2020 年美国数学建模竞赛电子科技大学模拟赛

承诺书

我们仔细阅读了数学建模竞赛的竞赛规则.

我们完全明白，在竞赛开始后参赛队员不能以任何方式（包括电话、电子邮件、网上咨询等）与队外的任何人（包括指导教师）研究、讨论与赛题有关的问题。

我们知道，抄袭别人的成果是违反竞赛规则的，如果引用别人的成果或其他公开的资料(包括网上查到的资料)，必须按照规定的参考文献的表述方式在正文引用处和参考文献中明确列出。

我们郑重承诺，严格遵守竞赛规则，以保证竞赛的公正、公平性。如有违反竞赛规则的行为，我们将受到严肃处理。

我们的题目编号是（填写：A 或者 B）： B

我们报名队伍号是： L142

参赛队员姓名学号（中文填写打印并签名）：1. 张镕麒

2. 徐雯婕

3. 唐浩

指导教师或指导教师组负责人（有的话填写）：

是否愿意参加 2019 年美国赛（是，否）： 是 日期：

2019 年 11 月 25 日

Abstract

The middle star sensor is a kind of attitude sensitive instrument with high accuracy and high reliability, which takes the star as the measuring object. In this paper, by analyzing the principle of determining the attitude of the star sensor, applying geometric relations and carrying out multiple sets of repeated random tests through the program, the method of finding the optimal orientation of the sensor satisfying the given conditions has found the solution method of the center position of the photosensitive surface of the star sensor. At the same time, a perturbation error analysis model is established to investigate the influence of optical axis deviation on the position information accuracy of the point. The optimization method of star sensor selection is proposed. The traditional star sensor is mainly used for feature extraction with angular distance, but it has the disadvantages of large storage space, poor real-time performance and low recognition rate. The accuracy of star sensor's attitude calculation is improved.

Keywords: computer simulation test, rigid body rotation, disturbance error analysis, feature extraction algorithm.

CONTENT

1.Introduction

1.1 Promblem Background

Attitude measurement is an important core technology of vehicle flight, which is of great significance to ensure the accurate entry and transformation of vehicle into orbit, high-performance flight, reliable ground communication, high-precision ground observation, and successful completion of various space missions.

Star sensor is the core part of autonomous attitude measurement of vehicle. It is a high-precision attitude measurement of vehicle in orbit by observing stars in space and using several star vectors. Because the star sensor utilizes the astronomical information of stars, it has the characteristics of good autonomy, high accuracy and reliable operation, and has a broad application prospect in space flight. Generally speaking, star sensor contains at least two working mods, namely initial attitude establishment mode and tracking mode. At the initial moment when the star sensor enters the working state or in the case of Lost in Space due to fault, the star sensor turns into the initial attitude acquisition mode. At this stage, since there is no prior attitude information, all sky star map identification is needed. Once the initial attitude is obtained, the star sensor enters tracking mode. All-day autonomous star map recognition is a key technology in star sensor technology, and it is also the focus and difficult of research. In astronomical navigation, celestial body sensor is usually used to realize the observation of natural celestial bodies. Since the spatial position of stars in the reference coordinate system can be considered fixed and the starlight vector has a high measurement accuracy, the star sensor can achieve a high attitude measurement accuracy.

The information available in the star map image is the position coordinates of the stars and the brightness of the stars, which correspond to the coordinates (right ascension and declination) and magnitude of the navigation stars in the celestial coordinate system in the star chart. It is essential for a star to glow on its own. The brightness of a star is the apparent brightness observed from earth. It is not only related to the luminescence nature of the star (temperature and size),but also depends on the distance from the earth. In astronomy, the brightness of a star is indicated by its magnitude, and the lower the magnitude, the brighter it is. The magnitude difference is 1,the brightness difference is 2.512 times, the brightness of the star is 100 times of the brightness of the star.

1.2 Reststement of the Problem

We are required to create an objective quantitative algorithm to establish a celestial coordinate system to find the position of point D. And also we should use mathmatical model to find the precision of the position and the solution we have used. Then we use assumption method and geometric model to identify the star map and determine the star number corresponding to each star image in the four star maps given in the question.

In order to solve those problems, we will proceed as follows:

Build an algorithm in order to find the point's positions.

Discuss the errors and the precision

Construction feature extraction algorithm and feature matching algorithm

2 Assumptions and Notations

2.1 Assumptions

• Assume that the data used is true and valid.

• Assume that the Celestial Navigation's data indicators do not change over time.

2.2 Notations

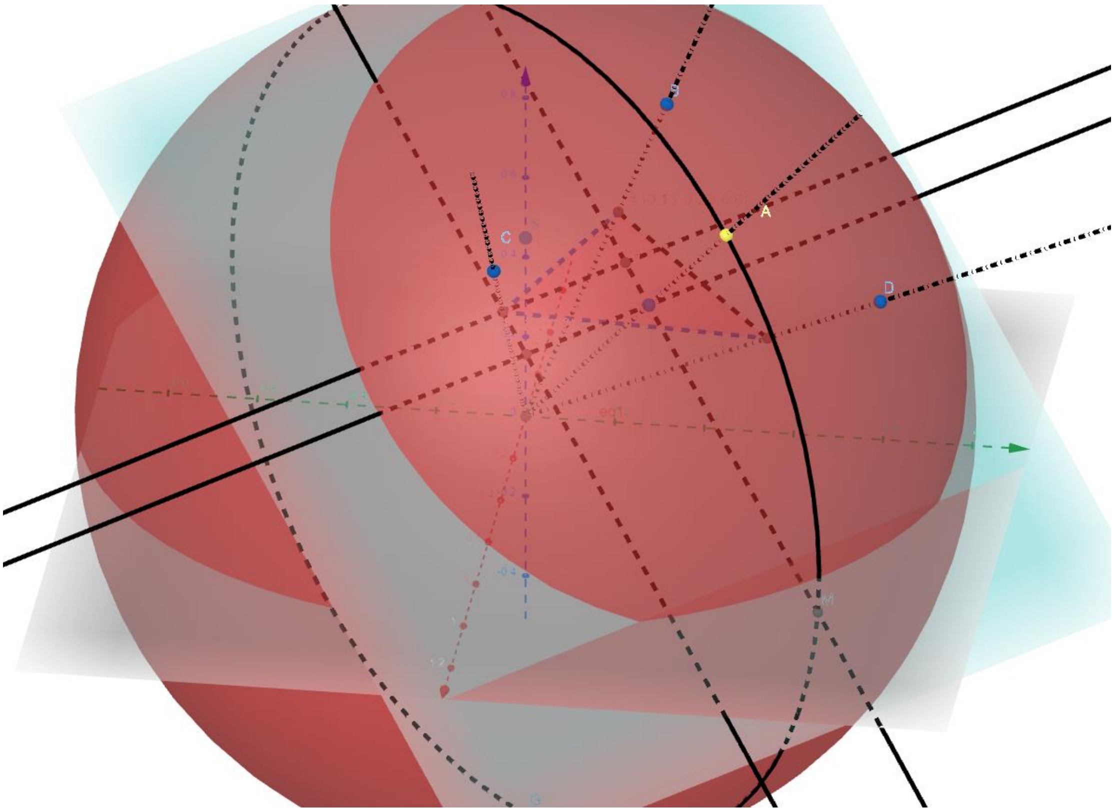
The primary notations used in this paper are listed in Table 1.

Table 1

Symbol	Definition
f	The distance between pathway and photosensitive center
α_i	Star right right ascension
δ_i	Star declination
a_i	The distance between the star image point and the center of the photosensory surface
x_i, y_i, z_i	Star coordinate
A_0	Equation coefficient matrix
X	star coordinate solution
R	Radius of the celestial sphere, 1 by default
R_i	Matrix coefficient of rigid body rotation
A	attitude matrix
W	the direction vector matrix of the observation star in the star sensor coordinate system
V	the direction vector matrix of the navigation star in the celestial coordinate system
r_i	the appropriate direction of observation star
b_i	the direction vector of navigation star
d	Screen length of star sensor
dx, dy	Pixel length of star sensor

3.Problem One

3.1 Problem Analysis



By studying the basic principle of star sensor, we can assume that the origin of star sensor coordinate system is equivalent to that of celestial coordinate system, which is obviously reasonable, because the distance of stars is too large to ignore the distance of observation points when looking at the same star in the same direction as the earth. This simplifies the operation. The first question is that we can find out the center of the photosensory surface by means of simple geometric relation, listing the linear equations and solving the coordinates, and then by means of computer test, we can find out the center of the photosensory surface. At the same time, by analyzing the structure of star sensor, an error influencing factor is found, and the optimal solution can be found by establishing disturbance error analysis model. And in view of the biography, In the defect of unified star map recognition method, we propose an adjacent triangle feature recognition algorithm. We then match the computer star map and find the star number corresponding to the star point.

Generally, star maps' star targets image one or many pixels at CCD focal plane. For the convenience of investing, if star images less or equal one pixel, we call it point target. if more than one pixel, we call it small target or plane target. Apparently, on the condition that establish a point target make the math model general.

Suppose initial position is(x_0, y_0),the target moving speed is $v = (v_x, v_y)$, v_x, v_y are the speed on x-axis and y-axis respectively.

Then, when pixel less than one, the target parameter can be shown below:

$$I(i, j, k) = \alpha(k)\delta(x - x_0, y - y_0)$$

a(k)stands for the intensity of the target,$\delta(-,-)$is two-dimensional impulse function.

The condition of the star sensitive detector is composed of two parts: target $r(i, j, k)$and background $u(i, j, k)$.

Use homogeneous coordinates and matrix to indicate as below

$$Ax_1 + By_1 + Cz_1 = f^2$$
$$Ax_2 + By_2 + Cz_2 = f^2$$
$$Ax_3 + By_3 + Cz_3 = f^2$$

Firstly, build a star sensor coordinate system, knowing the right ascension and the declination of the three stars (that is the distance between three stars' center and the center of light-sensitive surface

$$(\alpha_i, \delta_i) \ (i = 1,2,3)$$

In the celestial coordinate system, the starlight vector's direction vector is:

$$\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} = \begin{pmatrix} cos\alpha_i cos\delta_i \\ cos\alpha_i sin\delta_i \\ sin\delta_i \end{pmatrix} \ (i = 1,2,3);$$

Based on that ,we can suppose the celestial coordinates are (x, y, z)respectively.Meanwhile,suppose one star's coordinate is $(x, y, z)$the respective starlight vector is $\overrightarrow{OP} = (x, \ y, \ z)$Then suppose the distance above is $a = a_i(i = 1,2,3)$ the point D's vector is $\overrightarrow{OP} = (A, \ B, \ C)$.According to the relation between vector quantity and the geometry , it can be inferred:

$$cos < \overrightarrow{\mathrm{OP}}, \overrightarrow{\mathrm{OD}} >= \left| \frac{\overrightarrow{\mathrm{OP}} \cdot \overrightarrow{\mathrm{OD}}}{\|\overrightarrow{\mathrm{OP}}\| \|\overrightarrow{\mathrm{OD}}\|} \right| = \frac{f}{\sqrt{f^2 + a^2}}$$

$$< \overrightarrow{\mathrm{OP}}, \overrightarrow{\mathrm{OD}} > \in [0, \frac{\pi}{2})$$

Adding the coordinates of the three stars can get:

$$Ax_i + By_i + Cz_i = f^2; \ (i = 1,2,3)$$

The augmented matrix is:

$$\begin{pmatrix} x_i & y_i & z_i & f^2 \\ x_i & y_i & z_i & f^2 \\ x_i & y_i & z_i & f^2 \end{pmatrix}$$

Make:

$$A_0 = \begin{pmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{pmatrix}, \ X = \begin{pmatrix} A \\ B \\ C \end{pmatrix}, \ F = \begin{pmatrix} f^2 \\ f^2 \\ f^2 \end{pmatrix}$$

We get:

$$A_0X = F$$

Then we can get the position of D point:

$$X=\frac{f}{|A_0|}\begin{pmatrix} y_2z_3+y_3z_1+y_1z_2-y_3z_2-y_1z_3-y_2z_1 \\ x_3z_2+x_2z_1+x_1z_3-x_2z_3-x_1z_2-x_3z_1 \\ x_2y_3+x_3y_1+x_1y_2-x_3y_2-x_1y_3-x_2y_1 \end{pmatrix}$$

That is the right ascension and declination respectively in the celestial coordinates is

$$(\alpha, \ \delta) = (arctan\frac{B}{A}, arcsin\frac{C}{f})$$

The pseudo code for the first question is provided below

3.2 Concrete Algorithm

FIND-D_WITH-F$(\{(\alpha_i,\delta_i)\},\{a_i\},f)$

1. $x_i= cos\alpha_i cos\delta_i$, $y_i= cos\alpha_i sin\delta_i$, $z_i= sin\delta_i$
2. $A_0=[x_1,y_1,z_1;x_2,y_2,z_2;x_3,y_3,z_3]$
3. A$=\frac{f}{|A_0|}(y_2z_3+y_3z_1+y_1z_2-y_3z_2-y_1z_3-y_2z_1)$

   B$=\frac{f}{|A_0|}(x_3z_2+x_2z_1+x_1z_3-x_2z_3-x_1z_2-x_3z_1)$

   C$=\frac{f}{|A_0|}(x_2y_3+x_3y_1+x_1y_2-x_3y_2-x_1y_3-x_2y_1)$
4. return $(arctan\frac{B}{A}, arcsin\frac{C}{f})$.

4.Problem Two

4.1 Problem Analysis

Due to the lack of focus f information, we can only abandon the algorithm based on the first question of triangle similarity. Inspired by similarity, we can list the following equations

$$\frac{\tan\theta_1}{a_1}=\frac{\tan\theta_2}{a_2}=\frac{\tan\theta_3}{a_3}=\frac{1}{f}$$

It can be found that the relationship among the three equations on the left side is independent of F. a randomized determination method for measuring the included angle is introduced below.

Firstly, a random point $M(\alpha_0,\delta_0)$ is taken in the hemisphere where P_1,P_2,P_3 is located, and it is transformed into (x_0,y_0,z_0) (unit vector) by the relationship between the celestial coordinate system and the space rectangular coordinate system. Divide the od vector n into equal parts, take the right endpoint f_j $(f_j < r)$ of the $j-th$

segment, and assume the actual focal length f, establish the screen plane γ. Again, with the help of the similarity relation, we can get the intersection H_i of the $\overrightarrow{OP_i}$ vector and the plane γ. The hi module length is $t_i * f_j$. We can also calculate the angle between $\overrightarrow{OM}$ and $\overrightarrow{OP_i}$ with the help of the angle formula of the vector. Because $\overrightarrow{OM}$ is perpendicular to γ, odihi is a right triangle. With the help of Pythagorean theorem, we can calculate the distance AI J of dihi. We make $\varphi(j) = |Aij - ai|$ .The approximation degree of $\overrightarrow{OM}$ vector to the known condition a_i is $min\{\varphi(j)\} = \Phi\ (M)$, we repeat this process, and record the d value M' of the minimum $\Phi\ (M)$. With the increase of repetition times $m, \lim\limits_{n\to\infty} D' = D$.

4.2 Algorithm Analysis

In the process of dividing $\overrightarrow{OM}$ vector into n sections, the larger n is, the more times m of repeating random M points is, the more accurate M' is, but at the same time, the time complexity $O\ (nm)$ of the algorithm will increase year on year.

4.3 Problems in the Algorithm

Through many tests, it is found that there is a possibility of 20% for a given set of legal conditions (i.e. the existence of solutions). M'does not tend to one solution, but jumps within the range of two solutions. Therefore, it is speculated that (α, δ) is the solution of a binary quadratic equation system composed of known conditions. At present, due to the complexity of the equation form, it is hard to get an analytical solution

In the following table, two groups tested the experiment of n = 1e7 and M = 100 respectively. In the first group, with the continuous random generation of m point, $\Phi\ (M)$ gradually decreased, and $M\ '$point also moved closer to D point. In the second group, M' is obviously close to two points, and one of which is the solution

```
a[1]=1.486505 a[2]=0.172523 a[3]=0.575728
P_1(0.026804 0.876849 0.480018 )
P_2(0.741908 0.362329 0.564172 )
P_3(0.339232 0.675068 0.655137 )
ans
D(0.892534 0.094953 0.440870 )
O'(0.446267 0.047476 0.220435 )
delta=1.927210 M(0.324456 0.849416 0.416198 )M'(0.071380 0.186872 0.091564 )
delta=0.076337 M(0.727112 0.068762 0.683067 )M'(0.479894 0.045383 0.450824 )
delta=0.064676 M(0.721120 0.191409 0.665844 )M'(0.627374 0.166526 0.579284 )
delta=0.064455 M(0.742867 0.106899 0.660849 )M'(0.527435 0.075898 0.469203 )
delta=0.058915 M(0.750530 0.124223 0.649056 )M'(0.547887 0.090683 0.473811 )
delta=0.044831 M(0.909653 0.061689 0.410763 )M'(0.391151 0.026526 0.176628 )
delta=0.022221 M(0.919901 0.113074 0.375495 )M'(0.432353 0.053145 0.176483 )
delta=0.021507 M(0.863864 0.105522 0.492549 )M'(0.483764 0.059092 0.275827 )
delta=0.006561 M(0.888906 0.097316 0.447634 )M'(0.453342 0.049631 0.228293 )
delta=0.005380 M(0.907773 0.099602 0.407465 )M'(0.435731 0.047809 0.195583 )
delta=0.004645 M(0.861103 0.075065 0.502859 )M'(0.447773 0.039034 0.261487 )
delta=0.003659 M(0.872512 0.086900 0.480803 )M'(0.453706 0.045188 0.250018 )
delta=0.003222 M(0.847512 0.068895 0.526287 )M'(0.449181 0.036514 0.278932 )
delta=0.002574 M(0.900455 0.097424 0.423898 )M'(0.441223 0.047738 0.207710 )
delta=0.002542 M(0.871310 0.085443 0.483237 )M'(0.453081 0.044430 0.251283 )
delta=0.002443 M(0.858600 0.079607 0.506427 )M'(0.455058 0.042192 0.268406 )
delta=0.001797 M(0.852113 0.073171 0.518217 )M'(0.451620 0.038781 0.274655 )
delta=0.001342 M(0.893151 0.096024 0.439387 )M'(0.446575 0.048012 0.219693 )
delta=0.001079 M(0.856193 0.077174 0.510860 )M'(0.453782 0.040902 0.270756 )
delta=0.000670 M(0.891825 0.094023 0.442502 )M'(0.445912 0.047012 0.221251 )
delta=0.000575 M(0.892645 0.095212 0.440590 )M'(0.446322 0.047606 0.220295 )
delta=0.000558 M(0.893102 0.095705 0.439556 )M'(0.446551 0.047853 0.219778 )
```

```
a[1]=0.422971 a[2]=0.059605 a[3]=0.231673
P_1(0.035085 0.661262 0.749334 )
P_2(0.399675 0.090323 0.912196 )
P_3(0.502064 0.541429 0.674378 )
ans
D(0.485632 0.153617 0.860560 )
O'(0.242816 0.076809 0.430280 )
delta=0.604643 M(0.489284 0.666284 0.562732 )M'(0.039143 0.053303 0.045019 )
delta=0.150966 M(0.261988 0.099098 0.959970 )M'(0.107415 0.040630 0.393588 )
delta=0.131631 M(0.598399 0.214055 0.772074 )M'(0.281248 0.100606 0.362875 )
delta=0.101900 M(0.402009 0.060933 0.913606 )M'(0.184924 0.028029 0.420259 )
delta=0.004095 M(0.514724 0.107555 0.850583 )M'(0.231626 0.048400 0.382762 )
delta=0.003728 M(0.504238 0.116627 0.855653 )M'(0.231949 0.053649 0.393601 )
delta=0.003622 M(0.484976 0.151443 0.861315 )M'(0.242488 0.075722 0.430657 )
delta=0.003086 M(0.528903 0.050082 0.847203 )M'(0.211561 0.020033 0.338881 )
delta=0.002120 M(0.527441 0.052675 0.847957 )M'(0.210976 0.021070 0.339183 )
delta=0.002062 M(0.478244 0.160629 0.863412 )M'(0.243905 0.081921 0.440340 )
delta=0.001925 M(0.522632 0.076069 0.849158 )M'(0.219506 0.031949 0.356646 )
delta=0.001061 M(0.525322 0.062914 0.848575 )M'(0.215382 0.025795 0.347916 )
delta=0.000804 M(0.527072 0.064149 0.847396 )M'(0.216099 0.026301 0.347432 )
delta=0.000516 M(0.486055 0.153454 0.860350 )M'(0.243028 0.076727 0.430175 )
delta=0.000373 M(0.526598 0.064046 0.847698 )M'(0.215905 0.026259 0.347556 )
delta=0.000359 M(0.525951 0.063932 0.848109 )M'(0.215640 0.026212 0.347725 )
delta=0.000177 M(0.526266 0.063742 0.847927 )M'(0.215769 0.026134 0.347650 )
```

5.Problem Three

5.1 Error Analysis

The source of the star sensor is diverse. Here, we only discuss the optical axis offset error of star sensor calibration, and in this case, we can assume that there are no other errors in star sensor, that is, control variables. Easy to know, this assumption is reasonable. From this, we can set the offset of the optical axis is $\delta_t = (t_x, t_y, t_z)$.A star coordinate in the world coordinate system is(x, y, z).After Matrix rotation, it can be got the coordinate (x', y', z')

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x + t_x \\ y + t_y \\ z + t_z \\ 1 \end{pmatrix}$$

5.2 Disturbance Error Analysis Model

According to the question, the star sensor converts the world coordinate system into a rigid body transformation matrix of the image coordinate system without error. while the rigid-body transformation only changes the spatial position (translation) and orientation (rotation) of the object without changing its shape, two variables can be used to describe: the rotation matrix r and the translation vector t: the rotation matrix r is an orthogonal matrix, The transformation from the Rodrigues transformation to a rotation vector with only three independent variables: therefore, the rigid transformation can be described with six parameters that determine the transformation of the space point from the world coordinate system to the image coordinate system In other words, these parameters describe the position and orientation of the star sensor in the world coordinate system.

Let the unit vector of star light vector be X ,then the formula of the panning transformation is

$$X' = (x', y', z') = R(x,\ y,\ z) + t\ = AX;$$

Let $B = A^{-1}$,then $B\,X' = X$；

the error generated by the optical axis offset error calibrated by the star sensor δ_t can lead to an error δ_b in the resolution of the image coordinate system,

That is, there are perturbed equations:

$$(B + \delta_t)(X' + \delta_b)\ = X;$$

That is

$$\delta_t X' + \delta_b B + \delta_b \delta_t = 0$$

It can be got

$$\delta_b = -A\delta_t X' - A\delta_t \delta_b$$

The following theorem of disturbance error analysis is given:

When δ_t is small enough，let $\|A^{-1}\|\|A\| < 1$,Then there is an error estimate. ：

$$\frac{\|\delta_b\|}{\|X'\|} = \frac{\|A^{-1}\|\|A\|}{1-\|A^{-1}\|\|\delta_t\|}\frac{\|\delta_t\|}{\|A\|};$$

When $\|A^{-1}\|\|\delta_t\|$ is very small，The above formula can be approximately expressed as follows:

$$\frac{\|\delta_b\|}{\|X'\|} \approx \|A^{-1}\|\|A\|\frac{\|\delta_t\|}{\|A\|}$$

5.3 Model Analysis

When the star sensor is determined, the optical axis offset error of the star sensor calibration has been determined. That is $\|\delta_t\|$ is determined.At the same time, the coordinate transformation matrix is only related to the world coordinate system and image coordinate system. In the idea of controlling variables, it can be guaranteed that it will not change. That is $\|A^{-1}\|\|A\|$ is not changeable.

Then it is derived from the relational formula that the size of $\|\delta_b\|$ is only related to

$\|X'\|$. That is, the error of the solution of the image coordinate system in the projection coordinates of the star starlight vector on the photosensitive surface is only determined by the size of $\|X'\|$.Therefore, the position information in the final object coordinate system is only determined by the projection error formed by the star vector of the three stars in the image coordinate system. These three errors are respectively $\|\delta_b\|_1$, $\|\delta_b\|_2$, $\|\delta_b\|_3$.

Because the offset error of the optical axis calibrated by the star sensor is small, we can think that the error has a proportional coefficient k with the n power of the tangent angle between the direction vector of the three stars and the optical axis.
At the same time, by geometric relations.

$$\frac{\tan\theta_1}{a_1}=\frac{\tan\theta_2}{a_2}=\frac{\tan\theta_3}{a_3}=\frac{1}{f}$$

Define the position information error as

$$\begin{aligned} E&=\|\delta_b\|_1+\|\delta_b\|_2+\|\delta_b\|_3 \\ &=k\left(\tan^n\theta_1+\tan^n\theta_2+\tan^n\theta_3\right) \\ &=\frac{k}{f^n}\left({a_1}^n+{a_2}^n+{a_3}^n\right); \end{aligned}$$

The angle relation corresponding to the minimum value is obtained from the solution of Qin Sheng inequality

$$\theta_1=\theta_2=\theta_3;$$

Therefore, the optimal solution can be obtained when the angle between the starlight vector and the optical axis of the three stars is equal, and the error is the smallest.

6.Problem Four

6.1 Problem Analysis

When there are more than three stars in the field of view of the star sensor, the number is set as n. At this time, if an eigenvector is constructed with two ambiguity, the number of vector bars can be constructed as C_n^2. According to this, the number of vectors can be increased to the quadratic power of the number of stars.

However, in practice, when the number of stars used for positioning is too high, the increase of data volume is bound to seriously affect the recognition speed and recognition rate of star map recognition. Therefore, a balance should be taken between the two.

7.Problem Five

7.1 Problem Analysis

First of all, the imaging process of stars on the sensitive surface can be represented by perspective projection transformation. After perspective projection, the coordinates of star imaging points are

In conclusion, the imaging process of star sensor can be summarized as follows:

$$\begin{cases} x = -f\dfrac{a_1l_1+b_1l_2+c_1l_3}{a_3l_1+b_3l_2+c_3l_3} \\ y = -f\dfrac{a_2l_1+b_2l_2+c_2l_3}{a_3l_1+b_3l_2+c_3l_3} \\ \overline{w} = \dfrac{1}{\sqrt{x^2+y^2+f^2}}\begin{pmatrix} x \\ y \\ f \end{pmatrix} \end{cases}$$

And the imaging model of star sensor can be represented by homogeneous coordinates and matrix as follows:

$$\begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{dx} & 0 & u \\ 0 & \frac{1}{dy} & v \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} R_1 & R_2 & R_3 \\ R_4 & R_5 & R_6 \\ R_7 & R_8 & R_9 \end{pmatrix} \cdot \begin{pmatrix} cos\alpha cos\delta \\ cos\alpha sin\delta \\ sin\delta \end{pmatrix}$$

Next, in the construction of features, a navigation star is taken as the main star, and the surrounding three stars are taken as the neighboring stars to form the features of the navigation star. Due to the different positions of the three stars around the different navigation stars, the characteristics of the navigation stars are unique. The direction vectors of the four stars in the celestial coordinate system can form a matrix V. The observation stars corresponding to these four navigation stars can also construct the matrix W according to the similar method, with $W^TW = V^TV$, that is, whether in the celestial coordinate system or in the image coordinate system, the symmetry matrix constructed by the same group of stars remains unchanged, so this can be regarded as the feature of star map recognition.
This is the proof.

Let the right ascension and declination coordinates of the navigation star be (α, δ), in the celestial coordinate system, its direction vector is

$$\begin{pmatrix} cos\alpha cos\delta \\ cos\alpha sin\delta \\ sin\delta \end{pmatrix}$$

and the relationship between the star vector and the star sensitive coordinate system is $W = AV$, where W is the direction vector matrix of the observation star in the star sensor coordinate system, A is the attitude matrix, and V is the direction vector

matrix of the navigation star in the celestial coordinate system. When four stars are observed, $W = (b_1 \quad b_2 \quad b_3 \quad b_4)$, $V = (r_1 \quad r_2 \quad r_3 \quad r_4)$, where r_i is the appropriate direction of observation star and b_i is the direction vector of navigation star. $(i = 1, 2, 3, 4)$. W is a matrix of 3 * 4, $W^T W$ is a matrix of 4 * 4.
According to $W = AV$

$$(b_1 \quad b_2 \quad b_3 \quad b_4) = A(r_1 \quad r_2 \quad r_3 \quad r_4) \tag{1}$$

$$W^T W = (b_1 \quad b_2 \quad b_3 \quad b_4)^T (b_1 \quad b_2 \quad b_3 \quad b_4)$$

$$= \begin{pmatrix} b_1^T b_1 & b_1^T b_2 & b_1^T b_3 & b_1^T b_4 \\ b_2^T b_1 & b_2^T b_2 & b_2^T b_3 & b_2^T b_4 \\ b_3^T b_1 & b_3^T b_2 & b_3^T b_3 & b_3^T b_4 \\ b_4^T b_1 & b_4^T b_2 & b_4^T b_3 & b_4^T b_4 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & b_1^T b_2 & b_1^T b_3 & b_1^T b_4 \\ b_2^T b_1 & 1 & b_2^T b_3 & b_2^T b_4 \\ b_3^T b_1 & b_3^T b_2 & 1 & b_3^T b_4 \\ b_4^T b_1 & b_4^T b_2 & b_4^T b_3 & 1 \end{pmatrix} \tag{2}$$

$$W^T W = V^T A^T A V = V^T V \tag{3}$$

$$V^T V = (r_1 \quad r_2 \quad r_3 \quad r_4)^T (r_1 \quad r_2 \quad r_3 \quad r_4)$$

$$= \begin{pmatrix} r_1^T r_1 & r_1^T r_2 & r_1^T r_3 & r_1^T r_4 \\ r_2^T r_1 & r_2^T r_2 & r_2^T r_3 & r_2^T r_4 \\ r_3^T r_1 & r_3^T r_2 & r_3^T r_3 & r_3^T r_4 \\ r_4^T r_1 & r_4^T r_2 & r_4^T r_3 & r_4^T r_4 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & r_1^T r_2 & r_1^T r_3 & r_1^T r_4 \\ r_2^T r_1 & 1 & r_2^T r_3 & r_2^T r_4 \\ r_3^T r_1 & r_3^T r_2 & 1 & r_3^T r_4 \\ r_4^T r_1 & r_4^T r_2 & r_4^T r_3 & 1 \end{pmatrix} \tag{4}$$

Since b_i and r_i are unit vectors, the diagonal elements of matrices in formulas (1) and (4) are all (1). From formula (2), formula (3) and formula (4), we can get:

$$\begin{pmatrix} 1 & b_1^T b_2 & b_1^T b_3 & b_1^T b_4 \\ b_2^T b_1 & 1 & b_2^T b_3 & b_2^T b_4 \\ b_3^T b_1 & b_3^T b_2 & 1 & b_3^T b_4 \\ b_4^T b_1 & b_4^T b_2 & b_4^T b_3 & 1 \end{pmatrix} = \begin{pmatrix} 1 & r_1^T r_2 & r_1^T r_3 & r_1^T r_4 \\ r_2^T r_1 & 1 & r_2^T r_3 & r_2^T r_4 \\ r_3^T r_1 & r_3^T r_2 & 1 & r_3^T r_4 \\ r_4^T r_1 & r_4^T r_2 & r_4^T r_3 & 1 \end{pmatrix} \tag{5}$$

The elements of the matrices at both ends of equation (5) are equal. Because they are all symmetric matrices, there are

$$b_1^T b_2 = b_1^T b_2, b_1^T b_3 = r_1^T r_3,$$
$$b_1^T b_4 = r_1^T r_4, b_2^T b_3 = r_2^T r_3,$$
$$b_2^T b_4 = r_2^T r_4, b_3^T b_4 = r_3^T r_4$$

That is to say, a group of stars in the celestial coordinate system, their direction vectors are multiplied by each other. After transforming to the star sensor coordinate

system, the product results remain unchanged. Therefore, these products can be extracted as the features of recognizable graphs. Since 4 stars are selected to construct the eigenvector, 6 elements in the matrix constructed according to formula (2) and (4) are independent, which can be constructed into the eigenvector of the star, namely
So far, the proof of invariance of eigenvectors in different coordinate systems is completed

7.2 Feature Extraction

Therefore, the next step is to build a feature vector with a navigation star as the main star and three surrounding stars as the neighboring stars, so this feature vector is unique. First of all, I build the feature table of the navigation star table: take a navigation star as the main star, select the first three neighboring stars with the smallest angular distance from the main star to form the feature vector of the main star. In this way, we can traverse the whole navigation catalog and get a navigation catalog index feature table. For the observation chart, the formula

$$\alpha = 2arctan\frac{d}{2f}$$

Then, the length of unit pixel can be obtained. By eliminating focus f, the direction vector of observation star can be obtained from a given pixel coordinate. In the same way, each star is taken as the main star, and the first three neighboring stars with the smallest distance from the plane of the main star are selected to form the eigenvector of the main star. According to this principle, the eigenvectors of corresponding navigation star and observation star should be completely consistent.

7.3 Feature Matching

For an observation star map, only those around the stars as the main star, its eigenvector is effective. When the minimum error value is lower than a given value eps, it can be considered as a successful match.

7.4 Model Solving

When eps is set to 1e-7, it can be found that the eigenvector has better matching effect.

0.0000027443614864	2119	
0.0000020230561960	2984	
0.0000012192488449	2196	
0.0000003318445257	2826	A04
0.0000001800125252	1968	A05
0.0000000213226063	259	A06
0.0000000846877644	2817	A07
0.0000000000000820	1492	A08
0.0000000000000820	1488	A09
0.0000004516224300	4691	A10
0.0000001748373793	3571	A11
0.0000017066395595	3125	
0.0000000000001312	1688	A13
0.0000000000001312	1655	A14
0.0000007455831804	2190	A15

In the first star map, we matched:

A04->2826 A05->259 A06->2817 A07->2817 A08->1492 A09->1488 A10->4691 A11->3571 A13->1688 A14->1655 A15->2190

0.0000009056510438	245	B01
0.0000006554123987	4288	B02
0.0000017153506266	245	
0.0000008101548542	2478	B04
0.0000000000000519	491	B05
0.0000000000000349	469	B06
0.0000000000000250	482	B07
0.0000000000000250	503	B08
0.0000000000000250	478	B09
0.0000000000000250	499	B10
0.0000004433142354	4093	B11
0.0000000000000113	460	B12
0.0000000000000338	507	B13
0.0000012170264068	697	
0.0000004302326187	4796	B14
0.0000025791968011	768	

In the second star map, we matched:

B01->245 B02->4288 B04->2478 B05->491 B06->469 B07->482 B08->503 B09->478 B10->499 B11->4093 B12->460 B13->507 B14->4796

```
0.0000054943392851 2466
0.0000001128752170 4273 C02
0.0000000238643618 1825 C03
0.0000001092630261 4825 C04
0.0000001092630261 4645 C05
0.0000005013213970 3373 C06
0.0001653649456078 2466
```

In the third star map, we matched:
C02->4273 C03->1825 C04->4825 C05->4645 C06->3373

```
0.0000110578384469 768
0.0000028648799389 4341
0.0000016664169824 1724
0.0000003267864687 4083 D04
0.0000028967950349 2466
0.0000001082074200 2607 D06
0.0000000251926439 2762 D07
0.0000003008472038 2898 D08
0.0000001118688759 988   D09
0.0000001118688759 992   D10
0.0000001082074200 2637 D11
0.0000001564133025 4509 D12
```

In the fourth star map, we matched
D04->4083 D06->2607 D07->2762 D08->2898 D09->988 D10->992 D11->2637 D12->4509

7.5 Algorithm Defect

1. If the star is on the edge, the nearest star may not be in the field of view. As a result, we will get an eigenvector, and the very star matching success rate is greatly reduced. If the number of stars in the field of view is too small, it will also seriously affect the overall matching accuracy.

2. The principle of feature matching is flawed. In the celestial coordinate system, the star with the closest angular distance is not necessarily the closest in the pixel coordinate system. The theory of this method is based on the nearest star in the celestial coordinate system and the closest in the pixel coordinate system. The reason for this problem should be to consider a sphere with a radius of 1 and project it onto a plane, which is indeed the case.

Preference

[1] Xia Jianbai & Chen Yongling & Wang Zhizhuo. Practical Astronomy [Z]. Wuhan University Press, 2007
[2] Copernicus. Theory of celestial motion [Z]. Business press, 1543
[3] Li Houpu & Bian Shaofeng & Zhong bin. Geographic coordinate system -- computer algebra precision analysis theory. National Defense Industry Press, 2015
[4] Wang suihui. Error theory and measurement adjustment (Second Edition). Tongji University Press, 2015
[5] Zhang Guangjun. Star map recognition [Z]. National Defense Industry Press, 2011
[6][disturbance error analysis Baidu Library] https://m.baidu.com/sf_edu_wenku/view/cb10fdd2240c844769eaee72?fromShare=1&fr=copy&copyfr=copylinkpop
[7] fast star map recognition based on multi feature matching http://www.wanfangdata.com.cn/details/detail.do?_type=perio&id=gxjmgc201908023

Appendix A

Problem 2: angle randomized decision algorithm (written in C + +)

```
#include<bits/stdc++.h>
using namespace std;
double a[5][4],A[4];
const long long test=100000000000;
const double divs=100;
const double eps=1e-10;
const double f=0.5;
struct p
{
	double pos[4],del,t;
	void printp()
	{
		printf("%.6lf ",del);
		printf("(");
		for(int i=1;i<=3;i++)printf("%.6lf ",pos[i]);
		printf(")");
		printf("(");
		for(int i=1;i<=3;i++)printf("%.6lf ",pos[i]*t);
		printf(")");
		puts("");
	}
```

```
};
int dcmp(double x){return (fabs(x)<eps)?0:(x>0?1:-1);}
vector<p>v;
bool cmp(p a1,p a2){return a1.del<a2.del;}
void print1()
{
	for(int i=1;i<=3;i++)printf("%.6lf ",A[i]);
	puts("");
	for(int i=1;i<=3;i++)
	{
		for(int j=1;j<=3;j++)printf("%.6lf ",a[i][j]);
		puts("");
	}
	puts("ans");
	for(int j=1;j<=3;j++)printf("%.6lf ",a[4][j]);puts("");
	for(int j=1;j<=3;j++)printf("%.6lf ",a[4][j]/2);puts("");
}
int main()
{
	double mn=100;
	srand(time(NULL));
	for(int i=1;i<=4;i++)for(int j=1;j<=3;j++)a[i][j]=1.0*rand()/RAND_MAX;
	for(int i=1;i<=4;i++)
	{
		double len=sqrt(a[i][1]*a[i][1]+a[i][2]*a[i][2]+a[i][3]*a[i][3]);
		for(int j=1;j<=3;j++)a[i][j]/=len;
	}
	for(int o=1;o<=1;o++)
	{
		p tmp;tmp.pos[1]=a[4][1],tmp.pos[2]=a[4][2],tmp.pos[3]=a[4][3];
		double invcostheta[4];
		for(int i=1;i<=3;i++)

	invcostheta[i]=(tmp.pos[1]*tmp.pos[1]+tmp.pos[2]*tmp.pos[2]+tmp.pos[3]*tmp.pos[3])/(tm
p.pos[1]*a[i][1]+tmp.pos[2]*a[i][2]+tmp.pos[3]*a[i][3]);
		for(int i=1;i<=1;i++)
		{
			double t=1.0*i/2,ti[4],dis[4];
			for(int j=1;j<=3;j++)
			{
				ti[j]=invcostheta[j];
				dis[j]=sqrt((ti[j]*t*a[j][1]-t*tmp.pos[1])*(ti[j]*t*a[j][1]-t*tmp.pos[1])+
				(ti[j]*t*a[j][2]-t*tmp.pos[2])*(ti[j]*t*a[j][2]-t*tmp.pos[2])+
				(ti[j]*t*a[j][3]-t*tmp.pos[3])*(ti[j]*t*a[j][3]-t*tmp.pos[3]));
```

```
                    A[j]=dis[j];
                }
            }
        }
        print1();
        for(int o=1;o<=test;o++)
        {
            p tmp;double invcostheta[4];
            for(int i=1;i<=3;i++)tmp.pos[i]=1.0*rand()/RAND_MAX;
            double
len=sqrt(tmp.pos[1]*tmp.pos[1]+tmp.pos[2]*tmp.pos[2]+tmp.pos[3]*tmp.pos[3]);
            for(int i=1;i<=3;i++)tmp.pos[i]/=len;
            tmp.del=1000;
            for(int i=1;i<=3;i++)
                invcostheta[i]=(tmp.pos[1]*tmp.pos[1]+tmp.pos[2]*tmp.pos[2]+tmp.pos[3]*tmp.p
            os[3])/(tmp.pos[1]*a[i][1]+tmp.pos[2]*a[i][2]+tmp.pos[3]*a[i][3]);
            for(int i=1;i<=divs;i++)
            {
                double cnt=0;
                double t=1.0*i/divs,ti[4],dis[4];
                for(int j=1;j<=3;j++)
                {
                    ti[j]=invcostheta[j];
                    dis[j]=sqrt((ti[j]*t*a[j][1]-t*tmp.pos[1])*(ti[j]*t*a[j][1]-t*tmp.pos[1])+
                    (ti[j]*t*a[j][2]-t*tmp.pos[2])*(ti[j]*t*a[j][2]-t*tmp.pos[2])+
                    (ti[j]*t*a[j][3]-t*tmp.pos[3])*(ti[j]*t*a[j][3]-t*tmp.pos[3]));
                    cnt+=fabs(dis[j]-A[j]);
                }
                if(dcmp(tmp.del-cnt)==1)tmp.t=t,tmp.del=cnt;
            }
            if(dcmp(mn-tmp.del)==1)mn=tmp.del,tmp.printp();
        }
}
```

Appendix B

Problem 5: database feature vector extraction algorithm (written in C + +)

```
#include<bits/stdc++.h>
using namespace std;
const int maxn=5e3+10;
```

```
const double pi=acos(-1.0);
struct star
{
	double a,b,dis,x,y,z;
	int id;
}s[maxn];
double v[maxn][7];
int n,m;
bool cmp(star a,star b)
{
	return a.dis<b.dis;
}
int main()
{
	freopen("star.txt","r",stdin);
	freopen("char.txt","w",stdout);
	scanf("%d%d",&n,&m);
	printf("%d\n",n);
	for(int i=1;i<=n;i++)
	{
		double tmp1,tmp2,tmp3,tmp4;
		cin>>tmp1>>tmp2>>tmp3>>tmp4;
		tmp2*=1000.0,tmp3*=1000.0;
		tmp2*=pi/180.0;tmp3*=pi/180;
		s[i].a=tmp2,s[i].b=tmp3,s[i].id=i;
		s[i].x=cos(s[i].b)*cos(s[i].a),s[i].y=cos(s[i].b)*sin(s[i].a),s[i].z=sin(s[i].b);
	}
	for(int o=1;o<=n;o++)
	{
		int ID;
		for(int i=1;i<=n;i++)if(s[i].id==o)ID=i;
		for(int i=1;i<=n;i++)s[i].dis=sqrt((s[i].x-s[ID].x)*(s[i].x-s[ID].x)+(s[i].y-
s[ID].y)*(s[i].y-s[ID].y)+(s[i].z-s[ID].z)*(s[i].z-s[ID].z));
		sort(s+1,s+n+1,cmp);//23 24 25 34 35 45;
		int cnt=0;
//		printf("%d %d %d %d\n",s[1].id,s[2].id,s[3].id,s[4].id);
		for(int i=1;i<=3;i++)
		{
			for(int j=i+1;j<=4;j++)
			{
				v[o][++cnt]=s[i].x*s[j].x+s[i].y*s[j].y+s[i].z*s[j].z;
				printf("%.16lf ",v[o][cnt]);
			}
		}
```

```
        puts("");
    }
}
```

Appendix C

Problem 5: extraction algorithm of star table eigenvector(written in C + +)

```
#include<bits/stdc++.h>
using namespace std;
const int maxn=1e2;
const double pi=acos(-1.0);
const double pix=tan(pi/30.0)/256.0;
struct p
{
    double x,y,z,dis,a,b;
    int id;
}s[maxn];
bool cmp(p a1,p a2)
{
    return a1.dis<a2.dis;
}
int n;
double v[maxn][7];
int main()
{
    freopen("starmap01.txt","r",stdin);
    freopen("01.txt","w",stdout);
    scanf("%d",&n);
    printf("%d\n",n);
    for(int i=1;i<=n;i++)
    {
        cin>>s[i].x>>s[i].y;
        s[i].id=i,s[i].x-=256,s[i].y-=256;
        s[i].x*=pix;s[i].y*=pix;s[i].z=1.0;
        double len=sqrt(1.0+s[i].x*s[i].x+s[i].y*s[i].y);
        s[i].x/=len,s[i].y/=len,s[i].z/=len;
    }
    for(int o=1;o<=n;o++)
    {
        int ID;
```

```
        for(int i=1;i<=n;i++)if(s[i].id==o)ID=i;
        for(int i=1;i<=n;i++)s[i].dis=sqrt((s[i].x-s[ID].x)*(s[i].x-s[ID].x)+(s[i].y-
s[ID].y)*(s[i].y-s[ID].y)+(s[i].z-s[ID].z)*(s[i].z-s[ID].z));
        sort(s+1,s+n+1,cmp);
        int cnt=0;
        for(int i=1;i<=3;i++)
        {
            for(int j=i+1;j<=4;j++)
            {
                v[o][++cnt]=s[i].x*s[j].x+s[i].y*s[j].y+s[i].z*s[j].z;
                printf("%.16lf ",v[o][cnt]);
            }
        }
        puts("");
    }

}
```

Appendix D

Problem 5: eigenvector matching algorithm (written in C + +)

```
#include<bits/stdc++.h>
using namespace std;
const int maxn=5000;
double t0[maxn][7];
double t1[100][7];
const double eps=1e-10;
int dcmp(double x){return fabs(x)<eps?0:(x>0?1:-1);}
int main()
{
    freopen("char01.txt","w",stdout);
    int n1,n2;
    scanf("%d",&n1);
    for(int i=1;i<=n1;i++)
        for(int j=1;j<=6;j++)
            scanf("%lf",&t1[i][j]);
    freopen("char.txt","r",stdin);
    scanf("%d",&n2);
    for(int i=1;i<=n2;i++)
    {
        for(int j=1;j<=6;j++)
            scanf("%lf",&t0[i][j]);
```

```
    }
    for(int i=1;i<=n1;i++)
    {
        int id=0;
        double mndel=1.0;
        for(int j=1;j<=n2;j++)
        {
            double del=0;
            for(int p=1;p<=6;p++)
                del+=(t1[i][p]-t0[j][p])*(t1[i][p]-t0[j][p]);
            if(mndel>del)
            {
                mndel=del;
                id=j;
            }
        }
        printf("%.16lf %d\n",mndel,id);
    }
}
```