Lunar Image Matching Based on FAST Features with Adaptive Threshold

You Zhai[1](✉), Shuai Liu[2], Xiwei Guo[1], Peng He[1],
and Zhuanghe Zhang[1]

[1] Shijiazhuang Campus, Army Engineering University, Shijiazhuang, China
youyou1952@sina.com
[2] Technical Division, 66132 Troops, Beijing, China

Abstract. The contrast of lunar images is low, and few features can be extracted. Therefore, lunar images can be hardly matched with high accuracy. A lunar image matching method based on features from accelerated segment test (FAST) feature and speeded-up robust features (SURFs) descriptor is presented. First, entropy of image is adopted to automatically compute threshold for extracting FAST features. Second, SURF descriptors are used to describe candidate features, and then initial matches with nearest neighborhood strategy are obtained. Third, outliers are rejected from initial matches by RANSAC-based model estimation strategy and homography constraint. Experimental results show that the proposed method can get enough image correspondences and the matching errors are less than 0.2 pixels. It indicates that the proposed method can automatically achieve high-accuracy lunar image matching and lay good foundation for subsequent lunar image stitching and fusion.

Keywords: Image matching · Moon image · FAST · SURF

1 Introduction

With the progress of rocket propulsion, sensor technology, and image acquisition, more and more high-resolution lunar images are captured, which can be used to obtain useful information about the 2D and 3D information about the surface of the moon. Image matching is the premise for getting 2D and 3D data. Image matching is the process to find the corresponding points on different images of the same spatial point, which is the basis for further image processing, such as image mosaic and image fusion.

In recent years, the state-of-the-art image matching methods are feature-based matching methods. Common features used for image matching include corner features, such as Harris, CSS, and SUSAN [1–3], as well as blob features, such as DoG [4, 5] and FH [6–8], and zone features such as MSER [9]. SIFT and SURF are two commonly used feature extraction and description methods. SIFT algorithm includes DoG detection operator and SIFT descriptor, and SURF algorithm includes FH detection operator and SURF descriptor. They can achieve scale, rotation, and illumination invariance and partly viewpoint invariance. Compared to blob features and zone features, corners are easy to detect and fast, which usually appears near the actual corner points. Corner features have strong physical meaning and are still widely used in image

Q. Liang et al. (eds.), *Communications, Signal Processing, and Systems*, Lecture Notes in Electrical Engineering 516,
https://doi.org/10.1007/978-981-13-6504-1_2

matching. Especially in recent years, with the advent of fast corner detector, such as FAST [10, 11], AGAST [12], and BRISK [13], corners play more important role in image matching.

Lunar remote sensing images are different with common images. The distribution of gray levels is compact resulting in low image contrast. Therefore, when carrying image matching with the above features and usual threshold, it can hardly get enough features leading to inaccurate matching precision. One alternative method is to manually adjust the threshold for feature extraction, but this will greatly affect the automatic level of image matching. Therefore, lunar image matching based on FAST features with adaptive threshold is proposed. Contrast thresholds for image matching are automatically computed based on information entropy for matching image pair, respectively. The thresholds are used to extract FAST features. Then a dominate orientation is assigned to each feature, and SURF descriptor is computed along this dominate orientation to achieve rotation invariance. Finally, RANSAC-based homography estimation method is used to reject false matches, which can effectively improve the accuracy of the model estimation.

2 Adaptive Threshold Strategy

2.1 Overview of FAST Algorithm

FAST detects corners according to the self-dissimilarity of the candidate pixels with its surrounding pixels [10]. The FAST algorithm operates by considering a circle of sixteen pixels around the corner candidate p. Point p is classified as a corner if there exists a set of n contiguous pixels in the circle which are all brighter than the intensity of the candidate pixel $I_p + t$, or all darker than $I_p - t$, as illustrated in Fig. 1. Continuous n brighter or darker points mean the dissimilarity with its surroundings of p is low and then it is determined as corners, or reject. There exist different choices of n, such as 9, 10, 11, 12, resulting in different versions of FAST (FAST-9, FAST-10, FAST-11, and FAST-12). FAST-ER is a new version of FAST-n with higher repeatability [11], but low computing efficiency. Comparison studies show FAST-9 is best in different versions of FAST including FAST-ER. FAST-9 is fast, and its

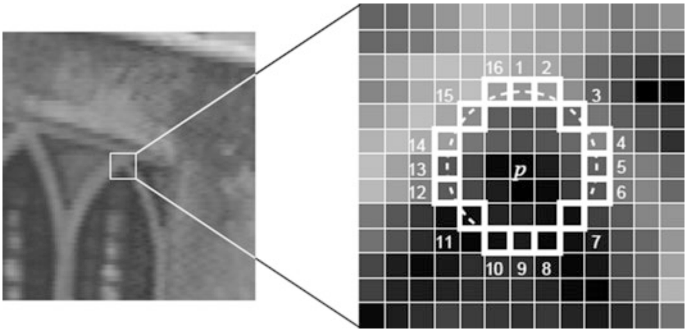
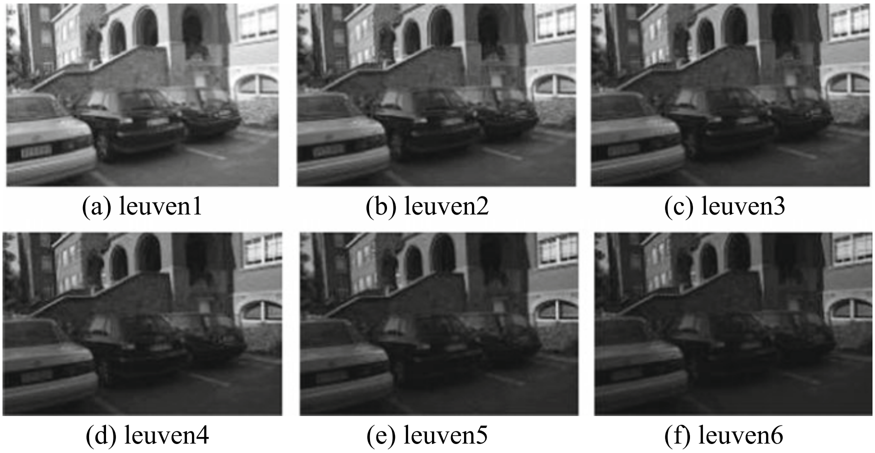


Fig. 1. Illustration of corner detection in an image patch

repeatability is almost the same with FAST-ER, even better in some cases. The computing efficiency of FAST-9 is more than 100% above FAST-ER. Therefore, in the following of the paper FAST-9 is chosen to detect features.

2.2 Analysis of FAST to Images with Different Contrast

As mentioned above, FAST detects corners with a contrast threshold *t*, which affects the adaptability of the FAST to images with different contrast. Literatures [10, 11] recommend that t is chosen between 35 and 40. In fact, the gray scale and contrast of images captured in different illumination conditions vary a lot. There is no uniform threshold t for all images. The Leuven images in Graffi standard test image set are images of the same scene under different illumination conditions, including six images, as shown in Fig. 2, representing different illumination conditions and contrast changes [14].



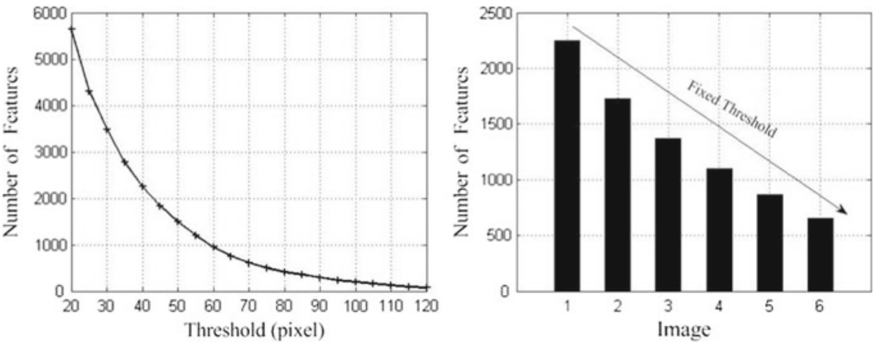
(a) leuven1 (b) leuven2 (c) leuven3

(d) leuven4 (e) leuven5 (f) leuven6

Fig. 2. Different images of the same scene under different illumination conditions

Two experiments are carried out to analyze the effects of threshold *t* and image contrast change on amount of features. First, detect corners to Fig. 2a with varying threshold from 20 pixels to 120 pixels with interval 5 pixels. Second, detect corners with fixed threshold $t = 40$ to all the six images shown in Fig. 2. The resolution of the images used in the experiment is 500 * 900, and the experimental results are shown in Fig. 3.

Figure 3a shows feature detection results with different thresholds on image leuven1. From the experiment results, it is known that the number of features decreases with the increase of the threshold *t*. When $t = 40$, 2249 features are detected and when $t = 40$, only 407 features are detected. Figure 3b shows feature detection results with fixed threshold to images from leuven1 to leuven6. It can be concluded that the number of features detected also decreases with the decreasing of image contrast. For example, when $t = 40$, the number of detected features of the leuven1 is about 3.5 times that of the leuven6. Therefore, the number of feature detected is related to the image contrast



(a) feature detection results to leuven1 (b) results to leuven 1 to leuven 6

Fig. 3. Feature detection results

and threshold *t*. So the threshold *t* should be selected according to the change of image contrast to ensure the algorithm can adapt to images with different contrast.

2.3 Adjusting Contrast Threshold Using Entropy

Different images have different scenes, gray scales, and ambiguity, and therefore, there is no universal threshold suitable for all images. The adaptive threshold-adjusting strategy proposed here is only applicable to lunar images or images similar to lunar images. The remote sensing lunar images have narrow gray scale, and edges and corners of the image are not apparent. Besides, there are obviously bright and dark areas, as shown in Fig. 5. We need to calculate the contrast threshold based on the characteristics of the lunar image. The key to compute the contrast threshold automatically is how to measure the contrast of the lunar image. Image contrast is related to the distribution of image gray scale. Generally speaking, image contrast varies with the distribution of the gray scale. When the distribution of the gray scale is narrow, the image contrast and the entropy of image are low. When the distribution of the gray scale is even, the image contrast and the entropy of image are high. Therefore, the image entropy can be used to describe the contrast of the image. In order to measure the entropy in the range [0, 1], the image entropy is normalized by its possible maximum value, as shown in Eq. (1):

$$norm\_image\_entropy = \frac{-\sum_{i=0}^{255} p_i \log 2(p_i)}{\log 2(L)} \tag{1}$$

where p_i represents the proportion of pixels with gray level equating to *i*, and *i* varies from 0 to 255 with integer value. *L* represents the number of the gray level of the image, and it is 256. log2(*L*) represents the maximum entropy when the gray scale is evenly distributed to all 256 gray levels. The values of normalized entropy of leuven1 are 0.96, 0.91, 0.88, 0.84, 0.80, and 0.76, respectively. It can be inferred that the normalized entropy varies with the image contrast. The following steps are used to determine the adaptive threshold of image based on image entropy.

First, determine the reference threshold t_0. The reference threshold is the basis of the subsequent adaptive threshold calculation. It is the threshold when the image normalized entropy is greater than 0.9. The original FAST algorithm recommends that t_0 should be selected between 35 and 40 empirically.

Second, compute normalized entropy for matching image pair using Eq. (1).

Third, determine the adaptive contrast threshold t_auto based on Eq. (2). The exponent function is used to nonlinearly map the reference threshold t_0 to (0, t_0).

$$t\_auto = \left(2^{norm\_image\_entropy} - 1\right) \times t_0 \tag{2}$$

3 Feature Description and Matching

After feature detection, the classic SURF descriptors are used to describe each feature for matching. SURF descriptors are fast, have invariance to image scale, rotation, illumination, and certain viewpoint changes, and have been widely used in stereo image matching. The detailed construction procedure can be referred in [6]. Feature description and matching are as follows.

First, determine the dominate orientations with the method of SURF. The dominate orientation is the starting direction of the local neighborhood coordinate system for each feature. The dominate orientation is used as a reference direction for matching, and therefore, the rotation invariance can be realized. Feature scale represents the size of local neighborhood for computing feature descriptor. When lunar satellites scan the moon surface, the flight height changes little. So the image scales of the same region have little change, and so feature scale is not considered. Scale factors are set to 1.

Second, build SURF descriptor for each feature centered at the FAST feature, taking the dominate orientation as reference and scale as 1. Theoretically, the SURF descriptors corresponding to the same space point have a higher similarity. This is the basis for false matches rejecting.

Third, KD tree searching strategy is used to get the nearest matching point and the second nearest matching point. When the ratio of nearest matching point and the second nearest matching point is less than the given threshold, the candidate matching is considered to be the correct match. According to the literature [4], the threshold is determined to be 0.7.

Fourth, RANSAC-based strategy and homography constraint are used to eliminate mismatches. The fluctuation of the moon's surface is much smaller than the height of the lunar satellite. Therefore, we can assume that the surface of the moon is planar. The transformation induced by images of the same scene is homography. Homography is a strong constraint, and under the constraint of homography, matching points from different images are mapped one to one, as shown in Eq. (3), where that x_r and x_t are image points belonging to the same spatial point. If more than four pairs of matches are obtained, the homography transformation matrix H can be solved.

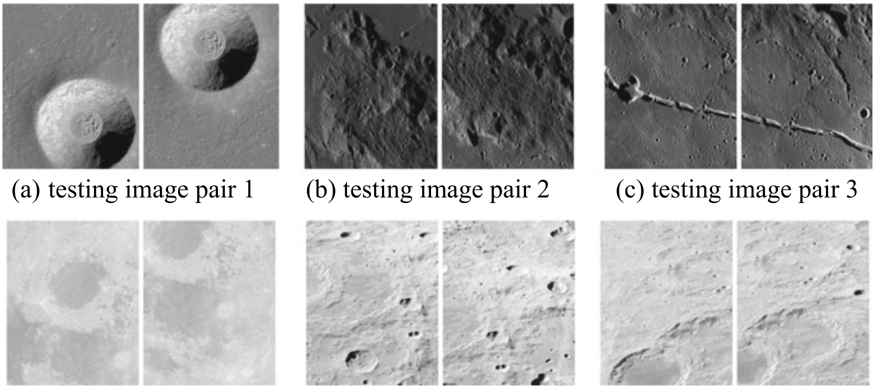
$$\boldsymbol{x}_t = \boldsymbol{H}\boldsymbol{x}_r \tag{3}$$

After the third step, large number of matches can be obtained. However, there are still many mismatches. In order to further reject false matches (outliers), RANSAC-based strategy is used to estimate homography matrix and eliminate the false matches. Two main parameters, random sampling times N and threshold for determining inliers (correct matches) and outliers, are needed. The random sampling times N depends on the ratio of the inliers in initial matching set. After the initial match, the ratio of the inliers is usually higher than 85%. In order to ensure the random selected 4 matches are inliers, at least 10 random sampling is needed. To improve the robustness, the number of random sampling is increased. N is set to 50, and after 50 times of random sampling, the probability of error occurrence is about 10^{-17}. The sum of the mutual Euclidean distance between the transformation point and the matching point is used as the error function as shown in Eq. (4). The threshold value is set to 0.2 pixels. When the error is less than 0.2, the matches are determined as inliers that are correct matches. Or the matches are determined as outliers.

$$\boldsymbol{e} = \|\boldsymbol{x}_t - \boldsymbol{H}\boldsymbol{x}_r\|_2 + \|\boldsymbol{x}_r - \boldsymbol{H}^{-1}\boldsymbol{x}_t\|_2 \tag{4}$$

4 Experiments

Experiments are carried out with actual lunar image. The images used are from the International Planetary Society [15], who collects large dataset of lunar images taken by different nations, including Chinese Chang'e mission. The testing images have certain degree of rotation, illumination and blur changes, and small viewpoint changes. The resolution of Fig. 4a–f is 1000 * 800.


(a) testing image pair 1 (b) testing image pair 2 (c) testing image pair 3

(d) testing image pair 4 (e) testing image pair 5 (f) testing image pair 6

Fig. 4. Testing images

The gray scale of lunar image is narrow. Therefore, the reference threshold t_0 is set to 25, which is lower than the recommended value. The experiment results and the auto-computed threshold are shown in Table 1. From Table 1, it can be seen that the above strategies can automatically calculate the threshold for lunar images with different contrast. There are more than 3000 matches for Fig. 4a, c, and the transformation error is less than 0.1 pixels. There are more than 700 matches for Fig. 4b, e, f, and the matching error is less than 0.15 pixels. There are far less matches for Fig. 4d, but the error after matching is still less than 0.06, which indicates that the accuracy of homography estimation is accurate.

Table 1. Experiment results with adaptive threshold

Figure	Adaptive threshold			Fixed threshold	
	No. of matches	Trans. errors	Adaptive threshold	No. of matches	Trans. errors
4a	3008	0.06	20.4, 20.6	1551	0.04
4b	1018	0.02	18.4, 19.1	281	0.03
4c	3660	0.03	19.4, 20.3	1694	0.02
4d	279	0.06	14.3, 14.0	9	0.08
4e	791	0.14	20.3, 20.3	197	0.13
4f	753	0.14	19.0, 19.2	114	0.12

For comparison, matching experiment with fixed threshold is also carried out here. The threshold is 35, which is the original algorithm recommended. The experiment results of fixed threshold are shown in the right two column of Table 1. It can be concluded that the original FAST algorithm recommends a higher contrast threshold and is not suitable for lunar image matching. The adaptive threshold-adjusting strategy is suitable for lunar image matching. The matching results of Fig. 4b, d, f with lines connecting the matching image pairs are shown in Fig. 5.

5 Conclusion

In order to achieve automatic lunar image matching, a matching method based on FAST features with adaptive threshold is proposed. The proposed method adopts entropy and reference threshold to automatically compute the contrast threshold for extracting FAST features, which can effectively promote the adaptability of FAST features. When matching, SURF descriptors and RANSAC-based homography-estimating strategy are used to improve the accuracy of matching. The experiment results show that the method is effective and suitable for lunar image matching. The automatic level and accuracy of matching are both improved, which lay good foundation for subsequent image mosaic and fusion. The disadvantage of this method is that the adaptability of adaptive threshold-adjusting strategy needs further improved to adapt to a wider variety of images.

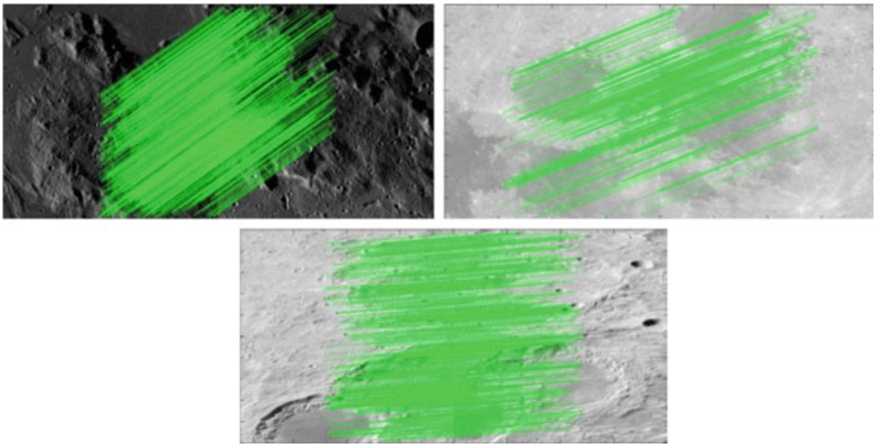


Fig. 5. Part of the image matching result

References

1. Cheng C, Wang X, Li X. UAV image matching based on surf feature and Harris corner algorithm. In: Proceedings of 4th international conference on smart and sustainable city; 2017. p. 1–6.
2. He XC, Yung NHC. Curvature scale space corner detector with adaptive threshold and dynamic region of support. In: Proceedings of international conference on pattern recognition, vol. 2; 2004. p. 791–4.
3. Huiqing Z, Lin G. Image registration research based on SUSAN-SURF algorithm. In: Proceeding of the 26th Chinese control and decision conference; 2014. p. 5292–6.
4. Lowe DG. Distinctive image features from scale-invariant keypoints. Int J Comput Vis. 2004;60(2):9l–1.
5. Ma W, Wen Z, Wu Y, et al. Remote sensing image registration with modified SIFT and enhanced feature matching. IEEE Geosci Remote Sens Lett. 2017;14(1):3–7.
6. Bay H, Ess A, Tuytelaars T, et al. Speeded up robust features. Comput Vis Image Underst. 2008;110(3):346–59.
7. Cheon SH, Eom IK, Moon YH. Fast descriptor extraction method for a SURF-based interest point. Electron Lett. 2016;52(4):274–5.
8. Wei Q, Qiao L, Cheng H, Yaohong X, et al. Strategies of improving matching accuracy about SURF. In: Proceedings of first international conference on electronics instrumentation & information systems; 2017. p. 1–6.
9. Matas J, Chum O, Urban M, et al. Robust wide baseline stereo from maximally stable extremal regions. Image Vis Comput. 2004; 22(10):761–7.
10. Rosten E, Drummond T. Machine learning for high-speed corner detection. In: Proceedings of European conference on computer vision, vol. 1; 2006. p. 430–43.
11. Rosten E, Porter R, Drummond T. Faster and better: a machine learning approach to corner detection. IEEE Trans Pattern Anal Mach Intell. 2010;32(1):105–19.
12. Mair E, Hager GD, Burschka D, et al. Adaptive and generic corner detection based on the accelerated segment test. In: Proceedings of ECCV 2010, volume 6312 of Lecture Notes in Computer Science; 2010. p. 183–96.

13. Stefan L, Chli M, Siegwart RY. BRISK: binary invariant scalable keypoints. In: Proceedings of the IEEE international conference on computer vision; Barcelona, IEEE; 2011. p. 2548–55.
14. Tuytelaars T, Mikolajczyk K. Local invariant feature detectors: a survey. Found Trends Comput Graph Vis. 2007;3(3):177–280.
15. http://www.planetary.org/.