Estimating camera parameters from starry night photographs

Naoto Ishikawa¹ (✉), Yoshinori Dobashi¹

© The Author(s) 2020.

Abstract We propose an efficient, specific method for estimating camera parameters from a single starry night image. Such an image consists of a collection of disks representing stars, so traditional estimation methods for common pictures do not work. Our method uses a database, a star catalog, that stores the positions of stars on the celestial sphere. Our method computes magnitudes (i.e., brightnesses) of stars in the input image and uses them to find the corresponding stars in the star catalog. Camera parameters can then be estimated by a simple geometric calculation. Our method is over ten times faster and more accurate than a previous method.

Keywords astrophotography; constellation; star identification; pattern matching

1 Introduction

A wide-angle photograph that includes both a starry night sky and the landscape beneath it is called a starscape. The beauty of starscapes makes them popular subjects for photographers. Moreover, an image containing stars can be used to determine the location of the viewer by identifying the stars or constellations in the image. In fact, in the past, stars and constellations played an important role in the navigation of ships. Even today, a device called a star tracker is used in spacecraft to determine its attitude by tracking the configurations of stars [1, 2]. However, for these methods, it is usually assumed that certain camera parameters, such as the focal length, are known. This paper focuses on estimating the camera parameters from a single starscape without

any prior knowledge of the camera.

In the field of computer vision, many methods have been developed for determining camera parameters from photographs [3–6]. However, these methods focus on images of ordinary objects such as buildings or landscapes, and are unsuitable for images of starry skies because their characteristics are completely different from those of ordinary pictures. Starry night sky images consist solely of small dots corresponding to stars, making traditional computer vision techniques useless. To address this problem, we make use of a publicly-available star catalog that records the coordinates of all stars above some brightness. We can assume that the stars are placed on a virtual sphere (the celestial sphere), with an infinite radius, centered on the Earth. The star catalog stores the coordinates of the stars on the celestial sphere. By matching the configuration of stars in the starry night image with those in the star catalog, we can determine the camera parameters. The problem, then, is to find the stars in the catalog that correspond to those in the image. Several methods have been proposed for doing so [1, 7–9]. These methods use geometric features such as triangles or quadrilaterals of stars in the image to find the corresponding stars in the star catalog. However, the search can be slow; often tens of seconds are required.

We propose an accurate and efficient method for determining the camera parameters from a single starry night image. We also use a star catalog. The proposed method firstly detects the stars in the image using image processing techniques. Next, the magnitudes (or brightnesses) of the detected stars are used to find corresponding stars in the star catalog. Having found the corresponding stars in the star catalog, we can compute the camera parameters, such as the focal point and orientation, by a simple

¹ Graduate School of Information Science and Technology, Hokkaido University, Sapporo, Japan. E-mail: N. Ishikawa, ishikawa@ime.ist.hokudai.ac.jp (✉); Y. Dobashi, doba@ime.ist.hokudai.ac.jp.

Manuscript received: 2020-07-15; accepted: 2020-09-18

geometric calculation. However, there are many candidate matches in the star catalog for the stars in the input image. So, we have developed a fast refinement algorithm to identify the most plausible correspondence. Our experiments demonstrate that our method is more accurate and dozens of times faster than previous methods.

This paper is organized as follows. In Section 2, we discuss related work for determining camera parameters from night sky images. In Section 3 we present an overview of our method; details are explained in Sections 4 and 5. Section 6 shows some experiments that verify the effectiveness of our method. Finally, our conclusions are given in Section 7.

2 Related work

The determination of camera parameters from starry sky images has been used to control the attitude of spacecraft. This task is known as blind astrometric calibration. The pose of the camera mounted on a spacecraft is determined by identifying the visible stars [1, 7]. There is plenty of work on this topic; it can be roughly classified into geometric and pattern-based approaches. Geometric approaches use geometric features such as triangles of stars to estimate the parameters [1, 10–12]. Pattern-based approaches use a pattern code based on the positions of neighboring stars in the image [13–18]. However, these methods are not fully blind because they use cameras specifically designed for the spacecraft. Pái and Bakos proposed a blind estimation method that starts with an initial estimate which is then refined [8]; however, it requires a good initial estimate. A more detailed survey can be found in Refs. [19, 20].

One of the most widely-used and robust systems for blind estimation is a system called Astrometry.net [9]. While previous methods used triangles of stars as geometric features for estimation, Astrometry.net uses quadrilaterals to improve robustness of estimation. The system detects the stars in the image and creates a list of quadrilaterals combining four stars. Then, for each quadrilateral, a geometric hash code is generated using the distances and angles between the four stars. The hash code is compared with ones precomputed using the star catalog to find the closest quadrilateral in the star catalog. The system identifies the part of the sky captured by

the image using the correspondence between the quadrilaterals in the image and the star catalog. However, the hash codes using the catalog must be precomputed, which is time-consuming. Estimation also takes a long time, often requiring tens of seconds. Recently, Ajdad et al. proposed a star identification algorithm for images captured with wide field of view (FOV) cameras [21]. It takes into account errors caused by lens distortion of the wide FOV camera. The method uses interior angles of triangles of stars in the image to look for corresponding stars in a database. It iteratively updates the projection function to improve the estimation accuracy. Once a precise projection is obtained, the method is very efficient. However, it may be inaccurate and is slow for a new camera setup. Although our method is not suitable for a wide FOV image, it can be used to provide a good initial estimate for this method.

3 Overview of the proposed method

Our proposed method takes a single image as input and estimates the camera parameters. It comprises two processes: detection of the stars in the input image, and parameter estimation using the detected stars.

The star detection process starts by removing noise from the input image. Then, stars are detected using the fact that their shapes are circular. The area of each star is used to estimate the star's magnitude. This logarithmic astronomical unit measures the brightness of a star; a more positive value means the star is dimmer. The brightest star in the Hipparcos catalog is Sirius, with magnitude -1.46 .

Next, in the estimation process, the camera parameters are calculated. We use a publicly-available star catalog in which the coordinates of all the stars on the celestial sphere (above a given magnitude) are recorded. First the stars in the catalog corresponding to the stars detected in the image are found, and the camera parameters can then be determined analytically by simple geometric calculations. However, finding the corresponding stars is not trivial; it is not obvious which stars in the catalog correspond to the stars detected in the image. Thus, in our method, a search for the best correspondence is undertaken. The basic idea is to compute a set of camera parameters for all possible combinations of correspondences and

choose the most-likely parameters from amongst them. However, it is impractical and time-consuming to take all correspondences into account since the number of correspondences is huge. Thus, we use the magnitudes and geometric relationships of the stars to reduce the number of correspondences to be considered.

We use the Hipparcos catalog [22] for the star catalog. This widely used catalog includes the coordinates and magnitudes of 118,218 stars. Assuming that the input is a relatively wide-angle image, we use 285 stars with magnitudes less than 3.5.

4 Star detection

Since an input starry sky image often includes objects on the ground, we need to eliminate these objects to accurately identify pixels corresponding to stars in the image. We apply a set of image processing operations to do so. Figure 1 illustrates this process.

First, we convert the input image to a grayscale image: see Fig. 1(a). We then apply a Gaussian blur filter to this image: see Fig. 1(b). The blurred image is then subtracted from the grayscale image to remove low frequency components: see Fig. 1(c). Note that negative intensities are clamped to zero. The kernel size of the Gaussian filter is $1/8$ of the length of the shorter side of the image. Next, the resulting image is converted into a binary image as shown in Fig. 1(d). The threshold used to do so is determined as in Ref. [9], using the mean μ and standard deviation σ of the difference image in Fig. 1(c). The threshold is set to $\mu + 8\sigma$; these parameters are determined experimentally.

Next, we detect connected regions in the binary image and extract their contours. Contour circularity is used to distinguish stars from objects on the ground. Circularity is defined as $4\pi A/P^2$ where A and P are the shape's area and the perimeter, respectively; it has value 1 for a circle. We exclude contours whose circularity is smaller than 0.55. For each of the remaining contours, we compute the smallest circumscribed circle (i.e., the smallest circle enclosing the shape). Its center and radius are used as the coordinate and radius of the corresponding star, respectively. Stars with extremely large radii are also excluded from subsequent processes since they are likely to be objects in the landscape. Figure 1 shows the stars detected by our method.

Fig. 1 Star detection process.

5 Estimation of camera parameters

5.1 Parameters to be estimated

First, we define the camera parameters estimated by our method. As shown in Fig. 2(b), we assume that the image plane is tangential to the celestial sphere and the images of the stars are their perspective projections onto the image plane. The stars are assumed to be on the celestial sphere. The camera parameters determine the position, the orientation, and the size of the image plane. Our method estimates four parameters: the right ascension and the declination of the tangent point (the center of the image plane), the camera orientation, and the size of the image plane. The right ascension and declination are an equatorial coordinate system that describes a point on the celestial sphere. The camera orientation is defined by its rotation angle around the viewing direction. The size of the image plane corresponds to

(a) a pair of stars in the input image

(b) corresponding 3D coordinates of the stars in (a)

Fig. 2 Estimation of the camera parameters.

the viewing angle and is defined as the length of the short side of the image plane.

5.2 Calculation of the camera parameters

We now describe the method for calculating the camera parameters, assuming a pair of stars in the image correspond to known stars on the celestial sphere. See Fig. 2(a). Let the screen-space coordinates of a pair of stars by a and b , respectively, where the origin o is at the center of the image. The coordinates of their corresponding stars on the celestial sphere are denoted A and B , respectively (see Fig. 2(b)). First, we compute the center of the image plane (i.e., the tangent point) and then compute the camera orientation and size of the image plane. The details are described below.

Refer to Fig. 2(a). Consider a line perpendicular to the line between a and b that passes through o . The intersection between these lines is denoted q . If we have the three-dimensional coordinates of A and B corresponding to a and b , respectively, the three-dimensional coordinates of Q corresponding to q are given by

$$\mathbf{OQ} = \mathbf{OA} + \frac{|\mathbf{aq}|}{|\mathbf{ab}|} \mathbf{AB} \quad (1)$$

Consider the viewing ray from the camera position O to the center of the image plane, $\mathbf{OR}$. This ray is in the plane containing Q and is perpendicular to $\mathbf{AB}$. The perpendicular distance from Q to the ray is given by

$$d = |\mathbf{AB}| \frac{|\mathbf{oq}|}{|\mathbf{ab}|} \quad (2)$$

If we denote the unit vector in the direction of $\mathbf{OR}$ by $\mathbf{R}_u$, $\mathbf{R}_u$ satisfies the following equations:

$$(\mathbf{OQ} \cdot \mathbf{R}_u)^2 + d^2 = |\mathbf{OQ}|^2 \quad (3a)$$

$$\mathbf{R}_u \cdot \mathbf{AB} = 0 \quad (3b)$$

$$|\mathbf{R}_u|^2 = 1 \quad (3c)$$

We obtain $\mathbf{R}_u$ by solving these equations. Then we compute the intersection of the ray with the celestial sphere, which gives us the center of the image plane, R .

Next, we compute α and β , which are the projected positions onto the image plane, of A and B , respectively (see Fig. 2(b)). Then, from the relationship between $\triangle oab$ and $\triangle R\alpha\beta$, we can compute the camera orientation and the size of the image plane.

5.3 Search for corresponding stars

5.3.1 Discussion

To obtain the camera parameters, we need to correctly identify the stars on the celestial sphere that correspond to the stars in the image, which unfortunately, is not possible in many cases. We have to search for the correct correspondence from a number of possible candidates. Our approach is to generate a set of candidates for the corresponding stars and compute the parameters for each set using the method described in the previous section. When there are multiple candidates that result in similar camera parameters, they should give us the best solution.

A possible approach to generate a list of candidates is to use RANSAC [23]. RANSAC is a robust

estimation algorithm for data with outliers, which it can remove by random sampling. In our case, candidates could be generated by randomly sampling stars from the star catalog. However, this approach does not guarantee that the correct corresponding stars will be included in the candidate list provided by random sampling. We need to include the correct correspondence in the list to obtain the accurate camera parameters. So, RANSAC is not a good choice for our purpose. We need to check all the possible correspondences to be certain to include the correct correspondence. However, since this is extremely time-consuming, our algorithm reduces the cost by excluding those stars that cannot be candidates by making use of their magnitudes and relative positions. We describe the details of the method in the following subsections.

5.3.2 Candidate correspondences

The magnitudes of stars are used to generate candidates. We assume that the apparent brightness of a star detected in the image is proportional to its area, or pixel count. However, it is difficult to estimate the absolute magnitude accurately because the apparent brightness depends on the exposure setting of the camera, which is unknown. We therefore use the difference in magnitudes of a pair of stars to generate candidates; this depends on their brightness ratio. For this purpose, we use Pogson's equation [24], which defines the relationship between magnitude and apparent brightness:

$$m - n = -2.5 \log_{10} \left(\frac{l_m}{l_n} \right) \quad (4)$$

where l_m and l_n are the apparent brightnesses of stars whose magnitudes are m and n , respectively. We use the areas of two stars detected in the input image for l_m and l_n . Thus, the magnitude difference $m - n$ is approximated by using the above equation with $l_m/l_n \approx A_m/A_n$, where A_m and A_n are the areas of the two stars in the image.

The candidate correspondences are then generated as follows. We first pick a pair of stars detected in the input image and compute their difference in magnitude. Next, pairs of stars with a similar magnitude difference are extracted from the star catalog as candidates. The candidates can be generated efficiently by sorting the stars in the catalog by magnitude in advance. However, our experiments showed that this simple approach produced erroneous

candidates due to lens filters being used when capturing the images, and atmospheric conditions. As a result, too many candidates were generated. We therefore introduced a tolerance on the magnitude to remove such erroneous candidates. We accept a pair of stars in the star catalog as a valid candidate only if their magnitude difference d_c satisfies $d - 2 < d_c < d + 2$ where $d = m - n$. Furthermore, the stars used for the candidates in the star catalog are limited to the 285 stars with magnitudes less than 3.5. Usually, a starscape photograph includes about 10 stars that have magnitudes less than 3.5 and this is a sufficient number of stars for generating valid candidates.

5.3.3 Using triplets

The method described in the previous section generates a list of candidate correspondences from the star catalog for a pair of stars detected in the image. Each of the candidates in the list is a pair of stars in the catalog whose magnitude difference is similar to a pair of stars in the input image. Since we only consider magnitude differences, the candidate list includes incorrect correspondences. We need to identify the reliable candidates in this list so we can eventually determine the correct camera parameters. In this section we explain our method for examining the reliability of the candidates. The key to our method is the use of triplets of stars detected in the image, which produces two pairs of stars sharing a single star. Our method finds reliable triplets from the star catalog that correspond to the triplet detected in the image by taking into account the fact that one of the stars must be shared. The camera parameters are determined from the reliable triplets using the method described in the next section.

Figure 3 shows how the reliable triplets are selected. Consider a triplet $\{a, b, c\}$ detected in the input image. The goal here is to find corresponding reliable triplets in the star catalog. Note that there could be multiple reliable triplets in the star catalog for a given triplet in the input image. From triplet $\{a, b, c\}$, we can generate two pairs of stars, $P_a\{a, b\}$ and $P_c\{b, c\}$, where b is shared between these two pairs. For each of the pairs, we generate a list of candidate pairs in the star catalog using their magnitude differences as described in the previous section. We then compute the camera parameters for each of the candidates in the list using the method described in Section 5.2; these are denoted $S_a = \{S_{a1}, S_{a2}, \dots\}$ and $S_c =$

$\{S_{c1}, S_{c2}, \dots\}$, respectively. Each element in these sets is a four-dimensional vector consisting of four camera parameters. Next, for each element in S_a , we find the nearest element from S_c using a kd -tree [25]. Suppose S_{cj} is the nearest element to S_{ai} . Since P_a and P_c share a single star in the input image, we consider S_{ai} and S_{cj} as reliable only when they share a single star as well. If they do, then the stars used for computing S_{ai} and S_{cj} form a reliable triplet. We go through all the elements in S_a and collect reliable triplets in the same way. Figure 3 shows two examples; S_{ak} and S_{cm} are reliable while S_{aj} and S_{cl} are not.

We now have a list of reliable triplets for a given triplet $\{a, b, c\}$. For each reliable triplet, we have two sets of camera parameters, each of them computed by a pair of stars in the triplet. We then compute the average camera parameters for each of the two pairs and the distance between the camera parameters. The triplets are sorted by distance between them and we keep the top n_t of them together with the average camera parameters. In this paper, we set $n_t = 20$. These n_t sets of camera parameters are used to determine the final camera parameters in the next section.

5.3.4 Final camera parameters

We iteratively generate n_t sets of camera parameters by the method described in the previous section. We repeat this process until we find a pair of camera parameters that are sufficiently similar, measured by the angular differences in right ascension, declination, and camera orientation. We terminate iteration when these are less than one degree. Note that we do not use the size of the image plane for this termination criteria because sufficient accuracy has been achieved

in our preliminary experiments.

5.3.5 Selection of triplets

As mentioned above, our method iteratively chooses a triplet of stars detected in the input image. The efficiency and accuracy of the proposed method depends on how a triplet is chosen for each iteration. This section describes our method for selecting the triplets.

Our strategy is to use bright stars while avoiding using the same stars in different iterations. A bright star detected in the input image has a larger area than other stars, so the error in calculating the magnitude difference should be smaller: using bright stars for the triplets should produce better results. We should also use different stars in different iterations in order to increase variation in the triplets, preventing the solution from converging to a local minimum. Furthermore, using different stars reduces the influence of bright regions, e.g., light sources on the ground, erroneously detected as stars in the input image.

Using these ideas, we have developed a heuristic algorithm to choose the triplets. Table 1 shows examples of triplets generated by this approach. First, the detected stars are ranked according to their brightness. The stars in a triplet are denoted by their rank (i, j, k) and are arranged according to rank $(i < j < k)$. We compute the sum of the ranks of the first and second stars, $n = i + j$, and the difference between the ranks of the second and third stars, $m = k - j$. We first fix $m = 1$ and we choose a set of triplets with n less than a specified number. Then, m is incremented and we choose another set of triplets. In Table 1, we first generate six triplets with $m = 1$. Then m is incremented and two more

Fig. 3 Reliable triplets selection.

Table 1 Example of triplets selection

trial	n	m	triplet		
			i	j	k
1	3	1	1	2	3
2	4	1	1	3	4
3	5	1	1	4	5
4	5	1	2	3	4
5	6	1	1	5	6
6	6	1	2	4	5
7	3	2	1	2	4
8	4	2	1	3	5

triplets are generated. The maximum of n in each set is specified by the user.

6 Results

6.1 Setting

We compared our method with Astrometry.net [9] to investigate the effectiveness of our method. We used a laptop computer with an Intel Core i5-8250U with 16 GB of memory.

We prepared two datasets of images for the comparison. One consisted of synthetic images of stars created using the star catalog, and the other consisted of real images taken by a digital camera. For these datasets, the true camera parameters are available. To enable a fair comparison, in each case, to detect stars in the input image, and their coordinates, areas, and intensities, we used the method described in Section 4 both for our method and the Astrometry.net system. We measured the computation time and the accuracy of the estimated parameters. We considered the estimates to be sufficiently accurate when the errors fell below the tolerances in Table 2. With these tolerances, the input image and an image generated using the estimated parameters are visually indistinguishable.

Table 2 Tolerances for accuracy

parameter	tolerance
right ascension	2°
declination	2°
camera orientation	5°
view angle	3°

The index files used for Astrometry.net were also created from the Hipparcos star catalog for a fair comparison. Since our method takes into account only stars with magnitudes less than 3.5, we prepared the index data using these stars only.

6.2 Comparison using synthetic images

The synthetic images were created by perspective projection of the stars in the Hipparcos star catalog with brightnesses greater than a magnitude of 6.5. Each star was rendered as a white disk with size calculated using Pogson's equation (Eq. (4)). We created 1000 images rendered with randomly selected camera parameters. We limited the range of viewing angles to those of a typical starscape photograph, that is, from 50° to 110°, which is equivalent to a focal length ranging from 15 to 45 mm for a 35 mm full-frame camera. Figure 4 shows example synthetic images.

Table 3 shows the results of the comparison. The Astrometry.net system failed to find reliable parameters for 9.20% of the input images before the computation time reached a predetermined duration of 5 min. We excluded these results from Table 3. Accuracy is the proportion of images for which the tolerances were satisfied compared. This table shows our method is superior in terms of both accuracy and efficiency.

Table 3 Comparison using synthetic images

	Astrometry.net	Our method
average time	13.55 s	1.10 s
accuracy	70.20%	100.00%

Fig. 4 Example synthetic images.

6.3 Comparison using real images

For real images, we took 32 pictures of the night sky with a digital camera. We also used 5 images from Flickr, giving a total of 37 real images. The angles of view were wider than an equivalent focal length of 90 mm for a 35 mm full-frame camera. Images with strong distortion, e.g., from a fisheye lens, were not included. Figure 5 shows some examples of the images used.

Table 4 shows the results of the comparison. In terms of accuracy, our method was approximately 16% better than Astrometry.net. The computation time of our method is much faster than that of Astrometry.net; our method took only 8% of the time it required. For this dataset, Astrometry.net failed to find reliable parameters for 18.92% of the input images; Table 4 does not contain these results.

Figure 6 compares input images (left) with rendered images (right), using estimated camera parameters, and the method described in Section 6.2. Dashed circles indicate corresponding stars in these images. This figure demonstrates that our method successfully estimates accurate camera parameters.

Fig. 5 Example real images.

Table 4 Comparison using real images

	Astrometry.net	Our method
average time	32.59 s	2.86 s
accuracy	72.97%	89.19%

Fig. 6 Comparison of input and rendered images.

6.4 Experiments using different numbers of stars

Computation time and accuracy depend on the number of stars from the star catalog used in the estimation process. In order to investigate this dependency, we undertook an additional experiment. In the previous experiments, we took into account only stars with magnitudes less than 3.5. We increased the maximum magnitude from 2.4 to 4.8 in steps of 0.1, giving an increasing numbers of stars, and measured the computation time and accuracy using the dataset of real images. Figure 7 shows the results of this experiment. We see that the computation time increases exponentially as the number of the stars increases. However, our method achieves highest accuracy (94.59%) when the maximum magnitude is 3.6.

Fig. 7 Accuracy and computation time for different numbers of stars.

7 Conclusions and future work

In this paper, we proposed a method for estimating camera parameters from a starscape photograph. We have demonstrated that our method outperforms a previous method, the Astrometry.net system [9]. It requires no precomputation while Astrometry.net requires preprocessing to build index data.

There are a few things to be done in the future. Firstly, we need to improve our method of handling severe lens distortion. We assume perspective projection for the input images, but certain specific lenses, such as fisheye lenses, produce highly distorted images. In this case, our method fails to find accurate camera parameters. Secondly, we could use additional camera information, such as the focal length, the location, and the date on which the image was captured to obtain better estimates. This information is often stored as metadata together with the image.

References

- [1] Liebe, C. C. Pattern recognition of star constellations for spacecraft applications. *IEEE Aerospace and Electronic Systems Magazine* Vol. 8, No. 1, 31–39, 1993.
- [2] Lamy Au Rousseau, G.; Bostel, J.; Mazari, B. Star recognition algorithm for APS star tracker: Oriented triangles. *IEEE Aerospace and Electronic Systems Magazine* Vol. 20, No. 2, 27–31, 2005.
- [3] Salvi, J.; Armangué, X.; Batlle, J. A comparative review of camera calibrating methods with accuracy evaluation. *Pattern Recognition* Vol. 35, No. 7, 1617–1635, 2002.
- [4] Zhang, Z. *Camera Calibration*. Prentice Hall, 2005.
- [5] Remondino, F.; Fraser, C. Digital camera calibration methods: Considerations and comparisons. In: Proceedings of the ISPRS Commission V Symposium “Image Engineering and Vision Metrology”, 266–272, 2006.
- [6] Song, L.; Wu, W.; Guo, J.; Li, X. Survey on camera calibration technique. In: Proceedings of the 5th International Conference on Intelligent Human–Machine Systems and Cybernetics, 389–392, 2013.
- [7] Junkins, J.; White III, C. C.; Turner, J. D. Star pattern recognition for real time attitude determination. *Journal of the Astronautical Sciences* Vol. 25, No. 3, 251–270, 1977.
- [8] Pál, A.; Bakos, G. Á. Astrometry in wide-field surveys. *Publications of the Astronomical Society of the Pacific* Vol. 118, No. 848, 1474–1483, 2006.
- [9] Lang, D.; Hogg, D. W.; Mierle, K.; Blanton, M.; Roweis, S. Astrometry.net: Blind astrometric calibration of arbitrary astronomical images. *The Astronomical Journal* Vol. 139, No. 5, 1782–1800, 2010.
- [10] Mortari, D.; Samaan, M. A.; Bruccoleri, C.; Junkins, J. L. The pyramid star identification technique. *Navigation* Vol. 51, No. 3, 171–183, 2004.
- [11] Cole, C. L.; Crassidis, J. L. Fast star-pattern recognition using planar triangles. *Journal of Guidance, Control, and Dynamics* Vol. 29, No. 1, 64–71, 2006.
- [12] Kolomenkin, M.; Pollak, S.; Shimshoni, I.; Lindenbaum, M. Geometric voting algorithm for star trackers. *IEEE Transactions on Aerospace and Electronic Systems* Vol. 44, No. 2, 441–456, 2008.
- [13] Padgett, C.; Kreutz-Delgado, K. A grid algorithm for autonomous star identification. *IEEE Transactions on Aerospace and Electronic Systems* Vol. 33, No. 1, 202–213, 1997.
- [14] Yoon, Y. Autonomous star identification using pattern code. *IEEE Transactions on Aerospace and Electronic Systems* Vol. 49, No. 3, 2065–2072, 2013.
- [15] Na, M.; Zheng, D. N.; Jia, P. F. Modified grid algorithm for noisy all-sky autonomous star identification. *IEEE Transactions on Aerospace and Electronic Systems* Vol. 45, No. 2, 516–522, 2009.
- [16] Pham, M. D.; Low, K. S.; Chen, S. S. An autonomous star recognition algorithm with optimized database. *IEEE Transactions on Aerospace and Electronic Systems* Vol. 49, No. 3, 1467–1475, 2013.
- [17] Samirbhai, M. D.; Chen, S. S.; Low, K. S. A hamming distance and spearman correlation based star identification algorithm. *IEEE Transactions on Aerospace and Electronic Systems* Vol. 55, No. 1, 17–30, 2019.
- [18] Mehta, D. S.; Chen, S. S.; Low, K. S. A rotation-invariant additive vector sequence based star pattern recognition. *IEEE Transactions on Aerospace and Electronic Systems* Vol. 55, No. 2, 689–705, 2019.
- [19] Spratling, B. IV, Mortari, D. A survey on star identification algorithms. *Algorithms* Vol. 2, No. 1, 93–107, 2009.
- [20] Ho, K. A survey of algorithms for star identification with low-cost star trackers. *Acta Astronautica* Vol. 73, 156–163, 2012.
- [21] Ajdadi, M. J.; Ghafarzadeh, M.; Taheri, M.; Mosadeq, E.; Ghomi, M. K. Star identification algorithm for uncalibrated, wide fov cameras. *The Astronomical Journal* Vol. 149, No. 6, 182, 2015.
- [22] Perryman, M. A. P.; Lindegren, L.; Kovalevsky, J.; Hög, E.; Bastian, U.; Bernacca, P. L.; Crézé, M.; Donati, F.; Grenon, M.; van Leeuwen, F. et al. The hipparcos catalogue. *Astronomy and Astrophysics* Vol. 500, 501–504, 1997.

- [23] Fischler, M. A.; Bolles, R. C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* Vol. 24, No. 6, 381–395, 1981.
- [24] Pogson, N. Magnitudes of thirty-six of the minor planets for the first day of each month of the year 1857. *Monthly Notices of the Royal Astronomical Society* Vol. 17, No. 1, 12–15, 1856.
- [25] Bentley, J. L. Multidimensional binary search trees used for associative searching. *Communications of the ACM* Vol. 18, No. 9, 509–517, 1975.

Naoto Ishikawa is a master's course student at Hokkaido University in the Graduate School of Information Science and Technology. He is interested in image processing.

Yoshinori Dobashi has been an associate professor at Hokkaido University in the Graduate School of Information Science and Technology, Japan since 2006. His research interests center on computer graphics, including realistic image synthesis, efficient rendering, and sound modeling for virtual reality applications. Dobashi received his B.E., M.E. and Ph.D. degrees in engineering in 1992, 1994, and 1997, respectively, from Hiroshima University. He worked at Hiroshima City University from 1997 to 2000 as a research associate.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made.

The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Other papers from this open access journal are available free of charge from <http://www.springer.com/journal/41095>. To submit a manuscript, please go to <https://www.editorialmanager.com/cvmj>.