RAA 2018 Vol. **18** No. **2**, 22(10pp) doi: 10.1088/1674–4527/18/2/22

http://www.raa-journal.org *http://iopscience.iop.org/raa*

Research in Astronomy and Astrophysics

Matching CCD images to a stellar catalog using locality-sensitive hashing

Bo Liu[1], Jia-Zong Yu[1] and Qing-Yu Peng[1,2]

[1] College of Information Science and Technology, Jinan University, Guangzhou 510632, China; *tlbxldd@jnu.edu.cn*
[2] Sino-French Joint Laboratory for Astrometry, Dynamics and Space Science, Jinan University, Guangzhou 510632, China

Received 2017 October 10; accepted 2017 November 28

Abstract The usage of a subset of observed stars in a CCD image to find their corresponding matched stars in a stellar catalog is an important issue in astronomical research. Subgraph isomorphic-based algorithms are the most widely used methods in star catalog matching. When more subgraph features are provided, the CCD images are recognized better. However, when the navigation feature database is large, the method requires more time to match the observing model. To solve this problem, this study investigates further and improves subgraph isomorphic matching algorithms. We present an algorithm based on a locality-sensitive hashing technique, which allocates quadrilateral models in the navigation feature database into different hash buckets and reduces the search range to the bucket in which the observed quadrilateral model is located. Experimental results indicate the effectivity of our method.

Key words: astronomical databases: miscellaneous — methods: data analysis — techniques: image processing

1 INTRODUCTION

Positional observations for objects in the sky are important to astronomical research. Generally, this problem (called the star-image-catalog matching problem in the following) requires using a subset of observed stars in a CCD image to find their corresponding matched stars in a stellar catalog, and deriving the transformation between detector coordinates and celestial coordinates, which allows for the position of the observed object to be calculated. The core of the matching problem is searching for objects that correspond to each other in two coordinate systems. Two main difficulties exist in solving this problem. One is high time consumption due to a large amount of stars listed in the stellar catalog. The other is mismatching or failing to find any matched stars because of uncertainty in the measurement system.

The star-image-catalog matching problem has been studied by researchers for many years, and a few techniques have been proposed in which two kinds of algorithms are widely used. The first kind is related to geometric subgraph isomorphism, which regards stars as vertices and looks for similar lines, triangles or polygons in two catalogs by distances, ratios, area ratios and so on. Algorithms of this kind include triangle based algorithms (Groth 1986; Cole & Crassidis 2006; Tabur 2007), the pyramid star identification technique (Mortari et al. 2004) and the decreasing redundancy matching approach (Lu et al. 2015). The other kind is based on star distribution features (i.e., star patterns) around a main star. Methods of this kind search for the most similar stars in two catalogs by pattern recognition, and representative approaches include grid algorithms (Padgett & Kreutz-Delgado 1997; Na et al. 2009; Tang et al. 2016) and the identification method using star radial and cyclic statistical features (Zhang et al. 2008; Xie et al. 2012; Ji et al. 2013). Generally, the second kind of approach has a requirement of high image quality because missing stars or noise objects lead to false star patterns. Therefore, the first kind of approach is more applicable.

Several improved techniques have been proposed to solve the mismatching and time consumption problems, and quadrilateral based methods are particularly notable. Binary search is used in the quadrilateral matching process (Lin et al. 2000; Qi et al. 2014). Lang et al. (2010) presented the Astrometry.net algorithm that uses a geometric hashing technique based on four stars in a quadrilateral. Given stars A, B, C and D, stars A and B are located on the origin and (1; 1), respectively, of a local coordinate system, in which the positions of stars C and D are computed. Coordinates (x_C; y_C; x_D; y_D) become the hash code that describes the relative positions of the four stars. Then, the approach organizes the hash codes into a kd-tree (Bentley 1975) for the rapid searching of quadrilaterals whose hash codes are near any given query hash code to expedite the matching process. In this method, although the hash code of a quadrilateral is constant under translation, scaling and rotation of the four stars, the method may not be preserved under shearing in the general affine transformation between two coordinate systems. Heyl (2013) studied a fast matching algorithm to cope with shearing. The algorithm builds quadrilaterals from sets of four objects in each catalog and calculates the ratio of areas of the triangles that span three vertices of quadrilaterals. Two area ratios are used to represent a quadrilateral, and the kd-tree data structure is applied in accelerating the quadrilateral search. Both Lang et al. (2010) and Heyl (2013) provided good solutions to the star-image-catalog matching problem and utilized quadrilaterals and a kd-tree data structure consistently, but with different geometric feature values.

Several studies pointed out that quadrilateral matching algorithms achieve better results than triangle algorithms. A triangle matching algorithm may fail if positional noises exist in astronomical images. Lang et al. (2010) highlighted that quintuples of stars are even more distinctive than quads, but near-neighbor lookup, even with a kd-tree, becomes more time-consuming with increasing feature dimensionality. The experiments in Heyl (2013) proved that the quadrilateral matching technique takes a significantly longer time to execute than the triangle algorithm. Obviously, more polygons are required for searching when a large number of stars are present in an asterism (with a set of four or more stars). However, the possibility of finding the correct transformation between two coordinate systems increases. Therefore, a more efficient technique for polygon matching is necessary.

The locality-sensitive hashing (LSH) (Indyk & Motwani 1998) technique has been explored in the approximate nearest neighbor (ANN) problem, and effective results have been achieved. LSH is an indexing method for high-dimensional similarity search, which uses a family of locality-sensitive hash functions to hash nearby objects in a high-dimensional space into the same bucket. The experiments in Datar et al. (2004) showed that the LSH technique is up to 40 times faster than the kd-tree technique.

In our research, LSH is applied in the star-image-catalog matching problem, in which the asterisms in a CCD image and a stellar catalog are hashed into buckets. According to the LSH principle, the approximate asterisms (e.g., quadrilaterals) in the two coordinate systems are mapped into the same buckets with high probability, and the candidate stars can be selected efficiently in a stellar catalog for a given asterism in a CCD image.

The main contributions of this paper are as follows:

(1) To solve the efficiency problem for matching approximate asterisms in two coordinate systems, the LSH technique is first introduced into the star-image-catalog matching problem. We approach the problem as an ANN problem, which searches the quadrilaterals in the stellar catalog for a matching quadrilateral in the CCD image.

(2) An algorithm (HashQuad) is proposed based on the LSH. This algorithm applies basic LSH (Indyk & Motwani 1998; Datar et al. 2004). A quadrilateral in a CCD image is only compared with the quadrilaterals in the stellar catalog that are mapped into the same bucket, thereby reducing the search range significantly.

(3) Experimental results demonstrate the ability of the LSH technique to solve time-consuming problems for polygon matching with high accuracy.

The rest of this paper is organized as follows. Section 2 introduces the basic principles of LSH. Section 3 presents the algorithm that uses LSH in the star-image-catalog matching problem. Section 4 discusses several experiments and their results. Finally, Section 5 concludes this paper.

2 BASIC PRINCIPLES OF LSH

The concept of LSH was first introduced by Indyk and Motwani (Indyk & Motwani 1998). The principle of LSH is based on the simple idea that neighboring objects in the original data space remain close together in the new data space after projection transformation using LSH functions. Then, these objects are likely to collide in the same bucket by hash coding. In contrast, objects that are far apart are less likely to collide in the same bucket. The LSH technique is suitable for application in approximate searching, that is, the objects in a data space are grouped by distances. Hence, while retrieving the points close to an object q, only objects in the same group with q should be compared, thus saving search time.

Given that LSH provides a probabilistic guarantee that it will return the correct answer, LSH functions should satisfy several criteria that are provided in definition 1.

Table 1 Time of Search and Comparison for M67 (ms)

Image ID	Q-I	Q-II	HashQuad
1	1.76	1.46	3.82
2	1.55	1.15	1.55
3	1.51	1.06	1.04
4	63.12	21.6	106.57
5	13.51	1.23	2.95
6	3.38	1.42	1.52
7	1.82	1.36	1.4
8	41.57	4.42	21.85
9	5.11	1.33	1.02
10	5.23	1.17	1.57
11	1.52	3.73	0.96
12	1.53	4.34	0.99
13	1.34	4.38	0.9
14	1.29	4.16	0.87
15	2.3	3.94	0.86
16	4.73	3.73	0.92
17	1.32	3.68	0.9
18	1.27	3.47	0.91
19	2.85	1.76	0.92
20	57.84	6.75	14.93

Table 2 Time of Search and Comparison for NGC 6709 (ms)

Image ID	Q-I	Q-II	HashQuad
1	8.82	6.01	6.59
2	31.07	23.66	22.01
3	27.35	19.78	18.55
4	15.27	6.43	5.9
5	7.64	6.97	7.77
6	13.56	7.55	7.12
7	18.56	7.13	6.4
8	9.53	8.08	6.56
9	11.09	7.44	29.61
10	98.07	65.44	79.02
11	8.47	8.73	7.54
12	10.93	9.13	8.57
13	14.65	14.96	13.3
14	16.61	10.67	9.68
15	11.47	11.63	11.15
16	10.15	10.22	9.8
17	18.5	11.9	55.84
18	14.04	9.83	100.11
19	20.03	12.94	315.12
20	14.99	11.03	10.04

Table 3 Searching Times for M67 (times)

Image ID	Q-I	Q-II	HashQuad
1	32 270	360	217
2	32 270	194	217
3	32 270	194	217
4	18 599 860	5627	75 352
5	178 643	646	1733
6	178 643	646	1733
7	67 158	40	329
8	1 939 759	1425	12 630
9	68 471	32	1032
10	68 500	71	449
11	32 270	298	217
12	32 270	298	217
13	32 270	298	217
14	32 270	298	217
15	32 270	298	217
16	32 270	298	217
17	32 270	298	217
18	32 270	298	217
19	68 471	32	1032
20	1 939 745	1380	13 211

Table 4 Searching Times for NGC 6709 (times)

Image ID	Q-I	Q-II	HashQuad
1	14 481	316	136
2	49 385	1295	83
3	49 385	1038	83
4	14 481	315	136
5	14 479	100	188
6	14 828	100	193
7	14 479	100	185
8	14 479	100	185
9	437 147	1122	6725
10	510 361	1436	10 559
11	14 474	525	93
12	14 823	487	94
13	4836	566	143
14	4836	566	143
15	4836	579	143
16	4836	579	143
17	514 291	1600	15 213
18	15 859	703	50 098
19	135 728	623	108 355
20	14 474	407	93

Definition 1: Let S be the domain of objects and D the distance measure between two objects. A function family $H = \{h : S \rightarrow U\}$ is (d_1, d_2, p_1, p_2)-sensitive if for any $o, q \in S$:

- if $D(o, q) \leq d_1$, then $Pr_H[h(o) = h(q)] \geq p_1$,
- if $D(o, q) > d_2$, then $Pr_H[h(o) = h(q)] \leq p_2$,

where $p_1 > p_2, d_1 < d_2$.

Different LSH families that comply with definition 1 can be applied in the approximate neighbor search (Slaney & Casey 2008; Matsushita & Wada 2009). In our matching problem, we use p-stable LSH (Datar et al. 2004), which is described by definition 3, and the concept of stable distribution is given by definition 2.

Definition 2: Distribution D over the real number set is called p-stable, if $p>0$ exists such that for n

real numbers v_1,..., v_n and independent identically distributed variables X_1,..., X_n with distribution *D*, the random variable $\sum_i v_i X_i$ has the same distribution as variable $(\sum_i |v_i|^p)^{1/p} X$, where *X* is a random variable with distribution *D*.

Stable distributions exist for any $p \in (0, 2]$ (Zolotarev 1986), such as the Gaussian distribution that is 2-stable.

Definition 3: Given a *d*-dimensional vector *v*, a *p*-stable hash function is defined as

$$h_{a,b}(v) = \lfloor \frac{a \bullet v + b}{W} \rfloor \tag{1}$$

where *a* is a *d*-dimensional random vector sampled from a *p*-stable distribution. The function projects vectors onto vector *a*, and the axis is quantized with interval *W*, i.e. *W* is the width of each quantization bin. Parameter *b* is a uniformly chosen real number from the range [0,*W*].

Datar et al. (2004) proved that a function family given by Equation (1) is (d_1, d_2, p_1, p_2)-sensitive.

In practice, more than one hash table can be constructed by LSH to ensure a successful approximate search. The procedure of constructing hash tables is described as follows (Indyk & Motwani 1998; Datar et al. 2004):

(1) Supposing that a function family $H = \{h : S \to U\}$ is locality-sensitive and a compound hash function family is $G = \{g : S \to U^M\}$, where an integer $M > 0$, then for $g \in G$, $g(v) = (h_1(v), ..., h_M(v))$;
(2) Given an integer L, g_1, g_2,..., g_L are selected from G independently and uniformly at random, and $g_i (1 \leq i \leq L)$ is used to construct one hash table, resulting in L hash tables.
(3) Each object in a data set is hashed into L buckets according to g_1, g_2,..., g_L.

Based on the constructed L hash tables, a k-nearest neighbor search for object o is processed as follows:

(1) The hash value $g_i(o)$ is computed and all the objects in bucket $g_i(o)$ for $i = 1$,..., L are considered as candidates.
(2) The candidates are ranked according to their distances to object o, and k-nearest neighbors are found.

In the basic LSH technique above, increasing the number of hash tables L will increase the probability of finding all the nearest neighbors. However, this step entails space and more candidates. To solve these problems, the multi-probe LSH method (Lv et al. 2007; Dong et al. 2008) was proposed by Lv et al. The main idea of the multi-probe LSH is to probe multiple buckets in each hash table such that fewer hash tables are required than in the basic LSH method. This step is based on the fact that if an object is close to a query object q, but not hashed to the same bucket as q, it is likely in an adjacent bucket to the hash bucket of q.

The LSH technique is still evolving (Liu et al. 2014; Zhou et al. 2016), widely used, and suitable for solving ANN, retrieval, classification and other problems (Rao & Zhu 2016; Liu et al. 2015; Liao et al. 2016; Kraus et al. 2016).

3 STAR-IMAGE-CATALOG MATCHING PROCEDURE USING LSH

Before star-image-catalog matching, the CCD images should be processed for star coordinate extraction. Usually, the positional measurement of a star in a CCD frame has good precision, depending on its signal to noise ratio and the algorithm used. For our test stars, a bright star possesses a positional precision as good as 0.02–0.05 pixel but a faint star has slightly poorer precision, like 0.2–0.5 pixel (Li et al. 2009). On the other hand, a star catalog also usually has a very good accuracy in position. For example, the recently released catalog Gaia DR1 (Gaia Collaboration et al. 2016) has very good accuracy for its primary astrometric data, as good as 0.3 mas for the position and parallax uncertainty. However, these high precisions cannot be displayed in our raw pixel positions. Some systematic errors obviously exist in these raw pixel positions, such as geometric distortion (Peng et al. 2012). Sometimes, this systematic error will reach as great as 1–3 pixels (Zhang et al. 2012). Therefore, we have to resort to some algorithm to overcome these uncertainties.

In the study, supposing that the coordinates and measured magnitudes of all stars in an image have been obtained, we focus on matching stars in the image with stars in a stellar catalog.

3.1 Feature Representation of a Quadrilateral in a Stellar Catalog and an Image

In our research, we employ the quadrilateral matching technique, where six features are mainly used for representing a quadrilateral. The more features (except star identifiers) that are present means there are more distinctive quadrilaterals of stars.

Definition 4: A quadrilateral of four stars is defined as (ID_1, ID_2, ID_3, ID_4, L_1, L_2, L_3, L_4, L_5, L_6), where ID_1, ID_2, ID_3 and ID_4 are the identifiers of the four stars, and L_1, L_2, L_3, L_4, L_5 and L_6, are the lengths of four sides and two diagonals of a quadrilateral.

3.2 Matching Based on Basic LSH

The procedure of matching based on LSH is as follows:

(1) Reference stars are selected from a star catalog according to the telescope pointing direction, field of view and limiting magnitude. These stars are used to build the navigation database, Guide_DB.
(2) A quadrilateral feature database Quad_DB is created based on the stars in Guide_DB, and each quadrilateral is represented as the features given in definition 4.
(3) The background stars are sorted from the CCD image based on measured magnitudes.
(4) The four stars are selected from the CCD image according to brightness to construct quadrilateral q and create its feature vector o, which consists of the features defined in definition 4.
(5) A set of locality-sensitive function families H is generated according to Equation (1), and compound hash function family G is defined.
(6) g_1, g_2,..., g_L ($L \geq 1$) are selected at random from G independently and uniformly, and L hash tables are constructed.
(7) For each quadrilateral in Quad_DB, its feature vector is hashed into L buckets according to g_1, g_2,..., g_L.
(8) g_1, g_2,..., g_L are used to map o into L buckets, and all objects (except o) are retrieved from the L buckets and their corresponding quadrilaterals are gathered into a candidate set C. Duplicate quadrilaterals in C are removed. If C is empty, go to step (4).
(9) For each quadrilateral in C, the distance to q is computed, and quadrilaterals in C are sorted by their distances.
(10) One quadrilateral in C is selected according to distance (starting from the nearest). To verify whether similar asterisms (q and the quadrilateral selected from C) are really matched, the transformation relationship between the coordinate systems of the image and the star catalog is computed by the six-plate constant model introduced in Zhao (1987). If the transformed coordinates of most stars (above 70% in our experiments) in the CCD image correspond to star coordinates in the star catalog, then the verification is successful; otherwise, the verification fails, and the next quadrilateral in C is selected.
(11) If the verification in step (10) is not successful, the above steps from step (4) are repeated until one verification is successful or all quadrilaterals in the CCD image have been used.

Figure 1 presents an example of matching based on basic LSH, where $L = 3$, and $\star$ and $\bullet$ represent either quadrilaterals or feature vectors in two coordinate systems. A range of objects in a stellar catalog D is inserted into three hash tables corresponding to hash functions g_1, g_2 and g_3. Given object o in image I, to find its matching star from D, buckets $g_1(o)$, $g_1(o)$ and $g_1(o)$ are searched in three hash tables, and candidate set $C = \{a, b, c, e\}$ is obtained. Then the issue of whether o and one of the objects in C are really matched is verified.

3.3 Time Consumption Analysis

The following parameters and identifiers are given:

• N_1: number of bright stars in the CCD image used to construct quadrilaterals;

• N_2: number of selected reference stars in Guide_DB;

• B: non-redundant object number in several buckets that a quadrilateral from the image is mapped into, that is, the size of candidate set C in HashQuad;

• V: number of times needed to verify whether the transformation of the two coordinate systems (i.e., image coordinates and celestial coordinates in the catalog) is correct.

The matching process consists of three main time-consuming steps: constructing quadrilaterals, searching similar candidate quadrilaterals from Guide_DB and comparing with quadrilaterals from the image (called search and comparison below), and conducting transformation verification (called verification below).

(1) Analysis of construction time. The maximal number of constructed quadrilaterals from stars in Guide_DB is $N_2(N_2 - 1)(N_2 - 2)(N_2 - 3)/24$, and the maximal number of constructed quadrilaterals from stars on the CCD image is $N_1(N_1-1)(N_1-2)(N_1 - 3)/24$. Therefore, the construction time is $O(N_1^4 + N_2^4)$.
(2) Analysis of search and comparison time. If the quadrilaterals from the CCD image are directly compared with all quadrilaterals from Guide_DB (called Q-I below), the time required for searching and comparing is $O(N_1^4 N_2^4)$. The quadrilateral method based on the kd-tree (called Q-II below) dramatically accelerates searching and comparing to $O((N_1^4 + N_2^4) \log N_2)$. By contrast, the searching time for HashQuad is $O((N_1^4 + N_2^4) \log B)$. Generally, $B \ll N_2$, and thus HashQuad has higher searching and comparing efficiency than Q-I and Q-II.
(3) Analysis of verification time. To verify whether the coordinates of most objects on the CCD image correspond to star coordinates in the catalog, the maximum comparison times for any approach is VN_1N_2,

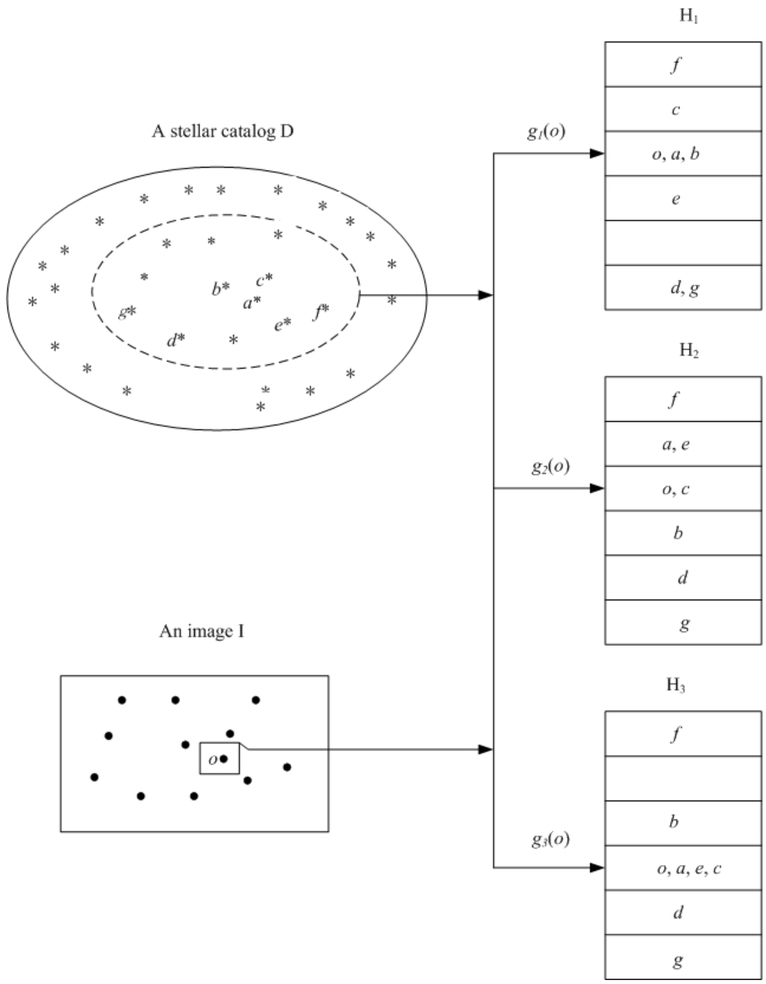

Fig. 1 Basic LSH and searching of star neighbors.

and the verification time is $O(VN_1N_2)$. Compared to construction and search time, complexity of verification is much lower.

From the above analysis, we present the time consumption of the three quadrilateral-based algorithms in Table 5. HashQuad has the lowest time consumption among the methods. The experimental results in Section 5 similarly demonstrate that HashQuad is the most efficient in most cases.

4 EXPERIMENTAL RESULTS AND ANALYSIS

We conduct several experiments to test the performance of the matching approaches. CCD images are selected from the images of NGC 6709 and M67 star clusters taken by the 1-meter telescope administered by Yunnan Observatories. The images are firstly processed for getting the coordinates and magnitudes of observed stars in the image coordinate system, which are used for computing features of the selected quadrilaterals on the images. For example, Figure 2 includes one captured image of star cluster M67. After processing it, each star in the image yields information, namely identifier, X-coordinate, Y-coordinate and magnitude, which is then saved in a table.

Navigation database Guide_DB extracts data from the GAIA catalog (*http://gaia.esac.esa.int/documentation/GDR1/Miscellaneous/seccreditandcitationinstructions.html*), which includes information about a star, namely identifier, right ascension, declination and magnitude. These are used for computing features of the quadrilaterals built from Guide_DB. Hundreds or thousands of bright stars are chosen from the GAIA catalog for creating the navigation database Guide_DB according to the region of each star cluster with a one degree field of view. The numbers of quadrilaterals built from Guide_DB are 2 302 786 for M67 and 1 361 992 for NGC 6709.

Three algorithms (Q-I, Q-II, HashQuad) were implemented in Java, and the experiments were conducted us-

Table 5 Time Complexity of Three Quadrilateral Based Algorithms

	Q-I	Q-II	HashQuad
Construction time	$O(N_1^4 + N_2^4)$	$O(N_1^4 + N_2^4)$	$O(N_1^4 + N_2^4)$
Search and comparison time	$O(N_1^4 N_2^4)$	$O((N_1^4 + N_2^4)\log N_2)$	$O((N_1^4 + N_2^4)\log B)$
Verification time	$O(VN_1N_2)$	$O(VN_1N_2)$	$O(VN_1N_2)$

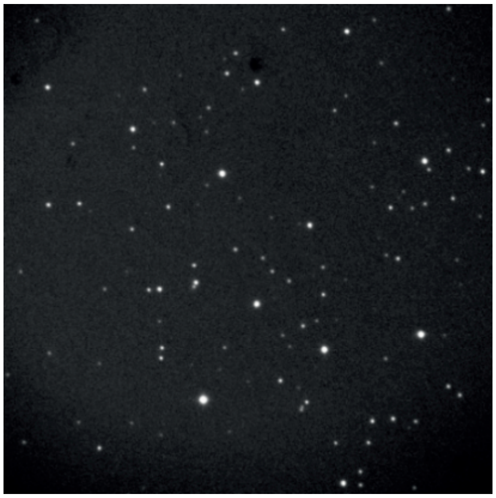

Fig. 2 One captured image of star cluster M67.

ing an Intel(R) Core(TM) i7-6700 CPU @ 3.4 GHz computer running Microsoft Windows 10.

Q-I, Q-II and HashQuad are all based on quadrilaterals. The indexing and searching approaches of their quadrilateral feature database are different. In Q-I, the feature database is sorted and binary search is used in the quadrilateral matching process (Lin et al. 2000). In Q-II, a kd tree is built as the index of the feature database (Heyl 2013), and the nearest search based on the kd tree is applied in matching quadrilaterals.

Each record in the quadrilateral feature databases for M67 and NGC 6709 is mapped into a bucket using LSH. 2-stable hash functions are applied in constructing one hash table. In the hash function given by Equation (1), we set: $W = 300, b = 56$.

The number of buckets produced and the size of the buckets (average number of star objects in a bucket) for the two star clusters are shown in Figures 3 and 4, respectively. The selected objects in the M67 star cluster are organized into 410 buckets, and the selected objects in the NGC 6709 star cluster are placed in 405 buckets. The number of objects in each bucket is obviously different. HashQuad maps a selected quadrilateral on the CCD image into one bucket, and the search range is only in this one bucket, instead of searching the entire quadrilateral feature database like Q-I.

We evaluate the performance of three algorithms according to the quadrilateral feature database construction time, search and comparison time, and searching times (number of searching quadrilaterals from Guide_DB for matching with the quadrilaterals in the CCD image). In addition, a selected quadrilateral in the CCD image may fail to match with the quadrilaterals from Guide_DB, and one or more quadrilaterals in the CCD image will be chosen. We analyze the number of selected quadrilaterals in the CCD image till the validation succeeds. The experimental results for 20 images of each star cluster are shown in Tables 1 to 9.

From Tables 1 to 9, the following conclusions can be drawn:

(1) In terms of search and comparison time, although HashQuad is not always less than Q-I or Q-II, the best performance is obtained for most of the images, and its total average time (validation time is ignored), including the feature database construction time, is the least among the three algorithms.

(2) The search times of HashQuad are less than those of Q-I, because Q-I searches similar quadrilaterals in the whole feature database, whereas HashQuad searches similar quadrilaterals in one bucket. For some images, HashQuad needs many more search times than Q-II, because the size of their hashing buckets is big.

(3) HashQuad requires minimal time to build the feature database because it only performs simple mappings using the LSH. Comparatively, Q-II needs to build a kd tree for the feature database and takes the most time.

(4) All images are matched successfully, but the number of selected quadrilaterals in each CCD image may be different. Except for image No.18 and No.19 of NGC 6709, Q-I and HashQuad have almost the same number of quadrilaterals selected in a CCD image. Theoretically, the LSH method is based on the p-stable hash function defined in Equation (1), and although it has a high probability to hash similar objects into the same bucket, similar objects may also be assigned to different buckets. The experimental results demonstrate that although the search range

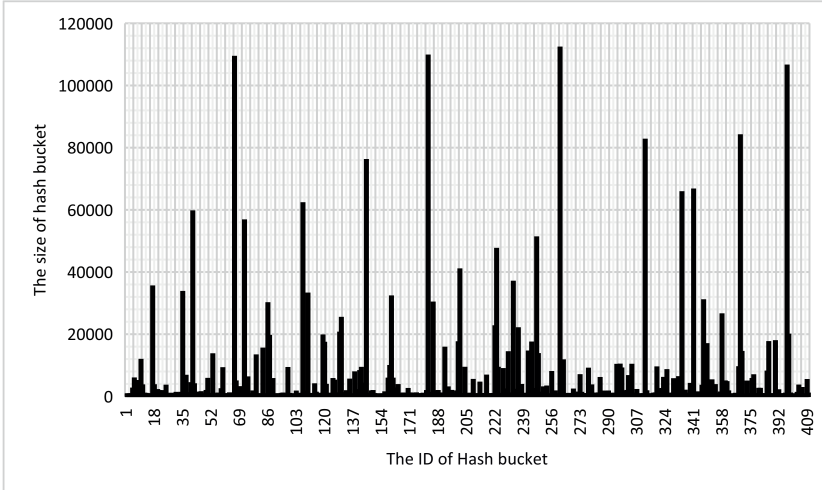

Fig. 3 Hash buckets produced by LSH for M67.

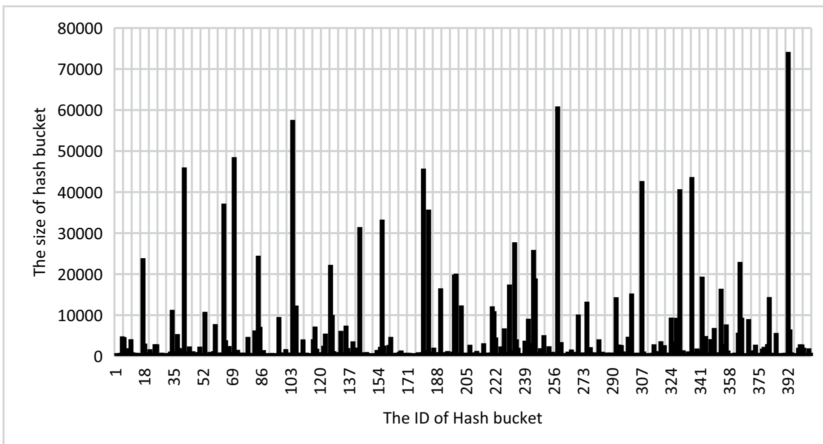

Fig. 4 Hash buckets produced by LSH for NGC 6709.

Table 6 Average Performance of Three Algorithms for M67

	Q-I	Q-II	HashQuad
Average time of constructing the feature database (ms)	1067	29 729	702
Average time of search and comparison (ms)	10.73	3.81	8.32
Total (ms)	1077.73	29732.81	710.32

is small, in most cases, HashQuad does not require more quadrilaterals to successfully make a CCD image match.

5 CONCLUSIONS

The position and brightness of stars are important information for astronomical research. The information extracted from CCD images of a dense star field is usually matched to the corresponding information from a stellar catalog. Various catalog matching algorithms have been developed. Given the big navigation feature database based on quadrilaterals, this paper presents an LSH-based algorithm to solve the star-image-catalog matching problem. The efficiency of the quadrilateral algorithm is improved by the proposed algorithm through hashing quadrilateral models in the navigation feature database to different hash buckets to reduce the search range. In the future, we plan to investigate further research on LSH-based methods. More parameters will be tested, and the application of multi-LSH techniques in the star-image-catalog matching problem will be explored to address missing matches.

Table 7 Average Performance of Three Algorithms for NGC 6709

	Q-I	Q-II	HashQuad
Average time of constructing the feature database (ms)	1870	12 352	1511
Average time of search and comparison (ms)	19.04	13.48	36.53
Total (ms)	1889.04	12365.48	1547.53

Table 8 The Number of Selected Quadrilaterals in the CCD Images of M67

Image ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Q-I	1	1	1	154	2	2	1	18	1	1	1	1	1	1	1	1	1	1	1	18
Q-II	3	2	2	154	2	2	1	18	1	1	3	3	3	3	3	3	3	3	1	18
HashQuad	1	1	1	154	3	3	1	18	1	1	1	1	1	1	1	1	1	1	1	19

Table 9 The Number of Selected Quadrilaterals in the CCD Images of NGC 6709

Image ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Q-I	3	2	2	3	3	3	3	3	9	13	3	3	2	2	2	2	10	2	4	3
Q-II	4	12	10	4	3	3	3	3	10	13	4	4	5	5	6	6	12	11	22	4
HashQuad	4	2	2	4	5	5	5	5	9	13	3	3	4	4	4	4	10	35	114	3

Acknowledgements This was supported by the National Natural Science Foundation of China (U1431227) and Guangzhou Science and Technology Planning Project (201604010037). We thank Dr. Na Wang of Jinan University for providing several suggestions and experimental images.

References

Bentley, J. L. 1975, Communications of the ACM, 18, 509

Cole, C. L., & Crassidis, J. L. 2006, Journal of Guidance Control Dynamics, 29, 64

Datar, M., Immorlica, N., Indyk, P., & Mirrokni, V. S. 2004, in Proceedings of the Twentieth Annual Symposium on Computational Geometry, ACM, 253

Dong, W., Wang, Z., Josephson, W., Charikar, M., & Li, K. 2008, in Proceedings of the 17th ACM Conference on Information and Knowledge Management, ACM, 669

Gaia Collaboration, Brown, A. G. A., Vallenari, A., et al. 2016, A&A, 595, A2

Groth, E. J. 1986, AJ, 91, 1244

Heyl, J. S. 2013, MNRAS, 433, 935

Indyk, P., & Motwani, R. 1998, in Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing, ACM, 604, 613

Ji, F., Jiang, J., & Wei, X. 2013, in Imaging Systems and Techniques (IST), 2013 IEEE International Conference on, IEEE, 228

Kraus, N., Carmel, D., Keidar, I., & Orenbach, M. 2016, in Proceedings of the 2016 International Conference on Similarity Search and Applications, Springer, 236

Lang, D., Hogg, D. W., Mierle, K., Blanton, M., & Roweis, S. 2010, AJ, 139, 1782

Li, Z., Peng, Q. Y., & Han, G. Q. 2009, Acta Astronomica Sinica, 50, 340

Liao, J., Yang, D., Li, T., et al. 2016, Multimedia Tools and Applications, 75, 15405

Lin, T., Zhou, J., & Zhang, J. 2000, Journal of Astronautics, 21, 82

Liu, Y., Cui, J., Huang, Z., Li, H., & Shen, H. T. 2014, Proceedings of the VLDB Endowment, 7, 745

Liu, J., Sun, H., & Ding, Z. 2015, in International Conference of Young Computer Scientists, Engineers and Educators, Springer, 250

Lu, Y., Zhang, X. X., & Sun, R. Y. 2015, Acta Astronomica Sinica, 56, 399

Lv, Q., Josephson, W., Wang, Z., Charikar, M., & Li, K. 2007, in Proceedings of the 33rd International Conference on Very Large Data Bases, VLDB Endowment, 950

Matsushita, Y., & Wada, T. 2009, Advances in Image and Video Technology, 374

Mortari, D., Samaan, M. A., Bruccoleri, C., & Junkins, J. L. 2004, Navigation, 51, 171

Na, M., Zheng, D., & Jia, P. 2009, IEEE Transactions on Aerospace and Electronic Systems, 45, 516

Padgett, C., & Kreutz-Delgado, K. 1997, IEEE Transactions on Aerospace and Electronic Systems, 33, 202

Peng, Q. Y., Vienne, A., Zhang, Q. F., et al. 2012, AJ, 144, 170

Qi, N., Xia, Q., Guo, J., et al. 2014, in Mechatronics and Control (ICMC), 2014 International Conference on, IEEE, 919

Rao, B., & Zhu, E. 2016, in Proceedings of the 2016 International Conference on Management of Data, ACM, 2257

Slaney, M., & Casey, M. 2008, IEEE Signal Processing Magazine, 25, 128

Tabur, V. 2007, PASA, 24, 189

Tang, W., Yang, J., Yi, W., Jia, H., & Cheng, P. 2016, Laser & Optoelectronics Progress (in Chinese), 53, 021002

Xie, J., Tang, X., Jiang, W., & Fu, X. 2012, ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 333

Zhang, G., Wei, X., & Jiang, J. 2008, Image and Vision Computing, 26, 891

Zhang, Q.-F., Peng, Q.-Y., & Zhu, Z. 2012, RAA (Research in Astronomy and Astrophysics), 12, 1451

Zhao, H.-Q. 1987, Acta Astronomica Sinica, 28, 190

Zhou, Y., Liu, C., Li, N., & Li, M. 2016, Remote Sensing Letters, 7, 965

Zolotarev, V. 1986, One-dimensional Stable Distributions (Translated from the Russian by HH McFaden), Translations of Mathematical Monographs, 65, ed. B. Silver (Providence: American Mathematical Society)