



Modification and hardware implementation of star tracker algorithms

Maliheh Hashemi¹  · Kamaleddin Mousavi Mashhadi¹ · Mohammad Fiuzy¹Received: 29 July 2019 / Accepted: 17 October 2019 / Published online: 1 November 2019
© Springer Nature Switzerland AG 2019

Abstract

In this paper, a laboratory model is designed to evaluate the performance of star tracker algorithms for attitude determination of a satellite. Star tracker is the most accurate attitude sensor that determines satellite direction by applying centroiding algorithm, star identification and attitude determination. To utilize such algorithms, first, high quality of star images are needed which should be provided through the star tracker camera. Then, such images are given to processors and it determines the attitude of camera and satellite in three axes based on mentioned algorithms. First, in preliminary design, we define important star tracker parameters, like accuracy, detector, processor and field of view. In this paper, we have considered the range accuracy in yaw/pitch axis less than 20 arcsec and for roll axis less than 100 arcsec. To improve the attitude determination accuracy, we have applied an adaptive structure in centroiding algorithm, as only brightness stars are selected and identification algorithm is done based on them. Another important parameter is speed of identification algorithm which is handled by an ARM processor and improving pyramid algorithm, we reached less than 25 ms. Duo to this time, updating rate would be desired. Also knowing the coordination of bore sight direction that is distance between focal lens and imager is another important parameter that affects the accuracy of attitude and by ground calibration of camera, this parameter could be estimated carefully. Finally, implementation results on real images that are captured by a star tracker demonstrate optimum performance of algorithms.

Keywords Star tracker · Centroiding algorithm · Star identification · Star catalog · Attitude determination

1 Introduction

One of the first ideas for navigating in the sky was the use of the position of the celestial bodies in the sky. With the advent of such an idea, they began their evolutionary astronomy navigation systems. In the past, the astronomical navigation systems were one of the key pillars of guiding planes, ships, missiles and satellites. But with the advent of modern radio navigation systems in the past decades, astronomical navigation has become a back-up system, and in the aftermath of GPS, virtually astronomical navigation was used only for space missions. At present, one of the most important pillars of the spacecraft navigation systems is the space mission of the star trackers. These detectors utilize the latest electronic and optical

systems with low volume, weight and power. One of the most important requirements of all satellites, since its inception, is the sub-system of control and determination of the situation, because without it, use of the satellite to carry out missions is not possible. The status determination system, an integral part of the satellite control system, plays a significant role in satellite control algorithms [1–3].

The solar panels should face the sun, and satellite antennas should face the earth. Other sensors must be in a certain direction for correct use of satellites. Generally, satellites in space rotate in three axis and they may go out of their orbit. In this case, sub-systems of attitude determination and control act and by using the facilities in satellite, the attitude is returned to the desired condition [4–7]. The attitude of satellite is measured by onboard sensors,

✉ Maliheh Hashemi, malihe_hashemi86@yahoo.com; Kamaleddin Mousavi Mashhadi, sk_mousavi@iust.ac.ir; Mohammad Fiuzy, mohammad.fiuzy@yahoo.com | ¹School of Electrical Engineering, Iran University of Science and Technology, Narmak, Tehran, Iran.



and corresponding actuators change the attitude along to desired direction [8]. Attitude determination accuracy is necessarily limited by sensor measurement accuracy and currently only lower accuracy attitude measurement instruments such as sun sensors and magnetometers have flown on CubeSats with success. Star trackers will provide the level of attitude determination needed to support more challenging pointing requirements [9–11]. Most spacecraft uses gyroscopes to measure their angular velocity continuously. The major problem with the use of gyroscope in spacecraft setup systems is that over time, the bias will lead to a gradual error, which will cause the actual difference in space between the time and the measured value to increase gradually. Figure 1 shows the satellite status using star tracker.

Today, because of high accuracy of star tracker, this sensor is used more than others and other sensors are used as support devices [5–7]. The task of star sensor is providing high-quality star images and matching them with catalogs of stars which are stored in sensor's memory and identifying stars in image and finally determining attitude of satellite based on them [8–10]. A star tracker works in two modes: lost-in-space (LIS) mode and tracking.

The difference between them 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 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 model. Typically, the identification can be accomplished in a few seconds. The tracking mode tracks previously identified stars at known positions. The tracking mode is the normal operation mode of the star sensor. [11–13]. Process of star

tracker consists of three main steps: centroiding, star identification and attitude determination. Centroiding takes the image from the camera and determines the coordinates of light sources in the image plane, which can then be converted to unit vectors in the tracker coordinate frame. Star identification is the crux of the star tracker. The unit vectors in the tracker frame are analyzed and compared to a star catalog to determine which stars are in the image frame and consequently provide unit vectors in the inertial reference frame. Finally, the list of unit vectors in both the tracker and inertial frame is run through a vector-based algorithm to determine the attitude of the star tracker in the inertial frame. The attitude can be output in various formats; among most common are quaternions, Euler angles and direction cosine matrices (DCMs) [13–15].

Because of the importance of attitude determination in LIS mode, our focus is on it. In this paper, after doing some researches on different types of algorithms in different star sensors, the best algorithm is found in each case. By modifying star tracker algorithms that are used in similar cases like centroiding and identification algorithms which are explained in details in [15–18], we could improve accuracy and speed of algorithms. These improvements consist of applying adaptive structure to centroiding and modifying search method in identification. Then, their performances are evaluated through MATLAB software and in the next step it is applied on a proper ARM processor. In this step, for evaluating algorithms used simulated star images that are provided via MATLAB software by orbital movement simulation of LEO satellite in a specific orbit. After ensuring the performance of algorithms and also processor, we evaluated system on real images that were captured by a star tracker camera. And finally, we could offer a preliminary design for manufacturing a star tracker with desired accuracy.

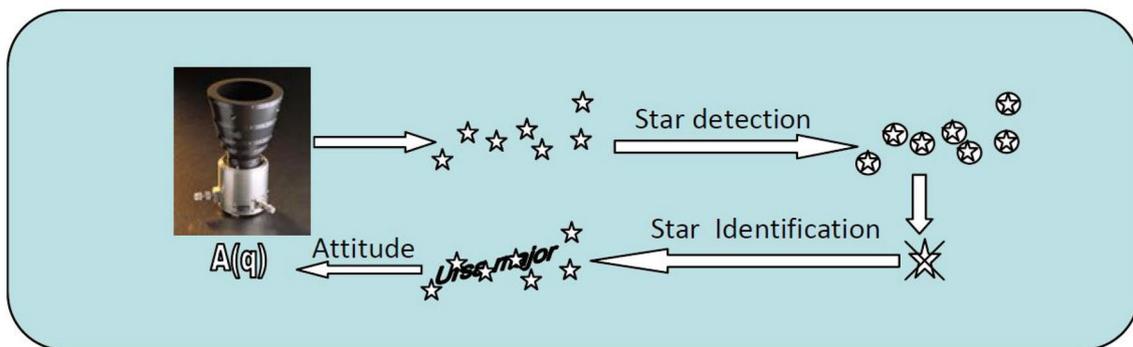


Fig. 1 Conventional flowchart to determine satellite status using star tracker

Table 1 Selected characteristics for preliminary design of star tracker

	Selected characteristics
Weight	2–3 kg
Accuracy	< 20 yaw and pitch < 100 roll
FOV	8° × 8°
Sensitivity	2.5–6.59 (M_I)
Update rate	4–5 Hz
Detector	CMOS
Processor	ARM
Input voltage	20–30 v
Temperature rate	–20 °C to 50 °C

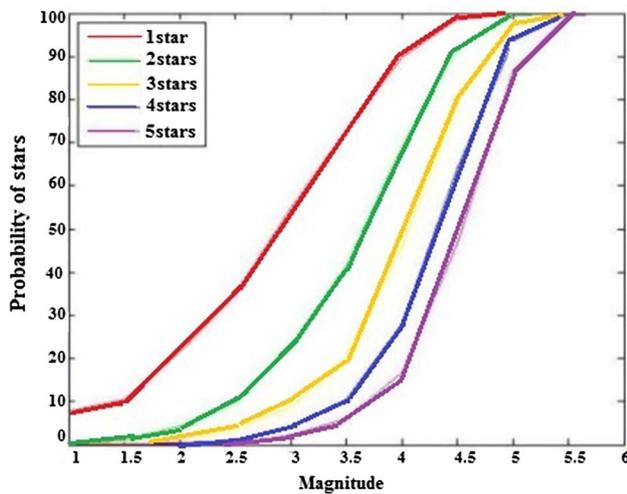


Fig. 2 Monte Carlo analysis for 8 * 8 rectangular FOV

2 Technological approach

Based on researches on parameters of twenty produced sensors, we consider some of the parameters as design requirements. These parameters are summarized in Table 1.

Field of view of star tracker needs to be chosen such that at least three detectable stars are always in the FOV. Three is the minimum amount of stars required to determine three-axis attitude. Wide FOV means wide part of sky is scanned by star tracker camera, but sensor accuracy decreases in this situation. The narrower FOV, the more accurate attitude determination, but it is harder to determine initial attitude of satellite. In order to choose minimum FOV required for different limiting magnitudes, a simulation was run in MATLAB. Figure 2 was created using Monte Carlo simulations. For doing this, we fixed FOV on 8° × 8° (rectangular) and then selected a star from star catalog. After that, we added FOV/2 to right ascension (RA) and declination (Dec) of selected star and then all stars which had smaller RA and Dec than that selected star put into a group. We repeated this algorithm for all stars in catalog.

As you can see, we definitely have five stars in FOV that have magnitudes between five and six.

The processor is one of the most important parts of star tracker that directly determines accuracy of sensor, updating rate, weight and its power consumption. The ARM processors have been used in star sensors due to low power consumption, the optimal capabilities and its accessibility. Table 2 compares the feature of two star tracker ST-100 and ST-200. In ST-200, the optical, electronic and mechanical parts and also the algorithms have been improved.

Due to our researches, using an ARM processor besides CMOS instead of CCD reduces the size of board and saving money (50%) and also decreases in power consumption and weight (77% and 90%). All in all, for testing algorithms,

Table 2 Comparison of features ST 100 and ST 200 [3]

	ST-100	ST-200	Improvement
Mass	740 g	74 g	90%
Power consumption	3 W (peak)	0.7 W (peak)	77%
Price	low	lowest	50%
Accuracy	30/200 arcsec	30/200 arcsec	
Design life	3 years	1 years	
Detector	CCD	CMOS	
Processor	FPGA and ARM7	ARM9	
Number of PCB	4	2	50%
PCB footprint	50 × 50 mm	35 × 35 mm	50%
Interface	RS422	RS485, SPI and I ² C	
Electronic integration	Moderate	High	
Market readiness	2009	2011	

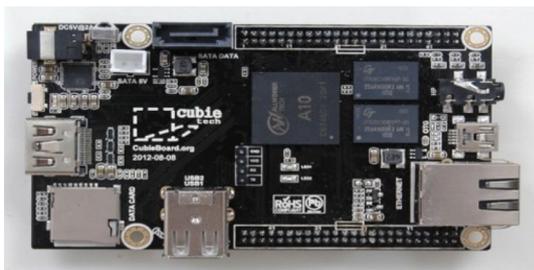


Fig. 3 Processing board used for implementation

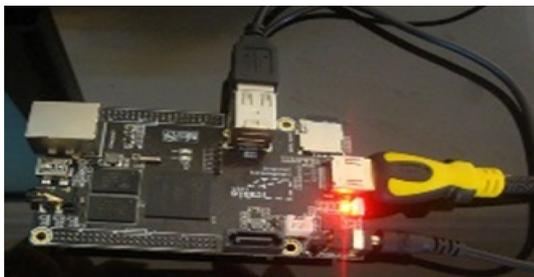


Fig. 4 Using of Cubieboard1

we have decided to use Cubieboard1 which includes the processor of Allwinner A10 (Cortex-A8). Figures 3 and 4 show the board and how it is used.

3 Description the algorithms used in star tracker

Star tracker is an electronic camera connected to a micro-computer. First by using images that are captured from sky, the stars are identified by centroiding and identification algorithm and satellite orientation is determined based on these observations. An autonomous star tracker can detect the patterns of stars in its field of view and determine attitude relative to celestial sphere. In this section, we describe algorithms which are used in this project.

3.1 Centroiding algorithm

After removing noise from star tracker images, they are given to processor for centroiding algorithm. In this step, situation of each star in the image is specified. Centroiding is very important since it affects the accuracy of other algorithms and sensor directly. It means that the place of each star can be specified by accuracy less than 1 pixel. If the stars are recorded in a focused way so that the light of star locates on one or two pixels, pixels will be saturated and the accuracy of centroiding will be reached within pixel range. Therefore, all the star trackers record the image in

an unfocused way which means the image photons will be disputed in a lot of pixels and in this case the algorithm of finding the center performs with accuracy lower than pixel. There are two major techniques for centroiding. 1: The center of mass 2: Point Spread Function.

Figure 5 explains these techniques. Speed and accuracy of centroiding algorithm are a criterion to compare and choose an optimal technique to apply this algorithm.

After examination and applying above algorithm in MATLAB, it has been concluded that the PSF algorithm is more accurate than COM algorithm, but it is more sensitive to noise and also the calculation load of COM technique is less than the other one, so this algorithm has been used in many missions. Among COM algorithms, the parallel method has been used which is faster than other methods. In this technique, first the stars are categorized and finding center is done for whole image. The complete frame of image includes many stars, so that this categorization is necessary for stars. The classification of stars means that all the pixels attached to a star are marked with a shared label. In other words, we distinguish the star pixels. After labeling, the center of each star is calculated through Eq. (1) [3]

$$x_0 = \frac{\sum_{x=1}^M \sum_{y=1}^N image(x, y)x}{\sum_{x=1}^M \sum_{y=1}^N image(x, y)} \quad y_0 = \frac{\sum_{x=1}^M \sum_{y=1}^N image(x, y)y}{\sum_{x=1}^M \sum_{y=1}^N image(x, y)} \tag{1}$$

where N and M are the numbers of pixels in which finding center occurs, x and y are the coordination of pixels, $image(x, y)$ is light intensity in pixel, x_0 and y_0 are the centres of star in the image.

Figure 6 shows how stars are labeled and centers are found in this algorithm.

In 1991, Buil proves that the spreading intensity function of stars is Gaussian. In Fig. 7, intensity for hypothetical star with center of $(0, 0)$ has been shown. To measure the accuracy of parallel algorithm COM, this hypothetical star as an input has been given to it. However, it should be noted that the range of spreading function is a number from 0 to 255 that it is matched with light intensity of a

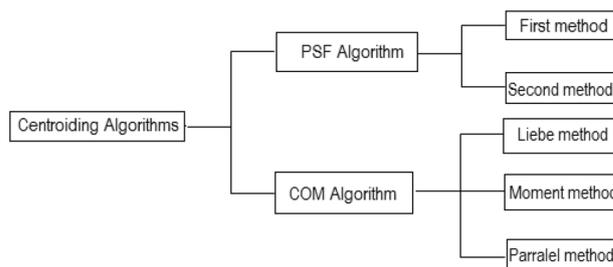


Fig. 5 Assortment of centroiding algorithm

Fig. 6 Flowchart for centroiding algorithm [3]

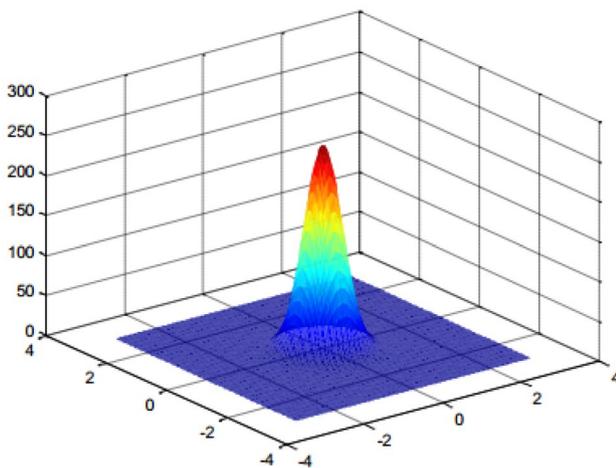
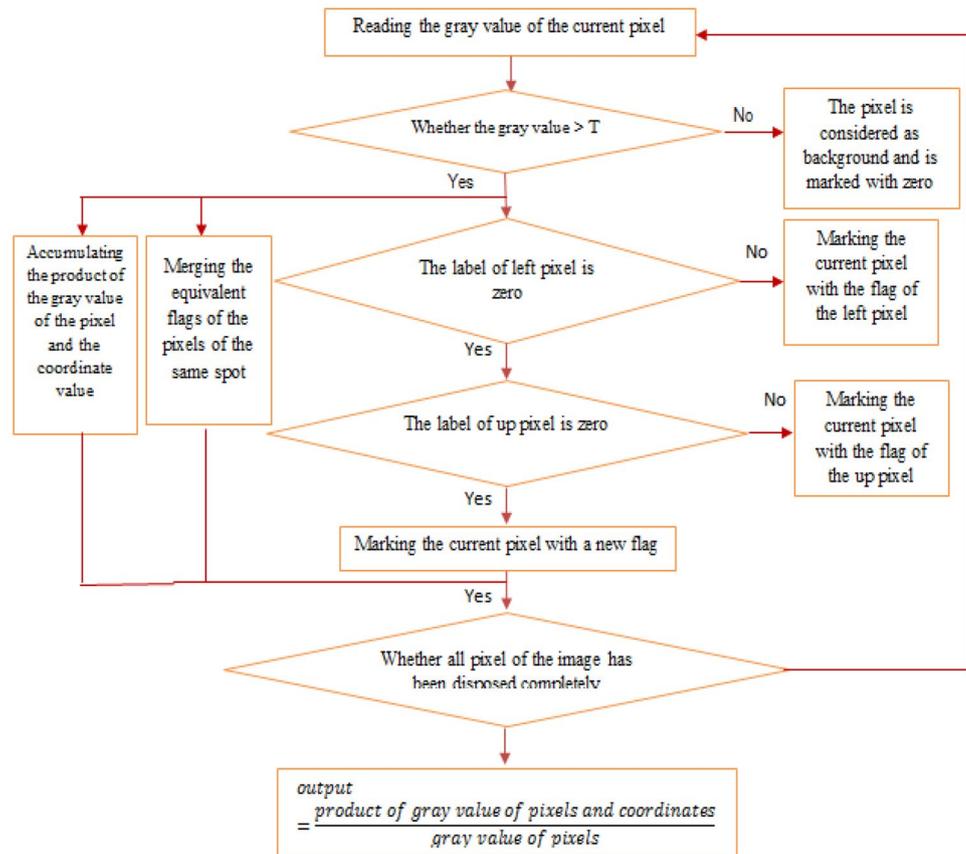


Fig. 7 Intensity for hypothetical star

real star. In fact, deviation rate of algorithm response (0, 0) is the same as algorithm error. Figure 8 shows the error algorithm in x and y axes 500 iterations.

The mean and variance of error for algorithm along x is 0.0269 and 0.00050488 and in y axis is 0.0269 and 0.00050580. To examine the accuracy of algorithm in real situation, Gaussian noise with the power of 0.01 is added

to light intensity of stars. In this case, the mean and variance of error in parallel with x and are 0.0371 and 0.000846 and in parallel with y are 0.0391 and 0.0015, respectively.

3.1.1 Improvement of centroiding algorithm by an adaptive structure

As you will see in Sect. 3.2, identification algorithm is based on selecting four stars and in each image several stars can be found which sometimes more than four. The way of choosing four stars among the number of found stars affects the accuracy of centroiding by sensor. Being careful in mechanism of applying of a COM algorithm, it is clear that the stars with more light intensity are calculated more accurately. Thus, by applying an adaptive pattern on parallel finding center COM algorithm, we correct the algorithm in a way that chooses four brighter stars. You can see this structure in Fig. 9.

3.2 Star identification algorithm

In this step, a pattern is formed and compared with constructed pattern from a catalog of stars. This catalog contains information like right ascension, declination, size, spectrum and...about stars. Hipparchus, Tyco, UCAC, HYG

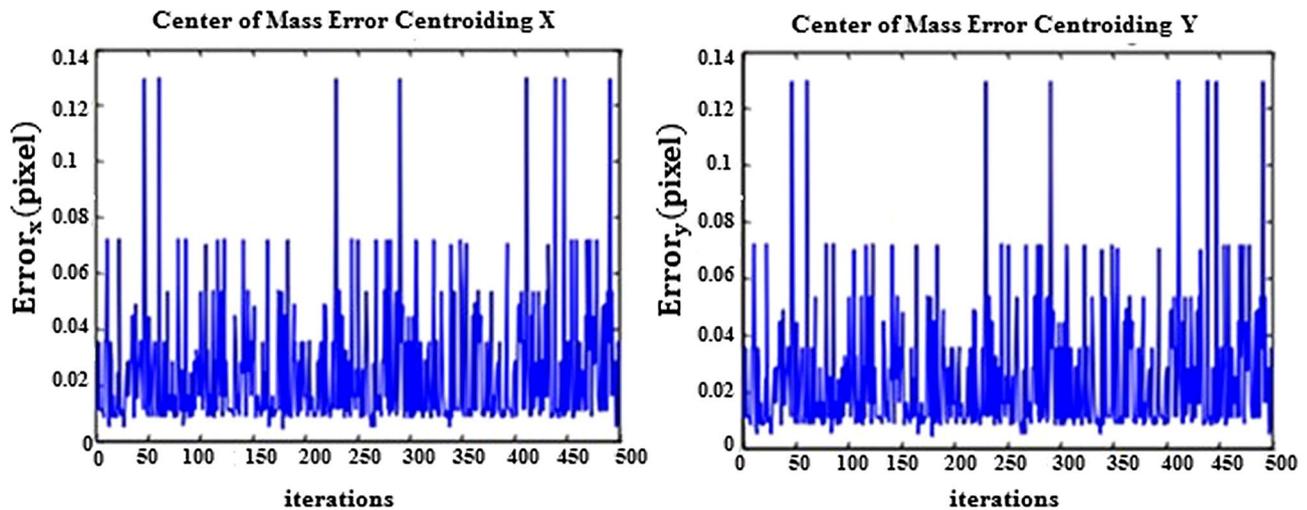


Fig. 8 Accuracy of centroiding algorithm in x and y axes with 500 iterations

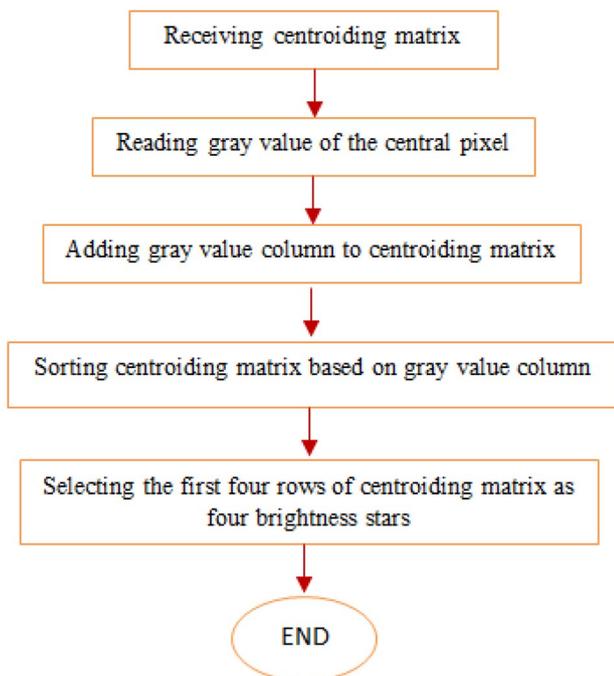


Fig. 9 Flowchart of adaptive structure of centroiding algorithm

and SKY-map 2000 are commonly used in star sensors as catalogs. The catalog which is used in this paper is Hipparchus catalog (It is about 19.907 megabytes). With regard to optical observation (lens weight...), catalog is filtered and the stars with magnitude of 2.5–6.59 are saved. The load of filtered catalog is reduced to 379 kilobytes. There are different techniques to read catalog by the processor. The techniques which consume time and they need to add libraries to compiler. To do so, firstly we transform the

catalog to a textual file and then fstream library is used and by this technique, whole catalog is read less than 0.1 ms by a processor.

```

float **catalog=new float*[7457];
        for (int i=0;i<7457;i++)
            catalog[i]=new float[6];
ifstream rod("catalog.txt");
float nc;
int zc=0;
int jc=0;
while(zc<7457){
    rod>>nc;
    catalog[zc][jc]=nc;
    jc++;
    if(jc>5){
        jc=0;
        zc=zc+1;
    }
}
    
```

3.2.1 Modified pyramid algorithm

Our measures to find the most proper algorithm from parameters like detection percentage, right detecting algorithms, processing time, the volume of database, the number of needed stars for processing in the image, the applicability of algorithm are the frequency use in space missions. In Table 3, we have compared the most well-known identification algorithms in different parameters. One of the most usable techniques for star identification is modified pyramid algorithm which is robust and helpful for solving problems in LIS and tracking modes. Regarding the results, we consider modified pyramid algorithm in LIS mode.

Table 3 Comparison of identification algorithms with each other

Algorithm	Correct identification (%)	No. star	Processing time (s)	CPU speed (MHz)	Clock speed ($\times 10^8$)	Star catalog size	Data base (size)	Mission
Liebe	94.6	3	0.5	32	0.165	8000	1 Mb	
Grid	99.7	10	0.187	1600	1.6	9000	313 Kb	
Polestar	99.7	10	1.04	350	3.6	4127		
Zhang	97.57	7	0.025	800	0.2	5102	344 Kb	
Oriented triangles	97.8	4	0.75	650	4.875	6000	73 Kb	
Angle	61	7	6.27	500	31.35	5000	12 Mb	
Planner angle	94	4	1	500	5	5000	167 Mb	Micro-satellite_TAS2 Fast star tracker
Spherical triangle	92	4	1.6	500	8	5000	167 Mb	STS 107
Modified pyramid	99.8	4	0.5	2000	10	8816	1.12 Mb	Pico satellite GIFTS_EO-3 Mission CanX1 CubeSat

At the heart of pyramid method is the *k*-vector approach for accessing the star catalog, which provides a searchless means to obtain all cataloged stars from the whole sky that could possibly correspond to a particular measured pair, given the measured interstar angle and the measurement precision. The pyramid logic is built on the identification of a four-star polygon structure—the pyramid—which is associated with an almost certain star identification. The set of $M = \frac{n!}{(n-2)!2!}$ interstar angles associated with a spherical polygon set of *n* stars, such as pairs (in pyramid algorithm *n* = 4). More specifically, the star pattern geometric structure for the purpose of star identification is defined by the set of *M* interstar angles $\{\theta_{ij} = \theta_{ji} = \cos^{-1}(b_i^T b_j)\}$ measured between each distinct pair of the *p* line-of-sight vectors $\{\theta_{ij} = \theta_{ji} = \cos^{-1}(b_i^T b_j)\}$ that point from the sensor toward the vertices of the star spherical polygon on the celestial sphere. Matching the set of *M* measured interstar angles $\cos^{-1}(b_i^T b_j)$ with a cataloged set of interstar angles $\cos^{-1}(r_i^T r_j)$ to within measurement precision provides the basis for a hypothesis that the stars at the vertices of the measured polygon of stars are indeed the cataloged stars at the corresponding vertices of the matching polygon from the star catalog. Figure 10 shows the basic star structure used within the algorithm, which consists of a basic star triangle, identified by the indices *i*, *j*, *k*, together with a “confirming fourth star” identified by the index *r*.

The pyramid algorithm contains several important new features. The first is access to the star catalog using the *k*-vector approach instead of the much slower binary search technique (see the appendix). The *k*-vector database is built a priori for some given working magnitude threshold and for the star tracker maximum angular aperture. Essentially, the *k*-vector table is a structural database of all cataloged star pairs that could possibly fit

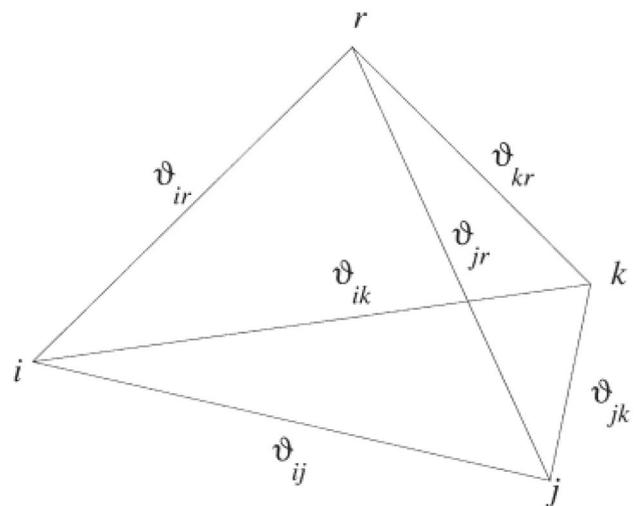
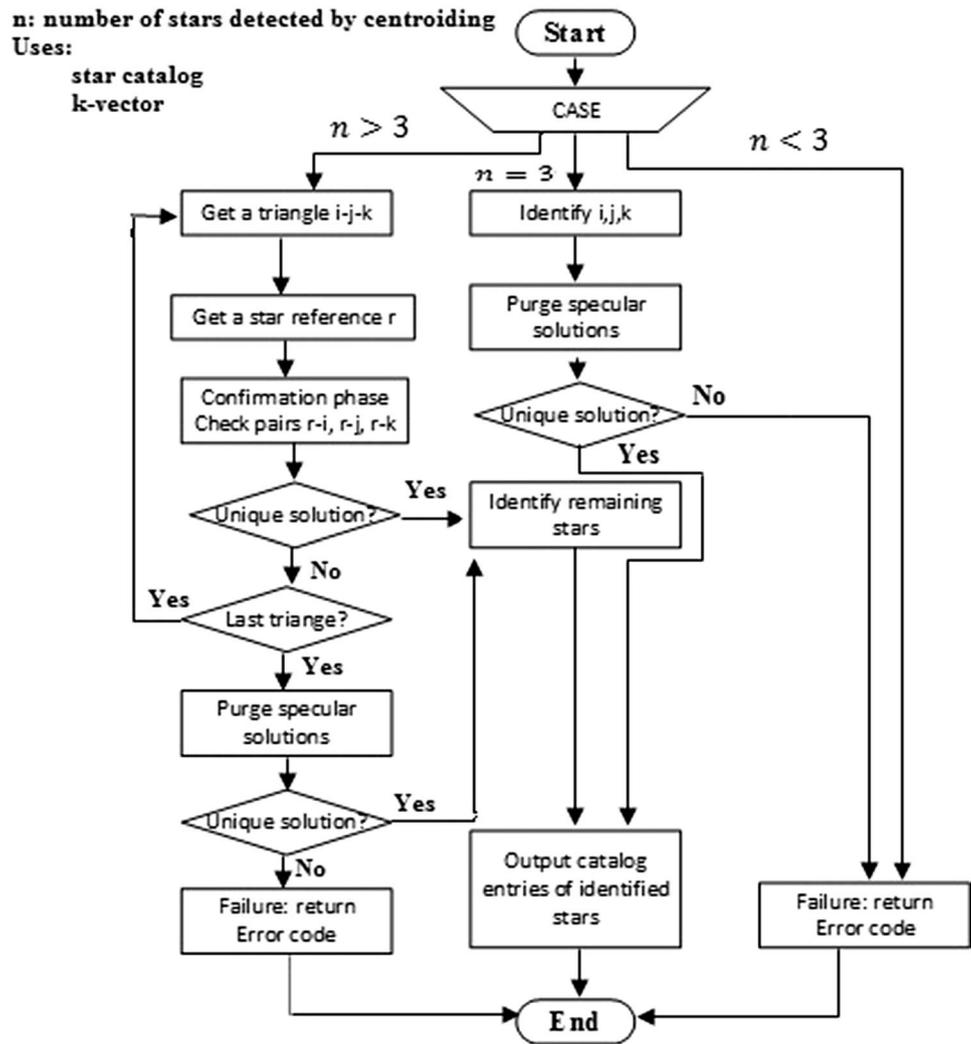


Fig. 10 Basic star triangle and pyramid [19]

in the camera FOV over the whole sky. The star pairs are ordered with increasing interstar angle. The data stored are the *k* index, the cosine of the interstar angle and the master catalog indices *I*[*k*] and *J*[*k*] of the *k*-th star pair. The *k*-vector access logic is invoked in real time for a minimal set of star pairs in elementary measured star polygons (three for a triangle, six for a four-star pyramid, etc.); the fact that the vertices between adjacent measured star pairs share a common cataloged star is the key observation leading to logic for identifying the stars efficiently by simply comparing the *k*-vector accessed catalog indices for the several sets of candidate star pairs (which must contain the common measured pivot star if it is in the catalog) [19]. The method, depicted in Fig. 11, essentially accomplishes the task by the following steps (where *n* is the number of observed stars).

Fig. 11 Pyramid flowchart [19]



We assumed the field of view is 8×8 . Then, with each of three stars of catalog, that can be neighbor to each other, we made a triangular and the angles of this triangular and the star identity and stored it in database. Some part of database can be seen in Table 4. The next step of algorithm is the extraction of visual image. In first step, we assume to find four stars in received image with hypothetical centers. Then, four triangles should be made by these centers and the angles of each triangle are calculated. For example for the made triangle with first three stars, we have Eq. (2).

$$\begin{aligned}
 a &= s_1 - s_2, b = s_2 - s_3, c = s_1 - s_3 \\
 \theta_1 &= \cos^{-1} \left(\frac{a \cdot b}{|a||b|} \right) \\
 \theta_2 &= \cos^{-1} \left(\frac{a \cdot c}{|a||c|} \right) \\
 \theta_3 &= \cos^{-1} \left(\frac{b \cdot c}{|b||c|} \right)
 \end{aligned}
 \tag{2}$$

Table 4 Part of database used in identification algorithm

	StarID1	StarID2	StarID3	Tetha1	Tetha2	Tetha3
1	1147	1377	3609	0.0018	2.7978	177.2004
2	2574	2908	3992	0.0030	2.6806	177.3164
3	117	2345	6113	0.0037	3.1458	176.8505
4	1126	3050	3164	0.0038	3.7642	176.2320
5	1344	3050	3164	0.0039	6.0480	173.9482
6	1745	2246	4157	0.0041	3.0735	176.9224
7	2774	3050	3164	0.0048	7.4946	172.5000
8	1262	3050	3164	0.0049	5.5631	174.4321
9	1128	3050	3164	0.0051	6.9632	173.0316
10	359	804	1312	0.0057	2.3563	177.6380
11	1410	2043	2392	0.0060	14.1846	165.8093
12	276	3050	3164	0.0066	4.4266	175.5668
13	1410	1569	2392	0.0066	10.5214	169.4720
14	1147	1377	3706	0.0079	5.1937	174.7985
15	1612	3050	3164	0.0080	11.0464	168.9450
16	3272	5675	5824	0.0083	9.6339	170.3578
17	300	311	2695	0.0085	2.3455	177.6460
18	540	657	6179	0.0086	2.5429	177.4485
19	2908	3992	6018	0.0087	12.2271	167.7642
20	300	3695	4281	0.0089	4.7298	175.2612
21	2345	5191	6113	0.0097	14.8572	165.1330

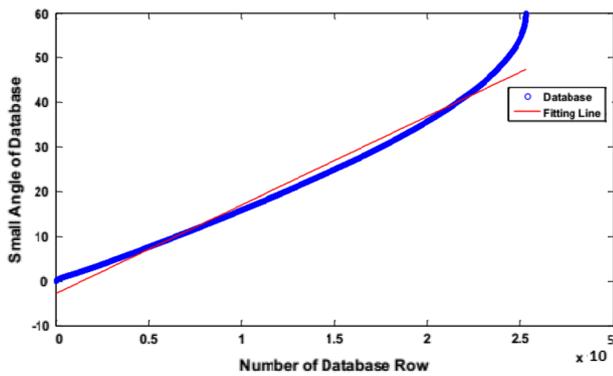


Fig. 12 Estimated line with one of the given angles of database

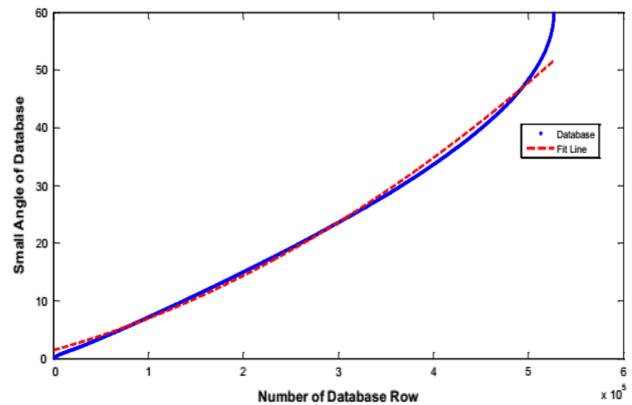


Fig. 13 Fitted quadratic curve to database

In the above, s_1, s_2 and s_3 are the centres of assumed stars. These relations occur for made triangle from first, second and fourth triangle out of first, third and fourth stars and also the triangle out of second, third and fourth. After calculation image features, we compare such features with information of database. To search the database, the technique of k -vector has been used in researches. This technique has been studied extensively in many references. [1–5]

3.2.2 Modifying searching technique of k -vector

In technique of k -vector, the database is arranged from small to large. First according to linear regression(3–5 formulas), a line has been estimated at the smallest angle of all made triangles in database and using that, the value of k can be calculated.

$$y = mx + q$$

$$\hat{y} = \hat{m}x + \hat{q} \tag{3}$$

where second equation is the estimation of first based on following formulas.

$$\hat{m} = \frac{(\sum \hat{y}) \cdot (\sum x^2) - (\sum x)(\sum x\hat{y})}{n \cdot (\sum x^2) - (\sum x)^2} \tag{4}$$

$$\hat{q} = \frac{n \cdot (\sum x\hat{y}) - (\sum x)(\sum \hat{y})}{n \cdot (\sum x^2) - (\sum x)^2} \tag{5}$$

where \hat{m} and \hat{q} are the estimation of m and q in first equation in (3) based on measurable data which \hat{y} and x are output and input, respectively, and n is number of iterations.

Blue line in Fig. 12 is database information, and the red is estimated of it, based on below.

$$\hat{y} = \hat{m}x + \hat{q} = 9.517 \times 10^{-5}x + 2.9214 \tag{6}$$

As it can be seen from Fig. 12, estimated line in some parts has a great distance and we have to search a great part of database to find the image features. To overcome this problem and reduce the searching time, according to Fig. 13, quadratic regression is more suitable. The equation of evaluated quadratic curve to database is obtained based on relations (7–9).

$$y = p_1x^2 + p_2x + p_3$$

$$\hat{y} = \hat{p}_1x^2 + \hat{p}_2x + \hat{p}_3 \tag{7}$$

As you know, \hat{y} is measurable output and estimated of y based on below:

$$\hat{p}_1 = \frac{\sum (x^2\hat{y}) \sum x^2 - \sum (x\hat{y}) \sum x^3}{\sum x^2 \sum x^4 - (\sum x^3)^2}$$

$$\hat{p}_2 = \frac{\sum (x\hat{y}) \sum x^4 - \sum (x^2\hat{y}) \sum x^3}{\sum x^2 \sum x^4 - (\sum x^3)^2} \tag{8}$$

$$\hat{p}_3 = \frac{\sum \hat{y}}{n} - \hat{p}_2 \frac{\sum x}{n} - \hat{p}_1 \frac{\sum x^2}{n}$$

Now, we estimate \hat{y} using measured data and we have:

$$\hat{y} = -5.8 \times 10^{-7}x^2 + 1.4x + 7.1 \times 10^4 \tag{9}$$

The angles are calculated by image, and the smallest angle detected is the base of searching range.

$$\Delta_1 = p_2^2 - 4p_1(p_3 - \theta + \epsilon)$$

$$\Delta_2 = p_2^2 - 4p_1(p_3 - \theta - \epsilon) \tag{10}$$

After calculation of image features, we determine the range of database for searching with the help of the smallest angle.

In relation (10), $\hat{\theta}$ the smallest angle of triangle and p_1 , p_2 and p_3 is the coefficients of evaluated curve based on Eq. (9). If the calculated angle accuracy is ε , two values of k_{bot} and k_{top} are calculated as Eq. (11). [5]

$$k_{bot} = \text{floor}\left(\frac{-p_2 + \sqrt{\Delta_1}}{2p_1}\right)$$

$$k_{top} = \text{ceil}\left(\frac{-p_2 + \sqrt{\Delta_2}}{2p_1}\right) \tag{11}$$

The functions of *floor* and *ceil* return the smaller and greater integers than their arguments. Therefore, the searching range of stars can be obtained by (12). In this relation, ID_{start} and ID_{end} are the beginning and ending of searching range in database.

$$ID_{start} = K(k_{bot}) + 1$$

$$ID_{end} = K(k_{top}) \tag{12}$$

To compare conventional technique with proposed technique, the searching range in database in Table 5 is given.

The reduction in three times in searching range results in faster applying of detecting algorithm, and as you can see in Table 6, the time of running algorithm has been reduced by 14 times.

3.3 Attitude determination algorithm

After finding the stars of image, the result would be a set of vector in sensor coordination and a set of vector in inertia coordination. Through attitude determination algorithm, the transfer matrix between this two coordination, which is the same as attitude matrix is calculated. So in this step, we need two sets of vectors. To find the vector set in coordination of sensor, after centroiding algorithm according to relation (13) for each star of image, a unit vector is obtained.

Table 5 The comparison of two techniques in database

Searching method	Start of interval	End of interval	Length of interval
<i>k</i> -vector	1	165,380	165,379
Suggested Method	112,136	163,650	51,514

Table 6 The comparison of time of running detecting algorithm with two searching techniques

Searching method	Time consumption for identification (ms)
<i>k</i> -vector	0.228295
Suggested method	0.01581

$$W_i = \frac{1}{\sqrt{f^2 + (x_o - x_i)^2 + (y_o - y_i)^2}} \begin{bmatrix} \mu_l(x_o - x_i) \\ \mu_w(y_o - y_i) \\ f \end{bmatrix} \tag{13}$$

In this relation, x_i and y_i are the coordination of center of *i*th star in the image, f is the focal distance of camera lenses in meter, x_o and y_o are the line coordination of camera's vision, μ_l is the pixel length of detector, and μ_w is the pixel width of detector. The coordination of camera's vision line is intersection between focal lens vector and detector. Figure 14 clarifies this issue. The coordination of vision line for a CCD or CMOS is the center of sensor ideally, yet there is always error which should be improved through calibration on the ground and then on-orbit. [18]

Although a single coarse calibration on the ground is deemed enough for a number of emerging applications in micro- and nanosatellite platforms, the case is really different for evolved spacecraft [18]. To determine a first and valid approximation for the focal length (f) and offset (x_o, y_o) values, we used proposed method explained in [18] that using of least square method solves ground calibration problem. It should be mentioned that for on-orbit calibration, we need statistical approach for estimation such as kalman method that is investigated with details in [18]. These estimation methods use ground calibration result as initial assumption. We have focused on ground calibration because we intended to design a star tracker for a micro-satellite.

Length and width coordination of the camera's pixel that we used for taking photos has been 6 Åµm, and its focal distance has been 41.1 mm. The set of vectors in sensor coordinate will be a 3 × 4 matrix. The set of vectors in inertia coordinate will be obtained according to identification algorithm and star catalog. For example, if *j*th star is identified in image, the features of this star such as RA and Dec can be obtained. Therefore, we have a unit vector for each identified star in form of relation (14). [8]

$$V_j = \begin{bmatrix} \cos \alpha_j \cos \delta_j \\ \sin \alpha_j \cos \delta_j \\ \sin \delta_j \end{bmatrix} \tag{14}$$

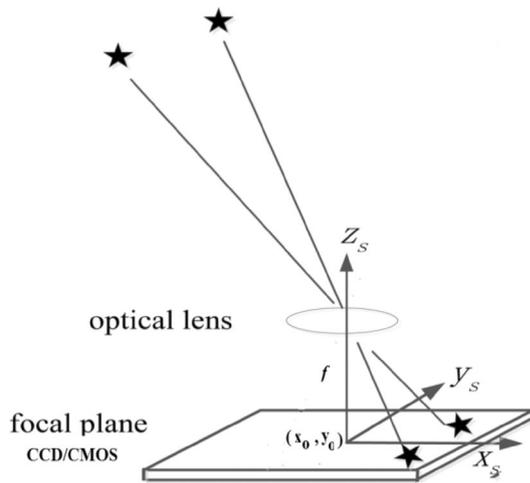


Fig. 14 How coordination of camera's line vision id found

In this relation, α_j is the RA and δ_j is the j th Dec in earth-centered inertial coordinate. The output of attitude determination algorithm is transfer matrix or quaternions that will be explained in following of quaternions.

3.3.1 Quaternions

In all attitude determination algorithms, the aim is finding matrix A so that the defined expense function L minimizes in form of relation (15). [2]

$$L(A) = \frac{1}{2} \sum_{i=1}^k a_i |W_i - AV_i|^2 \tag{15}$$

In this equation, W_i and V_i are unit vectors of stars of image and unit vectors of catalog stars and a_i are the positive weight coefficients of these vectors.

If $e = [e_1 \ e_2 \ e_3]^T$ is the eigenvector of the largest eigenvalue of transfer matrix A and α is the rotation angle, the quaternions are defined based on following relation (16). [2]

$$[q_1 \ q_2 \ q_3 \ q_4]^T = [e_1 \sin \frac{\alpha}{2} \ e_2 \sin \frac{\alpha}{2} \ e_3 \sin \frac{\alpha}{2} \ \cos \frac{\alpha}{2}]^T \tag{16}$$

The transfer matrix can be obtained in form of relation (17) by quaternions.

$$A = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_1q_2 - q_3q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2q_3 + q_1q_4) \\ 2(q_1q_3 + q_2q_4) & 2(q_2q_3 - q_1q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} \tag{17}$$

3.3.2 Wahba cost function

Function (15) was proposed by Chris Wahba for the first time in 1965. Also, it can be proved that cost function can be rewritten in form of relation (18). [2]

$$L(A) = \lambda_0 - \text{trace}(AB^T) \tag{18}$$

In above equation, λ_0 and B are calculated by relations of (19).

$$\lambda_0 = \sum_i a_i \ \& \ B = \sum_i a_i b_i r_i^T \tag{19}$$

To minimize expense function L, trace must be maximized and to do so matrices A and B should approach to each other.

Speed and accuracy are two important parameters in comparison with algorithms. The speed of algorithm is calculated through processing load calculation. In past, when calculation of locating was done by slow processors, speed was significant. However, nowadays, through powerful and fast processors accuracy is the determining parameter for choosing algorithm. According to probes in this paper, the algorithms of q-method, SVD and foam are the most accurate ones. Since these algorithms have the same level of accuracy, considering the applicability of algorithm in produced sensors is significant. Therefore, algorithm q-method is our selected algorithm in this project that its mechanism will be explained.

3.3.3 Q-method algorithm

Davenport proposed the first useful technique to provide solution to Wahba's problem. The output of this algorithm is quaternion matrix that can be calculated through transfer matrix according to relation (17). By assumption $q = [q_1 \ q_2 \ q_3]^T$ through relation (16), quaternion matrix can be defined in the form of relation (20) and transfer matrix will be in the form of Eq. (21). [2]

$$\tilde{q} = \begin{bmatrix} q \\ q_4 \end{bmatrix} = \begin{bmatrix} e \sin \frac{\alpha}{2} \\ \cos \frac{\alpha}{2} \end{bmatrix} \tag{20}$$

$$A(q) = (q_4^2 - |q|^2)I + 2qq^T - 2q_4Q \tag{21}$$

Where in relation (22), Q is obtained through (17).

$$Q = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix} \tag{22}$$

It has been said that to minimize Wahba cost function, $\text{trace}(AB^T)$ should be maximized. Furthermore, we have:

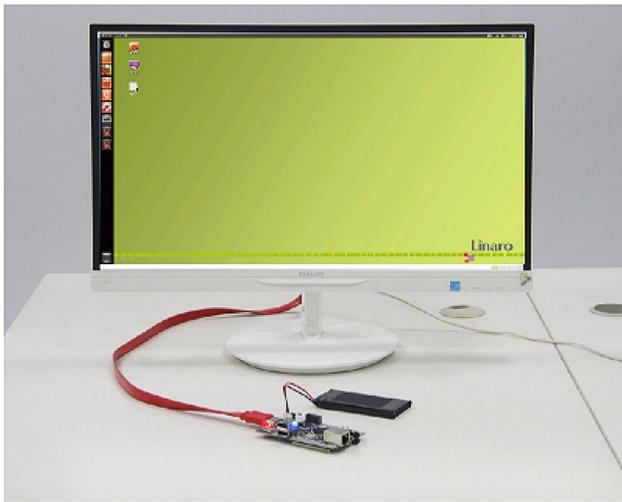


Fig. 15 Laboratory test system

$$\text{trace}(AB^T) = q^T K q \tag{23}$$

In relation (23), matrix K is defined as (24) in which matrices B, S, Z are calculated based on relations of (20). [2]

$$K = \begin{bmatrix} S - \text{trace}(B) \times I & Z \\ Z^T & \text{trace}(B) \end{bmatrix} \tag{24}$$

$$B = \sum_i a_i b_i r_i^T S = B + B^T \ \& \ Z = \begin{bmatrix} B_{23} - B_{32} \\ B_{31} - B_{13} \\ B_{12} - B_{21} \end{bmatrix} \tag{25}$$

To maximize relation (25) and also because of $|q| = 1$, the optimal quaternion is the special proper with the largest eigenvalue of matrix K . [2]

$$K q_{opt} = \lambda_{\max} q_{opt} \tag{26}$$

This algorithm has been used for locating the satellites HEAO(1-2-3) and PoSAT.

4 Experiments and results

In this section, we design a number of experiments to verify the laboratory test system designed in this paper. In all steps of algorithms evaluation, after transforming the MATLAB codes to C/C++, first we evaluate them in Eclipse software, after cross-compiling, we run them by the processor. In Fig. 15, you can see the laboratory test system designed in this paper.

4.1 The semi-physical test of algorithms

For semi-physical test, we take experiments in real night sky. We took some images under different orientations that you can see one of them in Fig. 16. (Image resolution = 480×752) (Table 7).

The attitude for Fig. 17 in three axes of Roll, Pitch and Yaw in relation to ECI coordinate is equal to relation (27). This image is given as an input to algorithms, and the obtained attitude in three axes is compared with correct attitude.

$$\text{correct_attitude} = \begin{bmatrix} \varphi_c \\ \theta_c \\ \psi_c \end{bmatrix} = \begin{bmatrix} 25.1254^\circ \\ -15.6286^\circ \\ 120^\circ \end{bmatrix} \tag{27}$$

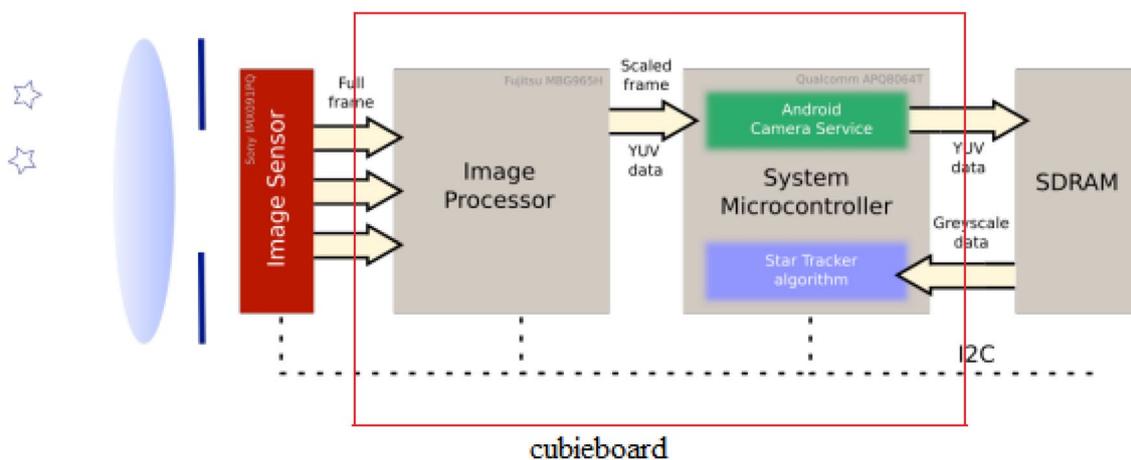


Fig. 16 Hardware in loop



Fig. 17 Input image with specific attitude to attitude determination

Table 7 Output of centroiding algorithm for Fig. 17

x_centroid (pixel)	y_centroid (pixel)
295	355
467.0254	217.0169
519.0142	112
625.0027	281.9973

4.2 Applying centroiding algorithm

Intensity parameter of threshold has been considered 60 in this algorithm.

After making the pattern with these four stars and their identifications based on modified pyramid algorithm, we find the identity of these four stars. The result is shown in Table 8. (We add the error of 0.6 degree to calculated angles (result of trial and error)).

Regarding the output of identification algorithm, the ID of four stars is 3381, 3052, 2802 and 2336.

Based on these IDs, the catalog is referred and the features of stars are extracted. Based on relations 9 and 10,

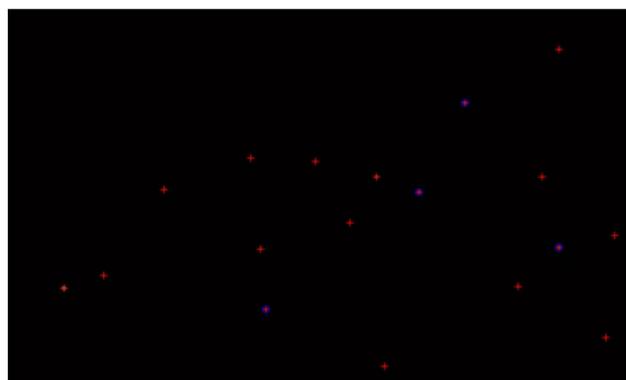


Fig. 18 Finding center of brighter stars of image

the unit vectors of image and reference vector are made and are given to algorithm q-method as input. The attitude in three axes of Roll, Pitch and Yaw that this algorithm gives us as an output equals with:

$$\text{measured_attitude} = \begin{bmatrix} \varphi_m \\ \theta_m \\ \psi_m \end{bmatrix} = \begin{bmatrix} 25.1239^\circ \\ -15.6266^\circ \\ 120.0101^\circ \end{bmatrix} \tag{28}$$

The attitude error that is obtained based on the difference between obtained and the correct attitude is as follows.

$$\begin{aligned} \text{output_error} &= |(\text{correct_attitude}) - (\text{measured_attitude})| \\ &= \begin{bmatrix} \varphi_e \\ \theta_e \\ \psi_e \end{bmatrix} = \begin{bmatrix} 0.0015^\circ \\ 0.002^\circ \\ 0.0101^\circ \end{bmatrix} \end{aligned} \tag{29}$$

In this case, the value of Wahba cost function is: $L(A) = 1.52420381 \times 10^{-10}$.

As you can see in Fig. 17, first four brighter stars of image are chosen based on adaptive centroiding algorithm and they are identified.

As mentioned in Sect. 3.1.1, each set of a four star of image can be candidates for identification and attitude determination algorithms. But as we mentioned earlier, the center of brighter stars is calculated more accurately. Therefore, considering brighter stars based on an adaptive approach in centroiding algorithm is significant for identification and attitude determination algorithms and affects the accuracy of attitude and determining value of cost function. To show this issue for Fig. 17, we repeat the

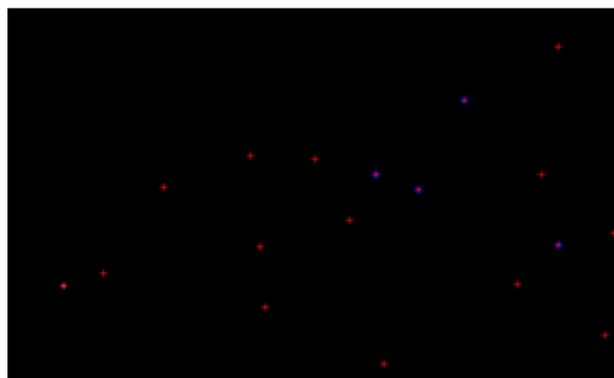


Fig. 19 Finding centers of stars of selected set based on labels

Table 8 The output of identification algorithm of Fig. 18

Output		
2336	2802	3052
2336	3052	3381
2336	2802	3381

Table 9 The output of finding center algorithm of Fig. 19

x_centroid (pixel)	y_centroid (pixel)
418.9203	198.8781
467.0254	217.0169
519.0142	112
625.0027	281.9973

Table 10 The output of detecting algorithm of Fig. 19

Output		
1934	3052	3381
1934	3052	2802
1934	3381	2802

algorithms. However, this time is based on the labels that are given to stars in centroiding algorithm. In fact, stars with label 1, 2, 3 and 4 are candidates for applying algorithms and the results are compared with previous case (Tables 9, 10).

Regarding the output of identification algorithm, the identities of four stars are 1934, 3052, 3381 and 2802.

Like previous step, with referring to catalog, right ascension and declination are obtained and the vectors of image and reference are obtained and are given to locating algorithm of data as input and the status is calculated in three axes.

$$\text{measured_attitude} = \begin{bmatrix} \varphi_m \\ \theta_m \\ \psi_m \end{bmatrix} = \begin{bmatrix} 25.122^\circ \\ -15.625^\circ \\ 120.021^\circ \end{bmatrix}$$

In this case, the value of Wahba cost function equals with: $L(A) = 5.24253951 \times 10^{-7}$.

The comparison between the present error and its previous value shows that present value has become two times greater than and the value of cost function in the second mode has multiplied by 3000. For star sensor, this value is signification in attitude so that using the adaptive centroiding algorithm is the optimal approach.

5 Conclusions

In this research, a laboratory design of star tracker and applying of its algorithms has been considered to attitude determination of satellite in LIS mode and the results were tested on a powerful ARM processor. Such algorithms include centroiding, pattern recognition and attitude

determination algorithm. In each case after searching, the best algorithm was chosen based on its practical use and it was turned to (C/C++) language to apply on a processor and the results were viewed. COM algorithm by parallel technique, modified pyramid algorithm and Q-method algorithm was the selected algorithms in centroiding, pattern recognition and attitude determination, respectively. Among these three algorithms, the pattern recognition algorithm needs the highest time of processing. Here by the change in searching technique that includes fitted quadratic curve instead of a line on database, we reduce the running time of algorithm by 14 times. In field of locating, it could be seen that choosing brighter stars of image significantly affects the accuracy and value of Wahba's cost function. Therefore, by an adaptive approach in centroiding algorithm, four brighter stars of image are chosen and determined attitude based on them.

Funding This study was funded by Iran University of Science and Technology.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

- Shankaraman R, Lourde M (2013) An attitude control of a 3-axis stabilized satellite using adaptive control algorithm. In: Proceedings of international conference on system of systems engineering. Los Angeles, pp 282–287
- Hou B, He Z, Zhou H, Zhou X, Sun B, Xu S, Wang J (2018) SINS/CNS integrated navigation system for ballistic missile based on maximum correntropy kalman filter. In: Annual American control conference (ACC) June 27–29
- Rawashdeh SA, Lump J (2014) Image-based attitude propagation for small satellite using RANSAC. IEEE Trans Aerosp Electron Syst 50(3):1864–1875
- Nikkhah AA, Nobari JH, Rad AM (2014) Optimal attitude and position determination by integration of INS, star tracker, and horizon sensor. IEEE Trans Aerosp Electron Syst 29(4):20–33
- de Galende M, Carvalho F (2014) Star tracker Orientation Optimization using Non-dominated Sorting Genetic Algorithm (NSGA). In: IEEE aerospace conference, Big Sky, MT, pp 1–8
- Gou B, Cheng Y-M, Li S, Mu H-L (2018) INS/CNS integrated navigation method for circular orbiting satellite. Chinese Automation Congress (CAC)
- Pham MD, Low KS, Chen S (2012) An autonomous star recognition algorithm with optimized database. IEEE Trans Aerosp Electron Syst 49(3):1467–1475
- Xinguo W (2014) Exposure time optimization for highly dynamic star trackers. Int J Sens 14(3):4914–4931
- Zhang G (ed) (2017) Processing of star catalog and star image. In: Star identification Methods techniques and algorithms. Springer, New York, pp 37–71

10. Roshanian J, Yazdani S, Barzamini F (2018) Application of PIV and delaunay triangulation method for satellite angular velocity estimation using star tracker. *IEEE Sens J* 18:10105
11. Aranda LA, Reviriego P, Toral RG, Maestro JA (2018) Protection scheme for star tracker images. *IEEE Trans Aerosp Electron Syst* 55(1):486
12. Yang J, Liang B, Zhang T, Song J, Song L (2012) Laboratory Test System Design for Star Sensor Performance Evaluation. *Journal of Computers* 7(4):1056–1063
13. Tappe J, Kim JJ, Jordan A, Agrawal B (2011) Star tracker attitude estimation for an indoor ground based spacecraft simulator. In: AIAA conference on modeling and simulation technologies, Portland, Oregon
14. Rufino G, Accardo D, Grassi M, Fasano G, Renga A, Tancredi U (2013) Real-time hardware-in-the-loop tests of star tracker algorithms. *Int J Aerosp Eng* 2013. <https://doi.org/10.1155/2013/505720>
15. Yahui S, Yingying X, Yunhai G (2002) Autonomous on-orbit calibration approach for star tracker cameras. *Adv Astronaut Sci* 112:39–57
16. Shashikala BK, Rao TH (2013) Design of very low noise amplifier for high accuracy star tracker in GEO missions. In: International conference on advanced electronic systems, pp 83–87
17. Jovanovic I, Enright J (2017) Towards star tracker geolocation for planetary navigation. In: IEEE aerospace conference
18. Medaglia E (2016) Autonomous on-orbit calibration of a star tracker. In: IEEE metrology for aerospace
19. Mortari D, Samaan M, Bruccoleri C, Junkins JL (2004) The pyramid star identification technique. *Navigation* 51(3):171–184

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.