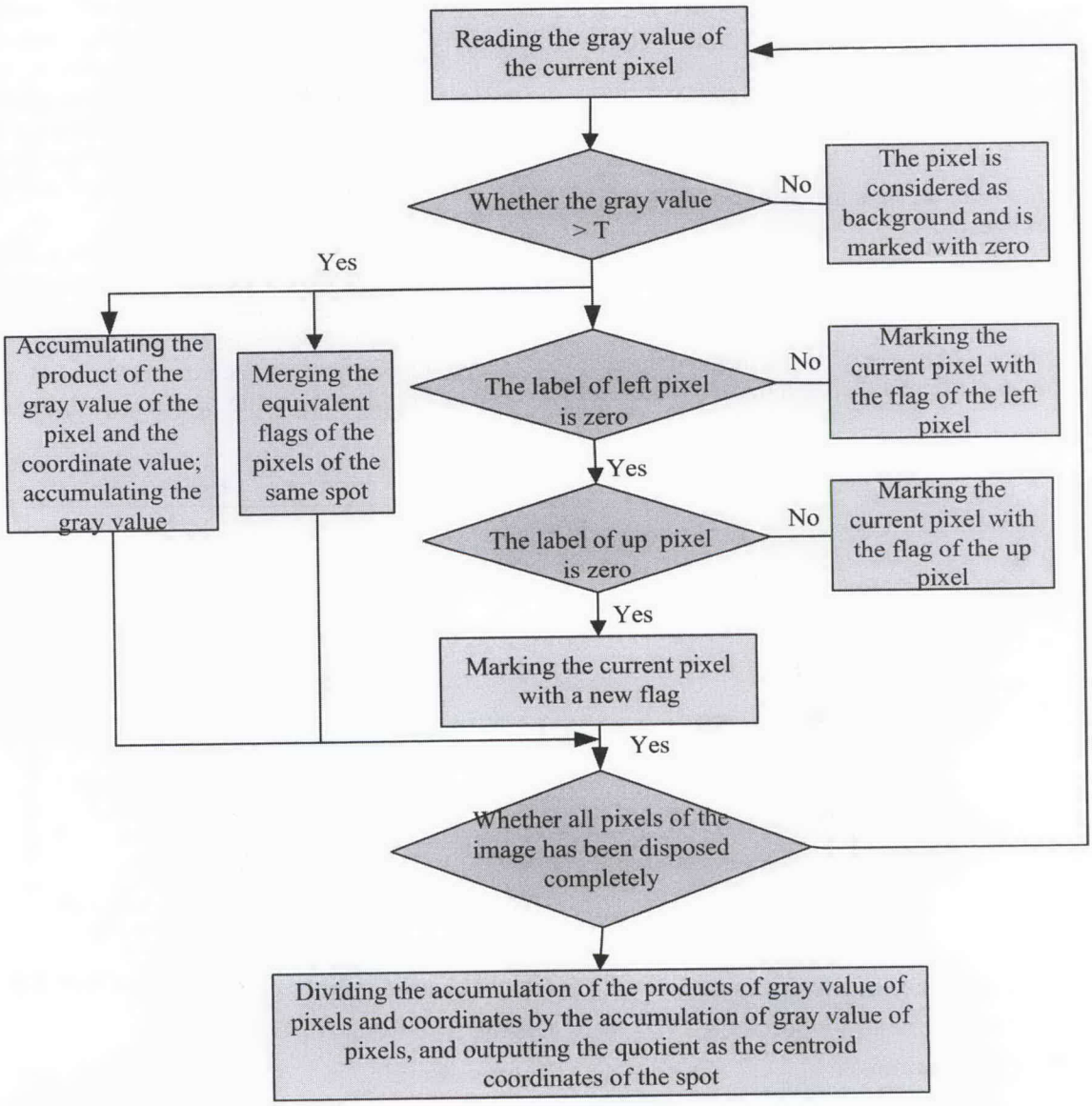

Fig. 5. The flow chart of parallel star centroiding implemented with FPGA

- or else if the label of the left pixel is not equal to zero, so the label of the current pixel is its copy.

- or else if the label of up pixel is not equal to zero, so the label of current pixel is its copy.

- or else allocate the current pixel a new label.

- or store the label of current pixel to the up label buffer and left label register for labeling next pixel.

The labeling is shown in Figure 6. At the same time, the star centroid is calculated as Figure 7 shows.

1) If the gray value of the pixel is larger than a preset threshold T, we accumulate the product of the gray value and the coordinate value of the same star; and at the same time, accumulate the gray value of the same star; and saving the results of the accumulation, respectively.

2) Merge the accumulation of the same star which has more than one label.

Repeat the processes above until all pixels are outputted. After all pixels of the image have been disposed of completely, the star centroid coordinates, not raw pixel data, are buffered and outputted. Compare with the tracking mode based on windows this tracking mode reduces the amount of data output from the image sensor by 4 orders of magnitude and avoids raw pixel data storage and transmission. The update rate of star sensor is improved and can reach 10 Hz for CMOS APS star sensor with the STAR1000 image sensor.

Sorting Before Matching

The star centriods of observation star image are stored randomly. Generally, the matching between the stars in observation star image and reference star image are implemented one-by-one. But matching between those stars with a long distance is meaningless, so sorting before matching is introduced. That is, the stars in observation star images are sorted in ascending order according to coordinates.

- To reduce the computational complexity, the distance is substituted by the difference between the two coordinate as follows,

$$dx = \left|x_{Oi} - x_{Rj}\right| \qquad dy = \left|y_{Oi} - y_{Rj}\right| \tag{3}$$

- Bubble sort is used before matching. Bubble sort is a simple sorting algorithm. It works by repeatedly comparing each adjacent pair of items in a list, in turn, swapping the items if they are in the wrong order and repeating the pass through the list until no swaps are done. The stars in observation images are sorted in ascending order according to the y coordinate. The stars in reference star image have been sorted last matching. The sorting result is shown in Figure 8.

- So after sorting, the range for matching is reduced. For example the star O4 is matched only with the star R3, R4, R5, R6, and R7. The time consumed for matching in star tracking is reduced greatly.

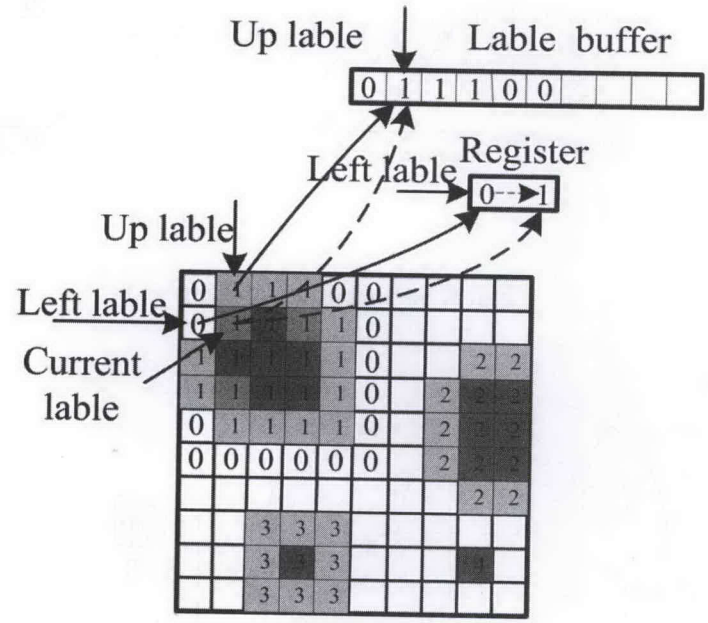

Fig. 6. Star segment

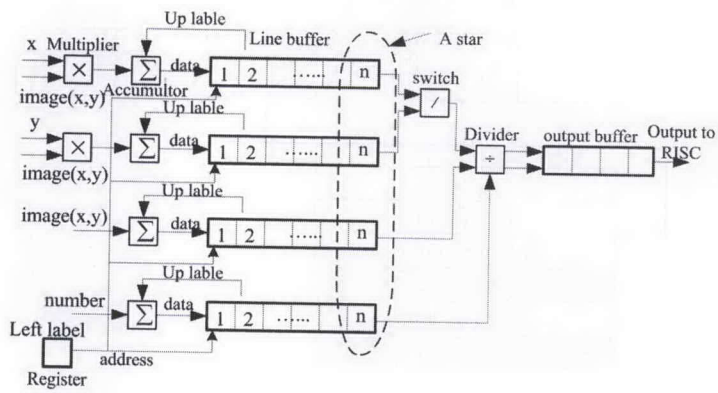

Fig. 7. Star centroiding

Star Catalogue Partition

In the star tracking, generation of star image for new star identification is star mapping which is time-consuming.

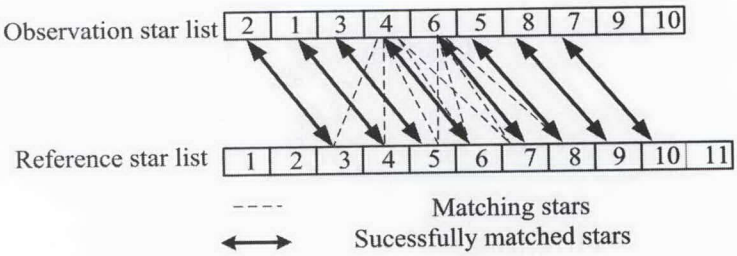

Fig. 8. Sorting and matching

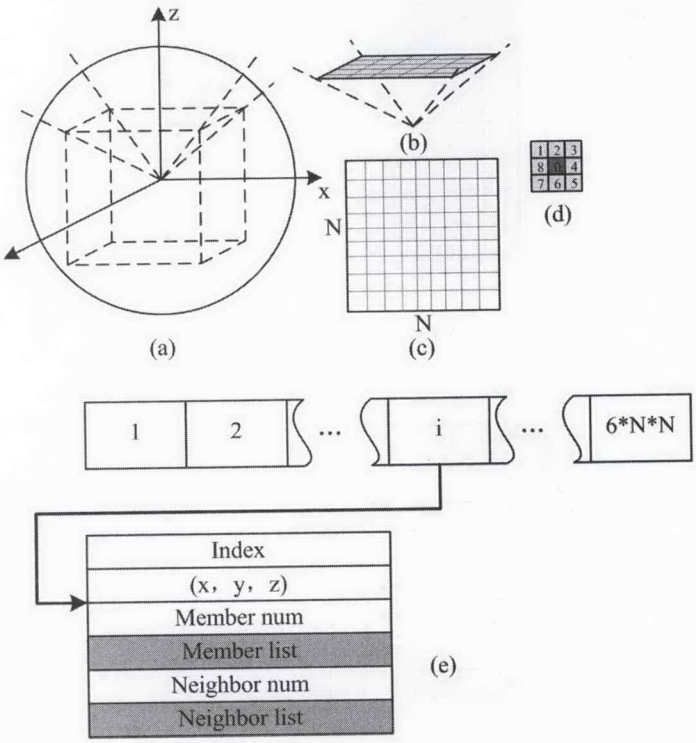

Fig. 9. Star catalog partition

Much time is consumed to search guide stars which can be mapped to the Field of View (FOV). Guide stars in the catalog are scattered irregularly. If all guide stars are searched to retrieve stars around the boresight, which is time-consuming and inefficient. The problem can be solved by introducing the concept of sub-catalog.

Naturally, the star distribution on the sphere is rather non-uniform [18-19]. A novel method of partition sphere by an inscribed cube in Cartesian coordinates is presented herein. The partitioning steps are as follows:

1) The celestial sphere can be divided into six equivalent zones by an inscribed cube which is shown in Figure 9A. A square pyramid is formed by connecting the center of the celestial sphere with the four vertexes on one face of the cube. The square pyramids divide the celestial sphere totally into six equivalent zones S1 - S6.
2) Each one of S1 - S6 can be partitioned by $N \times N$ sub-portions (see Figures 9B and 9C.) The entire celestial sphere can be partitioned into $6 \times N \times N$ sub-portions.

According to the steps above, candidate guide stars can be assigned into sub-catalog. A partition table including all sub-catalogs is generated.

The structure of the partition table is shown in Figure 9E. If the boresight is roughly known, the sub-catalog and its neighbors are to be found quickly. So the guide star around the boresight can be accessed immediately. The guide star catalog and partition table are preprocessed and stored. For an FOV of $12° \times 12°$, in order to cover the FOV with a sub-catalog and its 8 neighbors (see Figure 9D), N should be 9. So the guide star catalog is partitioned into $6 \times 9 \times 9 = 486$ sub-catalogs, the size of each sub-catalog is $10° \times 10°$. Because of the star catalog partition, searching guide star need not access all stars anymore. The average searching range can be narrowed to $9/486 = 1/54$ of the traditional searching range. The time consumed for searching is reduced greatly.

EXPERIMENT AND RESULTS

Software Simulation Results

The star sensor's FOV is $12° \times 12°$ in simulation. A guide star catalog is made of 5103 stars below magnitude 6, based on SAO. Parameter N in the star catalog partition method is 9, so the number of blocks divided is 486. Figure 10 is the interface for software simulation.

Tracking Robustness

When the star sensor's simulated real attitude changes non-linearly, comparison between the star tracking attitude and simulated real attitude every step is shown in Figure 11. The right figure is part of the left figure and is zoomed in. The circle is the real attitude and the cross is the tracking attitude. The figure shows that tracking attitude is consistent with real attitude.

To simulate the influence of the star location error to star tracking, a gauss noise with mean = 0 and standard deviation (std. dev.) = 0-2 pixels is added to the coordinates of each star. Figure 11 is the tracking process with std. dev. = 0.5 pixels. Figure 11 is the tracking process with std. dev. = 2 pixels. Figure 12 shows that the tracking is still successful, but suffers degradation of attitude accuracy. The comparison of attitude accuracy with different standard deviation is shown in Figure 13. The figure shows that because of the gauss noise the attitude error with std. dev. = 2 pixels is nearly 5 times larger than the attitude error with a std. dev. = 0.5 pixels.

Tracking Speed Comparison

To compare the tracking speed with and without sorting and star catalogue partition, the tracking algorithms are

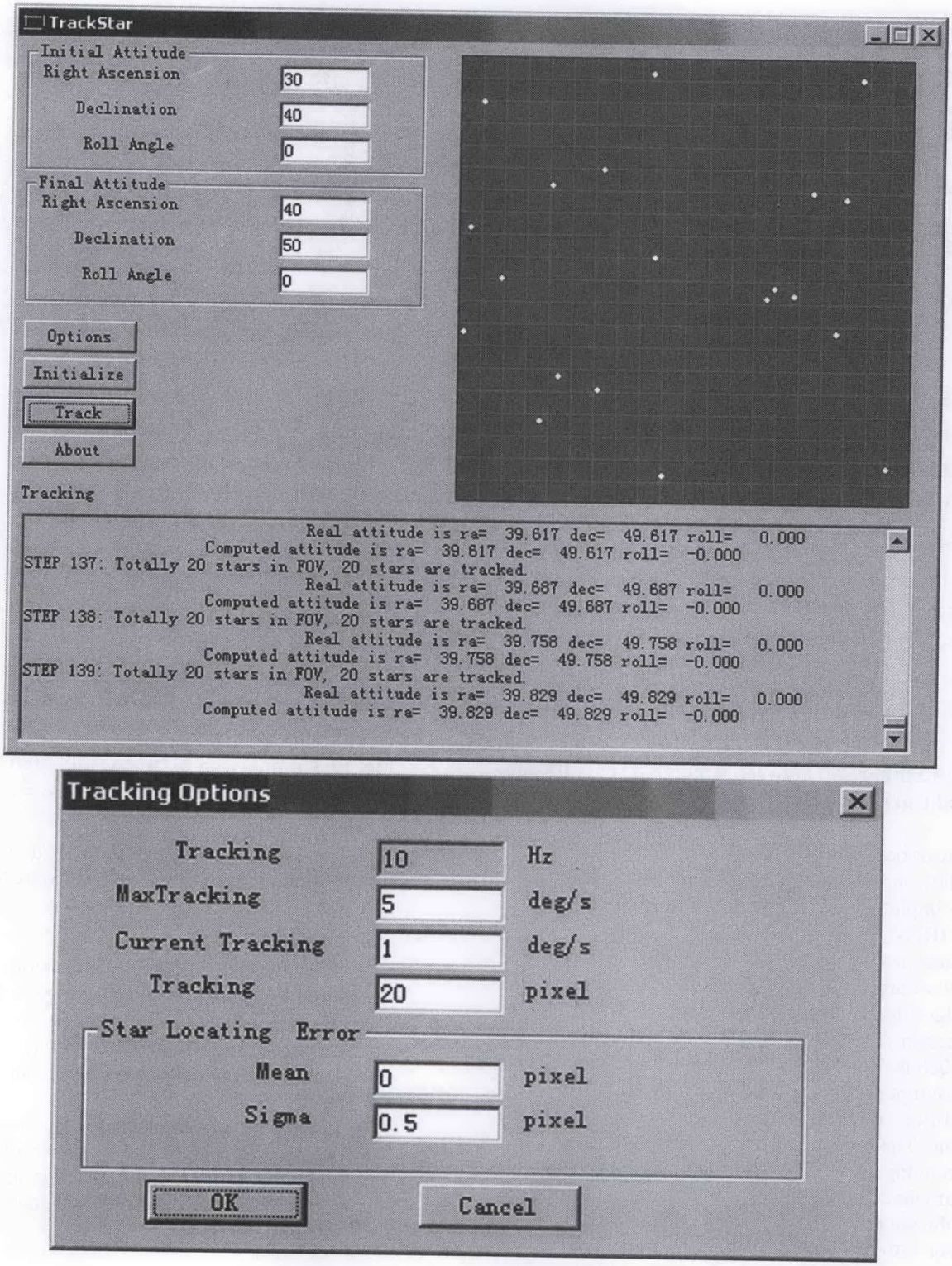

Fig. 10. The interface for software simulation

implemented in C language and run on an Intel Pentium (R) 4 CPU 2.80 G computer and the results are shown in Figure 14 and Figure 15. We can see that the time consumed is reduced greatly, and tracking speed is improved. Because of the star catalogue partition, the average searching range is reduced to 1/54, the star mapping time in star tracking is only 6% of the time without the star catalog partition. Compared with tracking algorithm without sorting, only 10 percent of the time consumed is required for the tracking algorithm with sorting shown in Figure 15.

Hardware Test Result

The hardware test system is shown in Figure 16. In this system, the star field simulator simulates the night sky and

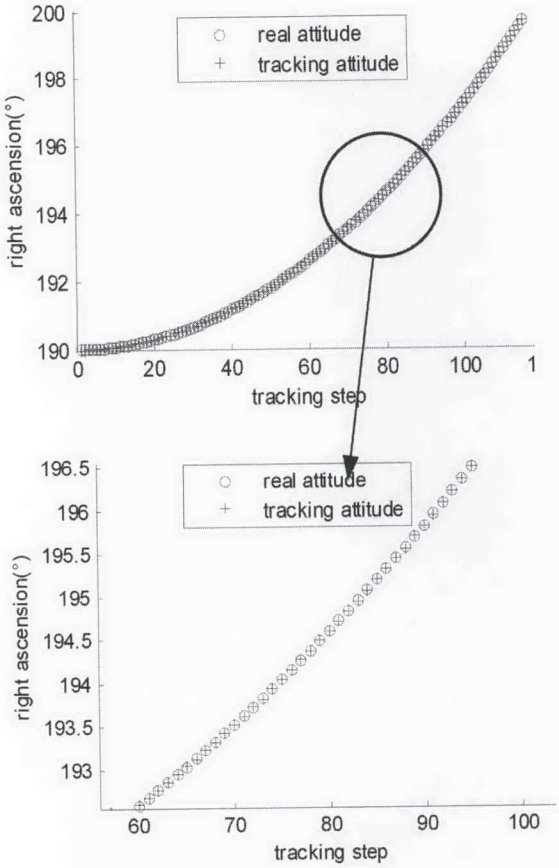

Fig. 11. Comparison between simulated real attitude and tracking attitude (std. dev. = 0.5 pixels)

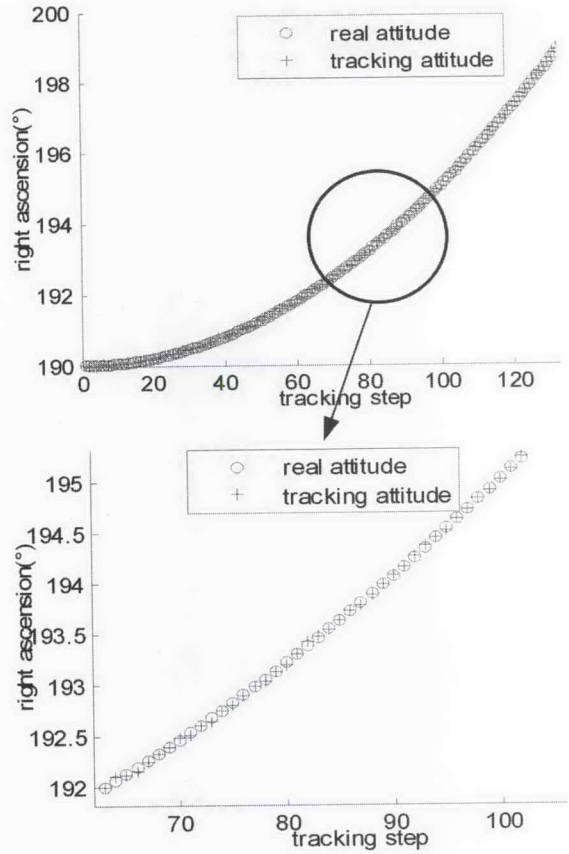

Fig. 12. Comparison between simulated real attitude and tracking attitude (std. dev. = 2 pixels)

the star sensor based on CMOS APS image sensor is developed by our laboratory. The star identification and tracking is implemented with Reduced Instruction Set Computer (RISC) processor. RISC processor is small-volume, low-power, low-cost, and high-performance, so this kind of processor is suitable for the star sensor. This tracking algorithm is simulated and test in an ARM-based RISC processor AT91R40008. The clock frequency for the AT91R40008 is 66 MHz, and the memory is 2 Mbyte. The power consumption for star tacking is only 400 mW.

For real-time tracking, tracking speed is important in the star tracking. The tracking results is transferred to the PC, and shown in Figure 17. The tracking time, tracked star number, simulated real attitudes, and computed attitude are shown in the window. The data shows that computed attitude is consistent with real attitude, so the star tracking is successful. The tracking time in every step is shown in Figure 18. The tracking time in every step keeps less than 100 ms and the tracking time is reduced with the decreasing number of reference stars. The update rate of our star sensor is 10Hz, so this algorithm is suitable for real-time tracking.

DISCUSSION

Although the tracking mode is based on windows is simple, it has some disadvantages:

1) The tracking mode based on windows output raw pixel data, so the data transformation is still the bottleneck of the star sensor.

2) The control logic of image acquisition for initial attitude acquisition and tracking mode is different because of the windows tracking. The difference increases the complexity of control logic and driver circuitry significantly.

3) Limited by data transformation, windows pre-defined is small, the deviation of attitude and rate information will result in prediction errors of the star positions, and cause lost tracking.

4) The star centroids are calculated in the processor which increases the processor load.

To compare with the tracking mode based on windows, the tracking mode proposed herein not only solves the problem above but also has the following advantages:

1) Because of the star centroiding at the same time as with image acquisition with FPGA, the

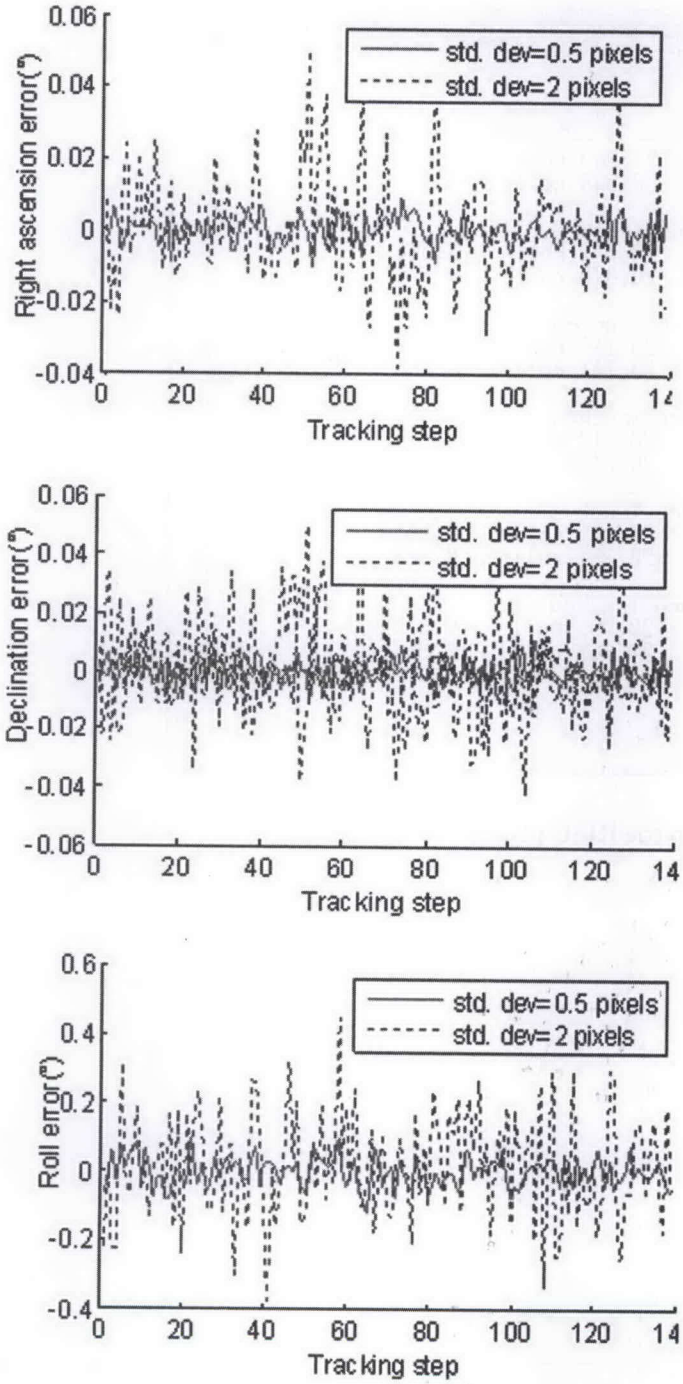

Fig. 13. Attitude accuracy with different standard deviation

amount of data transmission is reduced by 4 orders of magnitude in tracking mode, and the bottleneck of data transformation has been solved.

2) The control logic of image acquisition for initial attitude acquisition and tracking mode is the same in the proposed star sensor, which avoids the complicated conversion of the control logic

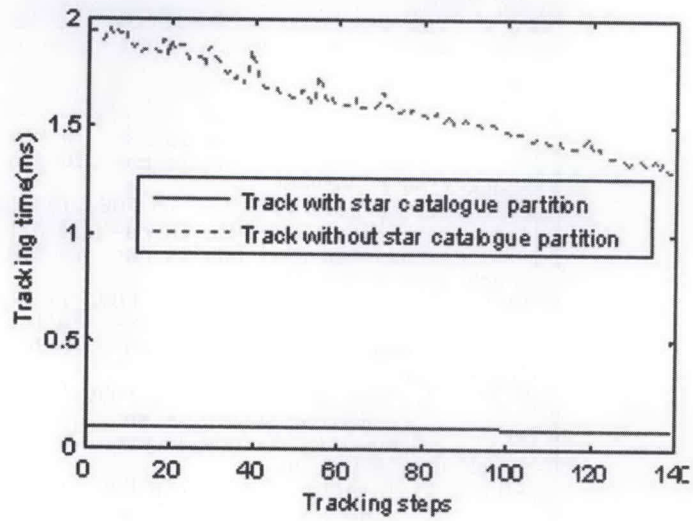

Fig. 14. Tracking speed comparison (star catalogue partition)

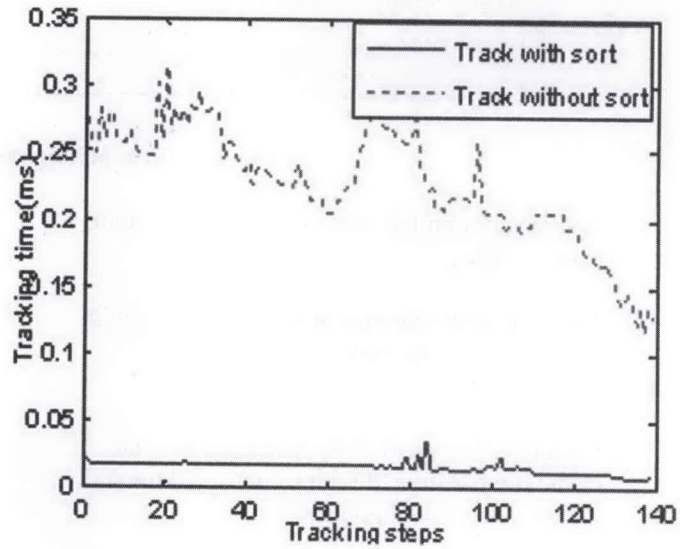

Fig. 15. Tracking speed comparison (sorting)

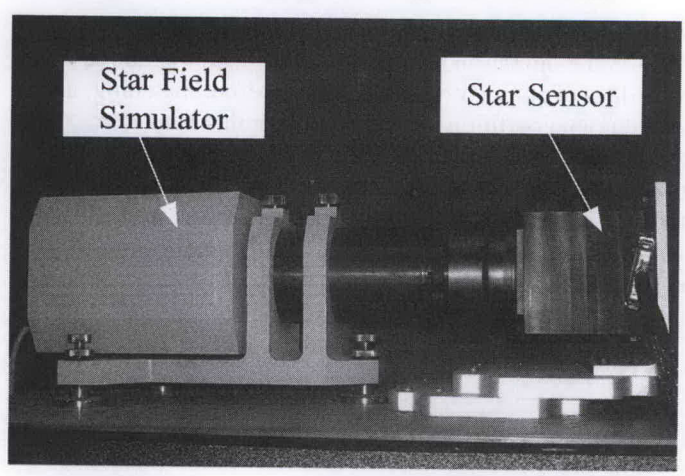

Fig. 16. The hardware test system

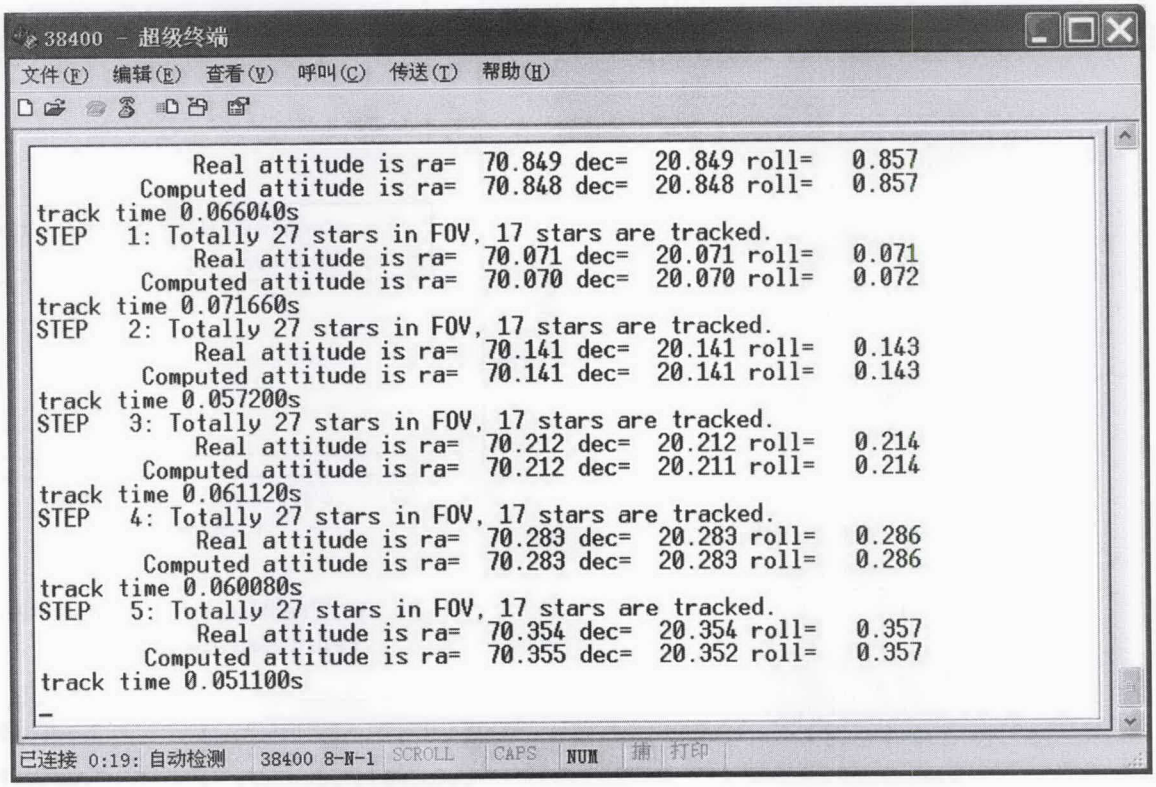

Fig. 17. The tracking results in the RISC processor

and driver circuitry, and simplifies the system control logic.

3) The proposed technique of star centroiding with FPGA reduces the signal processor load significantly.

4) Because of sorting before matching and star catalogue partition, the time consumed for star tracking is reduced greatly.

CONCLUSIONS

Herein, a rapid star tracking algorithm is presented. Initially, the operation principle is explained; then three techniques of star centroiding, sorting before matching, and star catalogue partition are introduced in this algorithms. The techniques improve not only the speed of tracking algorithms but also the star sensor's update rate. Finally, the simulation indicates that this algorithm is effective and reliable. The tracking speed of this algorithm is rapid from the results of the software and hardware tests.

ACKNOWLEDGEMENT

This work is funded by the China National Space Administration, China Aeronautics Science Foundation (03I51004), and China Astronautic Innovation Foundation. The author wishes to express her appreciation to them.

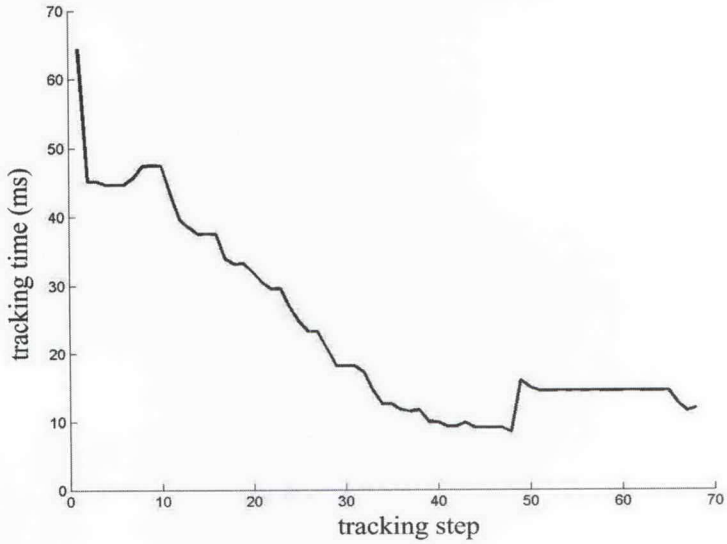

Fig. 18. Star tracking time in hardware test system

REFERENCES

[1] Liebe, C.C.,
Accuracy performance of star trackers: a tutorial [J],
IEEE Transactions on Aerospace and Electronic Systems,
Vol. 38, 2002, pp. 587-599.

[2] Liebe, C.C.,
Star Tracker for Attitude Determination [J],
IEEE Aerospace and Electronics Systems Magazine,
Vol. 10, 1995, pp. 10-16.

[3] Junkins, J.L., White, C. and Turner, D.,
Star pattern identification for real time attitude determination,
Journals of Guidance, Control and Dynamics,
Vol. 25, 1977, pp. 251-270.

[4] E. Silani and M. Lovera,
Star Identification Algorithms: Novel Aprroach & Comparision Study,
IEEE Trans. on Aerospace and Electronics Systems,
Vol. 42, 2006, pp. 1275-1287.

[5] Read-Eisenman, A. and Liebe, C.C.,
The advancing state-of-the-art in second generation star trackers,
IEEE Aerospace Conference Proceedings,
1998, pp. 111-118.

[6] Liebe, C.C., Alkalai L., Domingo, G., Hancock, B., Hunter, D., Mellstorm, J., Ruiz, I., Sepulveda, C. and Pain, B.,
Micro APS based star tracker,
IEEE Aerospace Conference Proceedings, 2002.

[7] Accardo, D. and Rufino, G.,
Brightness-independent star-up routine for star trackers,
IEEE Trans. on Aerospace and Electronic Systems,
Vol. 38, 2002, pp. 813-823.

[8] Alexander J. and Chang, D.,
Star tracking and identification algorithm, scene simulation, and testing,
SPIE, Vol. 2803, 1996.

[9] Quine, B.M. and Durrant-Whyte, H.F.,
Rapid star pattern identification,
SPIE, Vol. 2739, 1996.

[10] Van Bezooijen, R.W.H.,
Automated Star Pattern Recognition,
Ph.D. Dissertation, Stanford University, California, 1989.

[11] Padgett, C. and Kreutz-Delgado, K.,
A grid algorithm for autonomous star identification,
IEEE Aerospace and Electronics Systems Magazine,
Vol. 7, 1997, pp. 202-213.

[12] Mortari, D., Junkins, J.L. and Samaan, M.,
Lost-in-space pyramid algorithm for robust star pattern recognition,
presented at 24th Annual AAS Guidance and Control Conference, January 31-February 4, 2001, Breckenridge, CO.

[13] Hye-Young, Kim and Junkins, J.L.,
Self-organizing guide star selection algorithm for star trackers: Thinning method,
IEEE Aerospace Conference Proceedings, Vol. 5, 2002.

[14] Baohua Li, Yingchun Zhang, Huayi Li, etc.,
A star tracking algorithm suitable for star sensor,
Proc. SPIE 6595, 659523, (2007).

[15] Jiang Jie, Li Xiao, Zhang, Guangjun, etc.,
Fast star tracking technology in star sensor,
Proc. SPIE 6358, 63580A, (2006).

[16] Zheng Wanbo, Xia Liang and Hao Zhihang,
Method calculating of satellitic instantaneous attitude based on star tracker [J],
Journal of Ji Lin University, February 2003,
Vol. 21, pp. 28-30.

[17] Cohen, H.A.,
Parallel algorithm for gray-scale image segmentation,
Intelligent Information Systems,
Australian and New Zealand Conference,
1996, pp.143-146.

[18] Chen, Yuanzhi, Hao, Zhihang and Wang, Guohui,
Constitution of guide star catalog for star sensor [J],
Optics and Precision Engineering,
Vol. 27, 2000, pp. 5-10.

[19] Wei, Xinguo,
Star identification for star sensor, Ph.D. Dissertation,
Beijing University of Aeronautics and Astronautics, 2004.