Research in Astron. Astrophys. **2009** Vol. **9** No. **7**, 803 – 811
http://www.raa-journal.org *http://www.iop.org/journals/raa*

Research in Astronomy and Astrophysics

A convergent mean shift algorithm to select targets for LAMOST *

Guang-Wei Li and Gang Zhao

National Astronomical Observatories, Chinese Academy of Sciences, Beijing 100012, China;
gzhao@bao.ac.cn

Received 2008 October 28; accepted 2009 January 2

Abstract This paper firstly finds that the Mean Shift Algorithm used by the Observation Control System (OCS) Research Group of the University of Science and Technology of China in Survey Strategy System 2.10 (SSS2.10) to select targets for the Large Sky Area Multi-Object Fiber Spectroscopic Telescope (LAMOST) is not convergent in theory. By carefully studying the mathematical formulation of the Mean Shift Algorithm, we find that it tries to find a point where some objective function achieves its maximum value; the Mean Shift Vector can be regarded as the ascension direction for the objective function. If we regard the objective function as the numerical description for the imaging quality of all targets covered by the focal panel, then the Mean Shift Algorithm can find the place where the imaging quality is the best. So, the problem of selecting targets is equal to the problem of finding the place where the imaging quality is the best. In addition, we also give some effective heuristics to improve computational speed and propose an effective method to assign point sources to the respective fibers. As a result, our program runs fast, and it costs only several seconds to generate an observation.

Key words: methods: data analysis — methods: statistical

1 INTRODUCTION

The Large Sky Area Multi-Object Fiber Spectroscopic Telescope (LAMOST) (Su & Cui 2003; LAMOST Project Office 2005) is a reflecting Schmidt telescope with a 4 m aperture and a 5° field of view (which we will call "tile" hereafter). It has as many as 4000 optical fibers which can collect up to 4000 spectra during one exposure. The most important features of LAMOST are its large aperture, large field of view and many fibers. By comparison, SDSS only has a 2.5 m aperture, a 3° field of view and can only observe 640 targets simultaneously. The ability of AAT+2dF with a 4 m aperture is even less: its field of view is 2°, and it can only get 400 spectra during one exposure.

The problem of the optimal sky survey strategy is an NP-hard (nondeterministic polynomial-time hard) problem; that is, it is impossible for us to find the minimum-size set of tiles that can cover a large area and a large number of targets in a suitable time. SDSS tried to use the Network Flow Method to find a near optimal solution for this problem, but we cannot directly introduce the Network Flow Method into the LAMOST sky survey strategy. The reason for this is that the fibers on the focal panel of the LAMOST are distributed almost uniformly, and the movements of each fiber are limited to a small circle with a 2.6 m arc, unlike the fibers on the focal panel of the SDSS which can be put anywhere according to the target positions.

In order to effectively position the tiles for LAMOST, the Observation Control System (OCS) has given a series of methods, including the Static Method (SSS-0 2002), the Dynamic Method (SSS-1

* Supported by the National Natural Science Foundation of China.

2004) and the Mean Shift Method (SSS-2.10 2006). The Static Method and the Dynamic Method are both very primary, so we only discuss the Mean Shift Method in the following.

The Mean Shift Method is a greedy method, and OCS wants to use it to find the densest local sky area to observe during each observation. The Mean Shift Algorithm (Fukunaga & Hostetler 1975; Cheng 1995) is convergent, but the Mean Shift Method, which OCS gave to generate the sky survey strategy, just uses the Mean Shift Algorithm as an iterative step; as a result, the method that OCS gave is not necessarily convergent and the area it finds is not necessarily locally the densest.

It is well known that the different parts of the focal panel of the LAMOST have different imaging qualities. The farther a target is from the center of the focal panel, the worse imaging quality it has (Wang 1996; Xu 1997; Su et al. 1998; Cui et al. 2000). So, we will give a function to describe the imaging quality of the focal panel of the LAMOST.

In this paper, we will give a convergent Mean Shift Method and our sky survey strategy, which means that during each observation, we try to find the area where the focal panel of the LAMOST has the best imaging quality.

In Section 2, we will introduce the Mean Shift Algorithm. Then in Section 3, we will try to explain the physical meaning behind it and give the convergent algorithm. The program will be given in Section 4. Furthermore, we will give the optimization of computation in Section 5. In Section 6, we will give some examples. In the last section, we will present our remarks.

2 INTRODUCTION OF THE MEAN SHIFT ALGORITHM

2.1 Basic Mean Shift

Given n points $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, ..., $\boldsymbol{x}_n$ in N dimensional Euclidean Space, we can define the *basic mean shift vector at point* y as

$$M(\boldsymbol{y}) = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i - \boldsymbol{y}). \tag{1}$$

From the above, we can see that $(\boldsymbol{x}_i - \boldsymbol{y})$ is the offset vector of $\boldsymbol{x}_i$ from $\boldsymbol{y}$, and $M(\boldsymbol{y})$ is the mean offset vector of $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n$ from $\boldsymbol{y}$. We can also write $M(\boldsymbol{y})$ as

$$M(\boldsymbol{y}) = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i - \boldsymbol{y}. \tag{2}$$

If we assume that the mass of each point at $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n$ is 1, the mass center must be $\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i$. This means that $M(\boldsymbol{y})$ is the offset vector of the mass center of this mass system from $\boldsymbol{y}$.

2.2 Extensive Mean Shift

From Equation (1), we can see that the farther the $\boldsymbol{x}_i$ is from $\boldsymbol{y}$, the more strongly $\boldsymbol{x}_i$ influences $M(\boldsymbol{y})$. We can introduce a function $G(||\boldsymbol{x}_i - \boldsymbol{y}||)$ to denote how strongly the distance influences $M(\boldsymbol{y})$. Besides this, maybe a different $\boldsymbol{x}_i$ has a different weight, so we can use ω_i for $\boldsymbol{x}_i$ to show how important each $\boldsymbol{x}_i$ is. Therefore, we can rewrite $M(\boldsymbol{y})$ using a more complex form as

$$M(\boldsymbol{y}) = \frac{\sum_{i=1}^{n} G(||\boldsymbol{x}_i - \boldsymbol{y}||)\omega_i(\boldsymbol{x}_i - \boldsymbol{y})}{\sum_{i=1}^{n} G(||\boldsymbol{x}_i - \boldsymbol{y}||)\omega_i}. \tag{3}$$

Moreover, we can change this form to

$$M(\boldsymbol{y}) = \frac{\sum_{i=1}^{n} G(||\boldsymbol{x}_i - \boldsymbol{y}||)\omega_i\boldsymbol{x}_i}{\sum_{i=1}^{n} G(||\boldsymbol{x}_i - \boldsymbol{y}||)\omega_i} - \boldsymbol{y}. \tag{4}$$

If we let

$$m(\boldsymbol{y}) = \frac{\sum_{i=1}^{n} G(||\boldsymbol{x}_i - \boldsymbol{y}||)\omega_i\boldsymbol{x}_i}{\sum_{i=1}^{n} G(||\boldsymbol{x}_i - \boldsymbol{y}||)\omega_i}, \tag{5}$$

the $M(\boldsymbol{y})$ will have a simple form:

$$M(\boldsymbol{y}) = m(\boldsymbol{y}) - \boldsymbol{y}. \tag{6}$$

2.3 Mean Shift Algorithm

Now, let us talk about the *Mean Shift Algorithm* which makes $M(\boldsymbol{y})$ smaller and smaller until $M(\boldsymbol{y})$ is less than a given small number. This means that the purpose of the iteration is to find a point which is some "mass center" of targets that the tile first covered. The iterative method can be written as:

$$\boldsymbol{y}_{i+1} = m(\boldsymbol{y}_i). \tag{7}$$

There is a theorem (SSS2.10 2006) which makes this iteration convergent if the function $G(\cdot)$ is a convex function, which means that the larger the absolute value of r is, the smaller $G(r)$ is.

Let us see a very special example: $G(||\boldsymbol{x}_i - \boldsymbol{y}||) \equiv 1$ and $\omega_i \equiv 1$. Equation (4) becomes Equation (2). Because there is no variable $\boldsymbol{y}$ in function $G(||\boldsymbol{x}_i - \boldsymbol{y}||)$, $m(\boldsymbol{y})$ must be a constant. As a result, the iteration only needs one step, that is, $\boldsymbol{y}_1 = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i$.

2.4 Select Targets for LAMOST by SSS2.10's Method

SSS2.10 uses the Mean Shift Algorithm to find the densest region of targets, and the method used is as follows:

step 0: give a precision ε for iteration;

step 1: put the tile's center on some point $\boldsymbol{y}_0$ in the sky area that is supposed to be observed;

step 2: find positions of all targets that the tile can cover in this area. We can denote these positions as $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n$;

step 3: calculate the new tile position $\boldsymbol{y}_1$ by the formulation $\boldsymbol{y}_1 = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i$;

step 4: if $|\boldsymbol{y}_1 - \boldsymbol{y}_0| < \varepsilon$, stop, then the $\boldsymbol{y}_1$ is the desired position; else $\boldsymbol{y}_0 := \boldsymbol{y}_1$, go to step 1.

Now, we can give an illustration for the Mean Shift Method given by SSS2.10 according to Figure 1. Tile1 covers the "open circles" and "solid circles". Tile2 covers the "solid circles" and "open triangles". $\boldsymbol{y}_0$ is the initial iteration point and also the center of Tile1. $\boldsymbol{y}_1$ is calculated from the "open circles" and "solid circles" by Equation (7) and also the center of Tile2. $\boldsymbol{y}_2$ is calculated from the "solid circles" and "open triangles" by Equation (7). We can see that $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ are calculated from different target groups, though they have a common part of "solid circles" because the "open circles" have no relation with "open triangles". Thus, the stop condition for iteration $|\boldsymbol{y}_{i+1} - \boldsymbol{y}_i| < \varepsilon$ has no meaning. Moreover, this iteration is not necessarily convergent.

The most important point here is that the Mean Shift Algorithm is convergent, but SSS2.10 only uses the Mean Shift Algorithm as an iterative step, so its iteration is not necessarily convergent.

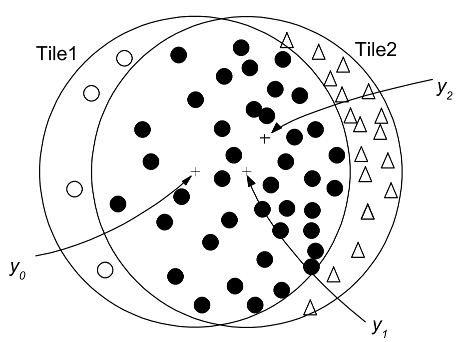

Fig. 1 Illustration for the Mean Shift Method given by SSS2.10.

3 THE CONVERGENT ALGORITHM

3.1 The Imaging Quality Function with Edge Effect

Now, we should consider the meaning behind the Mean Shift Algorithm and try to find a better method.

Let us see the formulation $M(\boldsymbol{y}) = \frac{\sum_{i=1}^{n} G(||\boldsymbol{x}_i - \boldsymbol{y}||)\omega_i \boldsymbol{x}_i}{\sum_{i=1}^{n} G(||\boldsymbol{x}_i - \boldsymbol{y}||)\omega_i} - \boldsymbol{y}$ again. The Mean Shift Algorithm tries to iteratively find a point $\boldsymbol{y}$ that makes $M(\boldsymbol{y}) = 0$. That is, to let

$$\boldsymbol{y} = \frac{\sum_{i=1}^{n} G(||\boldsymbol{x}_i - \boldsymbol{y}||)\omega_i \boldsymbol{x}_i}{\sum_{i=1}^{n} G(||\boldsymbol{x}_i - \boldsymbol{y}||)\omega_i}.$$

Again,

$$\sum_{i=1}^{n} G(||\boldsymbol{x}_i - \boldsymbol{y}||)\omega_i(\boldsymbol{x}_i - \boldsymbol{y}) = 0.$$

Now, we should wonder if this equation is the signal that some function $F(\boldsymbol{y})$ achieves its maximum value at the point $\boldsymbol{y}$. In other words, is there such a function $F(\boldsymbol{y})$ that

$$\nabla F(\boldsymbol{y}) = \sum_{i=1}^{n} G(||\boldsymbol{x}_i - \boldsymbol{y}||)\omega_i(\boldsymbol{x}_i - \boldsymbol{y}). \tag{8}$$

If so,

$$F(\boldsymbol{y}_0) = \max F(\boldsymbol{y}) \Longrightarrow \sum_{i=1}^{n} G(||\boldsymbol{x}_i - \boldsymbol{y}_0||)\omega_i(\boldsymbol{x}_i - \boldsymbol{y}_0) = 0.$$

Let

$$F(\boldsymbol{y}) = \sum_{i=1}^{n} f(||\boldsymbol{x}_i - \boldsymbol{y}||)\omega_i, \tag{9}$$

where the function $f(\cdot)$ is derivable. Then,

$$\nabla F(\boldsymbol{y}) = \sum_{i=1}^{n} -\frac{f'(||\boldsymbol{x}_i - \boldsymbol{y}||)}{||\boldsymbol{x}_i - \boldsymbol{y}||}\omega_i(\boldsymbol{x}_i - \boldsymbol{y}).$$

If we let $-\frac{f'(r)}{r} = G(r)$, then $\sum_{i=1}^{n} G(||\boldsymbol{x}_i - \boldsymbol{y}||)\omega_i(\boldsymbol{x}_i - \boldsymbol{y}) = 0$ is the necessary condition for $F(\boldsymbol{x})$ to achieve its maximum value at point $\boldsymbol{y}$. Thus, we know the intention of the Mean Shift Algorithm is to find a point where some function achieves its maximum value.

Now, we would like to see how the Mean Shift Algorithm can be applied to LAMOST in the following. In the function $F(\boldsymbol{y}) = \sum_{i=1}^{n} f(||\boldsymbol{x}_i - \boldsymbol{y}||)\omega_i$, we can let ω_i denote the weight of the target that rests at the position $\boldsymbol{x}_i$ because different targets in the input catalog may have different weights; we can also let $f(||\boldsymbol{x}_i - \boldsymbol{y}||)$ denote the edge effect, which means that if function $f(\cdot)$ is convex, the farther the $\boldsymbol{x}_i$ is from focal panel center $\boldsymbol{y}$, the worse imaging quality it has in the tile, so we will call it the imaging quality function later. Then, the function $F(\boldsymbol{y})$ is the whole imaging quality of targets that are covered by the tile. Thus, if we make function $F(\boldsymbol{y})$ achieve its maximum value at a point, the tile will have its best imaging quality at this point.

3.2 The Imaging Quality Function without Edge Effect

When the LAMOST's edge effect is very tiny, which means that the target beside the focal panel edge has the same imaging quality as the target in the center, then we can set the function $f(\cdot) \equiv 1$ and our objective function must change to

$$F(\boldsymbol{y}) = \sum_{i=1}^{n} \omega_i. \tag{10}$$

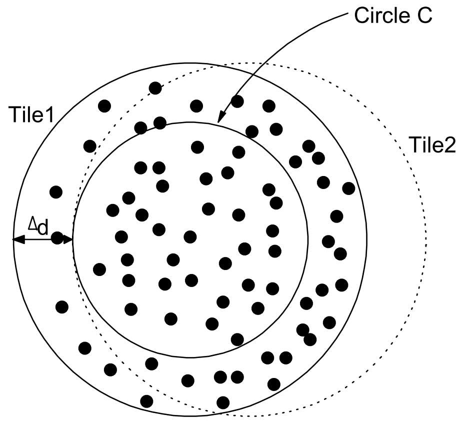

Fig. 2 For the case that the edge effect is very tiny, the mean shift vector is determined by the positions of the targets that are between Circle C and Tile1.

Now, what should the mean shift vector be? Let us consider Figure 2. Tile1 is the initial position. After the shift, the tile moves to the position of Tile2. If the maximum distance that the tile can move is Δd, then Tile1 and Tile2 have a common part, that is Circle C. Thus, the targets that really influence the mean shift vector are the targets that are between Circle C and Tile1. So, if we want to calculate the mean shift vector of a tile, we just need to use the targets that are in a small zone between Circle C and Tile1.

3.3 The Method of Selecting Targets for LAMOST

Now, we can give our Selecting Targets Algorithm for LAMOST:

step 0: give a precision ε, an initial position $\boldsymbol{y}_0$ and a tentative step l for iteration;

step 1: give a suitable function $f(\cdot)$ to describe the LAMOST focal panel imaging quality, and if the edge effect is very tiny, we let $f(\cdot) \equiv 1$;

step 2: from all targets covered by the tile with $\boldsymbol{y}_0$ as its center, calculate $F(\boldsymbol{y}_0)$ and $M(\boldsymbol{y}_0)$. Subsequently, calculate the unit vector $u(\boldsymbol{y}_0)$ of $M(\boldsymbol{y}_0)$;

step 3: $s := l$;

step 4: if $s < \varepsilon$, stop; else $s := 0.618s$, $\boldsymbol{y}_1 := \boldsymbol{y}_0 + u(\boldsymbol{y}_0) \cdot s$, and calculate $F(\boldsymbol{y}_1)$, $M(\boldsymbol{y}_1)$ and the unit vector $u(\boldsymbol{y}_1)$ of $M(\boldsymbol{y}_1)$;

step 5: if $F(\boldsymbol{y}_1) > F(\boldsymbol{y}_0)$, then $\boldsymbol{y}_0 := \boldsymbol{y}_1$, go to step 3; else go to step 4.

The details of step 2 are as follows:

For each target covered by the tile with $\boldsymbol{y}_0$ as its center, calculate the target local coordinate $\boldsymbol{y}'_i$ with respect to $\boldsymbol{y}_0$ (that is, if the covering point is located at (0,0) and the longitude it is on becomes 0 longitude, then the spherical coordinate of the target is $\boldsymbol{y}'_i$). Subsequently, calculate $f(||\boldsymbol{y}'_i||)\omega_i$ and $-\frac{f'(||\boldsymbol{y}'_i||)}{||\boldsymbol{y}'_i||}\boldsymbol{y}'_i\omega_i$ for the case with edge effect, and then add them to $F(\boldsymbol{y}_0)$ and mean shift vector $M(\boldsymbol{y}_0)$ respectively. For the case without edge effect, add ω_i to $F(\boldsymbol{y}_0)$ and if $2.5 - \Delta d < ||\boldsymbol{y}'_i|| < 2.5$, then add $\boldsymbol{y}'_i\omega_i$ to $M(\boldsymbol{y}_0)$.

3.4 The Proof of Convergence

This algorithm is convergent. Firstly, the number of targets that the LAMOST focal panel covers has an upper boundary which is denoted as M, because this number cannot surpass the number of targets that the input catalog has. Secondly, the function $f(\cdot)$ has an upper bound which is denoted as U, because $f(\cdot)$ is convex and derivable, and then it can reach its maximum value at the tile center. Lastly, it is impossible for us to give target weight ω_i an infinite value, so target weights ω_is also have an upper boundary which is denoted as W. Thus, the function $F(\boldsymbol{y}) = \sum_{i=1}^{n} f(||\boldsymbol{x}_i - \boldsymbol{y}||)\omega_i \leq MUW$. Because the goal of the algorithm is to find the maximum value, and every step finds a bigger value than the previous one, it must be convergent.

4 THE PROGRAM

We will give the implementation details of our program as follows.

4.1 Selection of Initial Iteration Points

The initial iteration points we used are from Hardin, Sloane, and Smith's catalog of covering points on the sphere (Hardin et al. 2001). These covering points are almost uniformly distributed on the sphere. In fact, it is their uniform distribution that makes us choose them as the initial iteration points. Because we only use a partial region, we select the covering points that are in the desired region.

4.2 Initialization of the Covering Points

To initialize the covering points, we should calculate the $F(\cdot)$ values and their mean shift vectors at the covering points. We could calculate these values one point at a time, but this method is very time consuming, because we must read the hard disk as many times as there are covering points. In order to save computation time, we use the following method, called the Initialization Algorithm.

step 0: initialize the values of $F(\cdot)$ and mean shift vector $M(\cdot)$ of each covering point with 0;

step 1: read a target with position $\boldsymbol{x}_i$ from the input catalog and find all covering points that can cover it;

step 2: for each covering point $\boldsymbol{y}$ that can cover this target, calculate the target's local coordinate $\boldsymbol{y}'_i$ with respect to $\boldsymbol{y}$. Subsequently, for the case with edge effect, calculate $f(||\boldsymbol{y}'_i||)\omega_i$ and $-\frac{f'(||\boldsymbol{y}'_i||)}{||\boldsymbol{y}'_i||}\boldsymbol{y}'_i\omega_i$, and then add them to $F(\boldsymbol{y})$ and mean shift vector $M(\boldsymbol{y})$ respectively; for the case without edge effect, add ω_i to $F(\boldsymbol{y})$ and if $2.5 - \Delta d < ||\boldsymbol{y}'_i|| < 2.5$, then add $\boldsymbol{y}'_i\omega_i$ to $M(\boldsymbol{y})$;

step 3: if there are no targets left, go to step 4; else go to step 1;

step 4: calculate the unit vector of the mean shift vector of each covering point.

From the above, we can see that to initialize all covering points, we only need to read each target once, so it saves a lot of time.

4.3 The Structure of the Program

The program is a greedy algorithm. Every time, it finds a tile position where $F(\cdot)$ achieves its maximum value. The details of the program are as follows:

step 0: give a tentative step l and a precision ε;

step 1: initialize all the covering points by the Initialization Algorithm;

step 2: find the covering points with the maximum value of $F(\cdot)$;

step 3: use the Selecting Targets Algorithm to find the new observation position where the tile has the best imaging quality for the case with edge effect or where the tile can cover the maximum weight of targets for the case without edge effect;

step 4: assign targets to tile fibers;

step 5: if there are no more targets that can be assigned, stop;

step 6: delete targets that have been assigned from the input catalog, and update values of $F(\cdot)$ and $M(\cdot)$ of all covering points that can cover these targets, go to step 2.

5 THE OPTIMIZATION OF COMPUTATION

The bottleneck of this program is having to read the hard disk too many times. To reduce reading times, we use some tricks.

We study this algorithm carefully and find that there are three places that cost too much time. The first place is for a given position to calculate its $F(\cdot)$ and $M(\cdot)$ value. It spends much time in finding the targets that the tile can cover. If we do not carefully manage the process, it may check many targets that the tile never covers. To improve the search speed, we use many tricks: split the input catalog by 0.2° declination into many files. In each file, we arrange targets by right ascension. Then index each file by 0.2° right ascension. Thus, we divide the region into many $0.2^\circ \times 0.2^\circ$ blocks. As a result, we can use spherical formulations to find out which block can be covered by the title, so the algorithm does not waste much time checking the targets that are never covered by the tile.

The second place is that when we get a tile that has the maximum $F(\cdot)$ value, how do we assign covered targets to fibers?

Before we discuss our strategy, we should learn about how the fiber positioning units are arranged. There are seven fiber positioning units in Figure 3. Each circle represents one fiber's movement range. Each range has an overlapped area with others, but each point in one circle can be covered by at most three circles at the same time, that is, each target can be reached by at most three fibers simultaneously (Xing et al. 1998; Hu et al. 2000; Peng et al. 2003; Xing et al. 2007).

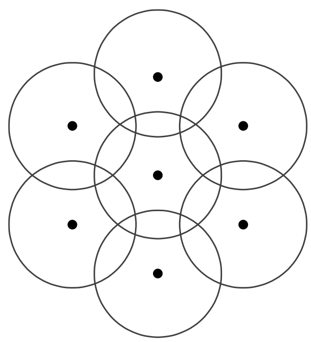

Fig. 3 Fiber positioning units on the LAMOST focal panel.

Now we switch to our strategy. Read a target from the input catalog and find the fibers that can reach it. There are at most 3 fibers that can reach it, so this step will not cost much time. (2) Decide which fiber observes the target. We use the following principles to do this. Firstly, assign the target to the fiber that is free. Secondly, if all fibers that can cover this target are not free, but there are some targets that have less priority than the read target, then replace the lowest priority target by the read target. Thirdly, if the read target is in the region that only one fiber can reach, and there is still another target on the fiber which is farther than the read target, then replace the farther target by the read target. Fourthly, assign the target preferentially to the rightmost fiber (Peng et al. 2003). That is, we wish to have as many fiber arms as possible stretch to the same left orientation. Thus, we will more effectively reduce fiber collisions. Lastly, if the four conditions above appear simultaneously, the priority follows their above order.

To collect as many targets as possible, the tile can use this method many times until no targets can be collected. We believe that two times are sufficient.

The third place is after we have collected the targets for the tile, how do we could delete these targets from the catalog? Maybe we delete them directly from the input catalog, but this costs too much time in reading and writing to the hard disk. However, we can use a heuristic to avoid this. Firstly, we give each target an ID number, and then allocate a space in the memory. For each target ID, there is only a bit in the memory space to mark it. When this target is read, the corresponding bit will be set to '1', otherwise, it is ‘0’. If there are 100 million targets in the input catalog, it only uses 12 Mbytes of memory. Thus, we will save much time without using much memory.

6 THE EXAMPLES

We use USNOA2.0 (Monet et al. 1998) as our input catalog. For this input catalog, we do some treatments because the diameter of a fiber is 3.3 arcsec and if two stars have a distance less than 3.3 arcsec and the difference of their magnitudes is less than 5 mag, they will contaminate each other. We drop the star that has a distance less than 3.3 arcsec from its neighbor and an Rmag 5 mag larger than its neighbor. If there are two or more stars within 3.3 arcsec with Rmag differences less than 5 mag, then we drop all these stars. From this pretreated catalog, we select stars between 15 Rmag and 17 Rmag as our final input catalog. From this input catalog, we use area RA: [75,115] $\times$ DEC: [0, 40] as our testing area. There are 5 832 168 stars in this area. The precision for iteration that we use is $0.01\,^\circ$; the tentative step is 0.4°; we set all $\omega_i \equiv 1$.

6.1 Selecting Targets Algorithm with Edge Effect

The imaging quality function we use is

$$f(r) = \begin{cases} 1 - (\mathrm{r}/2.5)^3, & r < 2.5, \\ 0, & r \geq 2.5. \end{cases} \tag{11}$$

We do four trials. The first uses about 85% of the total stars; the second about 90%, the third about 95%, and the last about 99%. All these proportions (Observed Proportion) are listed in the first column in Table 1. In addition, we also give the number of tiles used for observation (Tile Number), the total number of stars collected (Number of Collected Stars), and also the fiber utilization ratio (Fiber Utilization Ratio), which is the value of $\frac{\text{Number of collected stars}}{3950\times\text{Tile Number}}$.

Table 1 Results of Four Trials for the Selecting Targets Algorithm with Edge Effect

Observed Proportion (%)	Tile Number	Number of Collected Stars	Fiber Utilization Ratio (%)
85.03	1 379	4 959 158	91.04
90.00	1 518	5 249 154	87.54
95.02	1 710	5 541 588	82.04
99.00	2 048	5 774 124	71.38

6.2 Selecting Targets Algorithm without Edge Effect

Again, we use the Selecting Targets Algorithm Without Edge Effect to redo this example. The Δd we use is 0.25° which is somewhat arbitrary. The new results are listed in Table 2. Each column of Table 2 has the same meaning as Table 1.

Table 2 Results of Four Trials for Selecting Targets Algorithm without Edge Effect

Observed Proportion (%)	Tile Number	Number of Collected Stars	Fiber Utilization Ratio (%)
85.01	1 318	4 957 705	95.23
90.00	1 436	5 249 230	92.54
95.01	1 601	5 541 282	87.62
99.00	1 877	5 773 898	77.88

6.3 Comparison of the Two Examples

From the two tables, we can see that in order to observe the same number of stars, the Selecting Targets Algorithm With Edge Effect needs more tiles, so its fiber utilization ratio is lower. This is the certain cost for considering imaging quality, which is consistent with common sense.

7 DISCUSSION AND REMARKS

Our algorithm is a convergent algorithm, which is a very different and important feature of SSS2.10. We also give this algorithm a meaningful physical explanation. Each item in our objective function has a very important physical meaning. As far as we know, this is the first time that someone has tried to describe LAMOST focal panel's character by a mathematical formulation so thoroughly.

In addition, to improve the computational speed, we use many heuristics, which is a big improvement. The computer we used is a DELL microcomputer with an Intel Pentium(R) 4 2.66 GHz processor and 512 Mbytes of memory. It took the computer several seconds to generate a tile.

From our statistical results, we find that about 75% of the fiber arms stretch to the left and more than 90% of the stars are in the area that only one fiber can reach, which confirms that our strategy of assigning stars to fibers is simple but useful.

Besides these merits, there is still a great deal of further work we should do. Firstly, what criteria should we use to set the imaging quality function? Perhaps this depends on experience. Secondly, though our strategy of assigning stars to fibers is very effective, it cannot avoid fiber collisions completely. Further work may need more Computer Graphic knowledge. Lastly, though the performance of our program is very good, it is just a greedy algorithm, which means that it is not optimal, and there is still room for improvement. We will track this point in our future work.

Acknowledgements The authors thank Dr. Haotong Zhang for his helpful suggestions. This research is supported by the National Natural Science Foundation of China under grants 10433010 and 10521001, and the National Basic Research Program of China (973 Program) under grant 2007CB815103.

References

Cheng, Y. Z. 1995, IEEE Trans. on Pattern Analysis and Machine Intelligence, 17(8), 790
Cui, X. Q., et al. 2000, SPIE, 4003, 347
Fukunaga, K., & Hostetler, L. 1975, IEEE Trans. on Information Theory, 21(1), 32
Hardin, R. H., Sloane, N. J. A., & Smith, W. D. 2001, *http://www.research.att.com/∼njas/icosahedral.codes/*
Hu, H. Z., Xing, X. Z., Li, W. M., et al. 2000, Journal of University of Science and Technology of China, 30(6), 707
LAMOST Project Office, 2005, The Large Sky Area Multi-Object Fiber Spectroscopic Telescop (LAMOST)
Monet, D., Bird, A., Canzian, B., et al. 1998, *http://vizier.u-strasbg.fr/vizier/VizieR/pmm/usno2.htx*
OCS Research Grop, 2002, SSS-0 System Design Report
OCS Research Grop, 2004, SSS-1 System Design Report
OCS Research Grop, 2006, SSS-2.10 System Design Report
Peng, X. B., Zhai, C. Xing, X. Z., Hu, H. Z., & Li, W. M. 2003, Journal of University of Science and Technology of China, 33(1), 78
Su, H. J., & Cui, X. Q. 2003, SPIE, 4837, 26
Su, D. Q., Cui, X. Q., Wang, Y. N., & Yao, Z. Q. 1998, SPIE, 3352, 378
Wang, Y. N. 1996, LAMOST Feasibility Study Report, 56
Xing, X. Z., Hu, H. Z., Du, H. S., & Zhai, C. 1998, SPIE, 3352, 839
Xing, X. Z., Hu, H. Z., & Chu, J. R. 2007, Journal of University of Science and Technology of China, 37(7), 596
Xu, W. L. 1997, Acta Astrophysica Sinica, 17, 96