

LETTER

Feature Detection Based on Significancy of Local Features for Image Matching*

TaeWoo KIM^{†a)}, Member

SUMMARY Feature detection and matching procedure require most of processing time in image matching where the time dramatically increases according to the number of feature points. The number of features is needed to be controlled for specific applications because of their processing time. This paper proposes a feature detection method based on significancy of local features. The feature significancy is computed for all pixels and higher significant features are chosen considering spatial distribution. The method contributes to reduce the number of features in order to match two images with maintaining high matching accuracy. It was shown that this approach was faster about two times in average processing time than FAST detector for natural scene images in the experiments.

key words: feature detection, significancy of local feature, feature correspondence, image matching

1. Introduction

For image matching, feature-based methods are more suitable for illuminated change and complicated geometric deformation and have also been widely used because they directly use the salient features that is extracted from two images instead of image intensity values. Feature-based image matching methods consist of four main steps which are feature extraction and feature description, feature correspondence, transformation estimation, and resampling. Feature extraction and feature correspondence are the most important step for accurate image matching. Any problem in the feature extraction will result in incorrect correspondences and incorrect transformation function that will lead to wrong matching results [1], [2].

Popular feature detectors including SIFT [3], SURF [4], FAST [5] detector, etc. have been reported in the literature. It is known that FAST is faster than the other methods in processing time. SIFT [3], BRISK [6], ORB [8], BRIEF [9], etc. are widely used as feature descriptors for the detected feature points. These can describe local features including shape for a local region of an image. Binary descriptors such as BRISK, ORB, and BRIEF have a merit of faster processing time in feature description and matching.

Feature detection and matching process require most of processing time in image matching and the time dramatically increases according to the number of features as shown

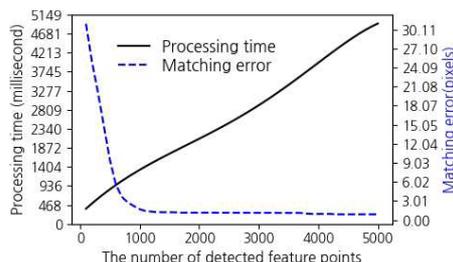


Fig. 1 Processing time of feature detection and matching increases according to the number of feature points. However, its matching error starts to be stable or saturated from near 1,000 feature points.

in Fig. 1. The number of features to be detected is needed to be controlled for specific applications because of their processing time. Therefore, this paper proposes a feature detection method based on significancy of local features which can control and reduce the number of features without sacrificing matching accuracy. Significancy of feature is computed for all pixels and higher significant features are chosen as final feature points considering spatial distribution. The algorithm contributes to reduce the number of features to match two images and can provide faster processing time of feature detection and matching because of considering feature significancy of local features and their distribution.

2. Detection of Significant Features

Feature detection algorithm of this paper consists of two stages of initial and final feature detection stage. FAST is used in the initial feature detection stage where other methods such as SIFT and SURF also can be used. FAST algorithm is to detect a corner using segment test as shown in Fig. 2 (a). A pixel in the image has different weight or significancy in image matching. This paper considers both corner strength of a feature point and a cluster of feature points as a measurement of significancy of a local feature.

2.1 Strength of Corner of a Local Feature

In the initial feature detection stage, the first step is to compute *strength of corner* K_s of a local feature for each pixel in the image. $Y = \{x | |x - x'| \leq 2, x \in X\}$ for $X = \{x | 1..16\}$. $Z = X - Y$ depicts a set of 11 contiguous pixels as shown in Fig. 2 (b) and (c), where I_x is intensity of pixel x and

Manuscript received May 8, 2021.

Manuscript publicized June 3, 2021.

[†]The author is with the Division of Electrical, Electronic and Communication Engineering, Hanyang Cyber University, Seoul, Republic of Korea.

*This work was supported by the research fund of Hanyang Cyber University.

a) E-mail: twkim22@daum.net

DOI: 10.1587/transinf.2021EDL8048

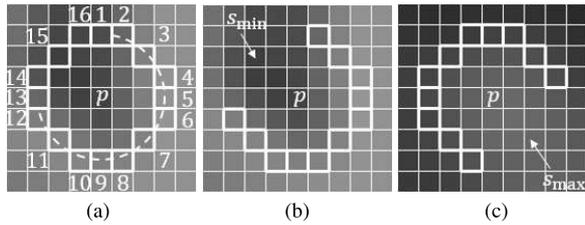


Fig. 2 (a) 11 point segment test for corner detection in an image patch. The highlighted squares are the pixels used in the corner detection. The pixel at p is the center of a candidate corner. The arc is indicated by the dashed line passes through 11 contiguous pixels which are brighter than p by more than the threshold. Candidate corners are (b) for bright surrounding and (c) for dark surroundings where s_{\min} and s_{\max} are minimum and maximum value in 16 surrounding points, respectively.

$$x' = \begin{cases} \operatorname{argmin}_{x \in X} I_x, & \text{(brighter contiguous pixels)} \\ \operatorname{argmax}_{x \in X} I_x, & \text{(darker contiguous pixels)} \end{cases} \quad (1)$$

$$K_s = \min_{z \in Z} |I_p - I_z| \quad (2)$$

Typical example of image composed of K_s values computed for Fig. 3 (a) is shown in Fig. 3 (b).

The second step is to determine threshold value T_{K_s} to achieve the desired number of features to be detected from histogram of the corner strength image without setting a fixed value before applying feature detector as shown in Fig. 3 (c). The initial feature points detected from thresholding have a set of pixels Q in the image as follows:

$$Q = \{p | p \text{ if } K_s > T_{K_s}, p \in P\} \quad (3)$$

2.2 Cluster of Feature Points

Another significance of a local feature is a cluster of feature points. Thresholding with strength of corner of a local feature gives the feature points which are connected together as shown in Fig. 4(a). Bigger clusters are referred to as more significant feature groups.

The features detected by a feature detector such as FAST algorithm are composed of clusters, where pixels within each cluster are spatially connected together as shown in Fig. 4 (a). A set of clusters C are composed of clusters as follows:

$$C = \{C_1, C_2, C_3, \dots, C_N\} \quad (4)$$

where each cluster C_i is composed of connected pixels

$$C_i = \{q | \text{connected pixels}, q \in Q\} \quad (5)$$

2.3 Final Feature Detection

High matching efficiency in the matching process requires the proper number of feature points. Its reason is shown

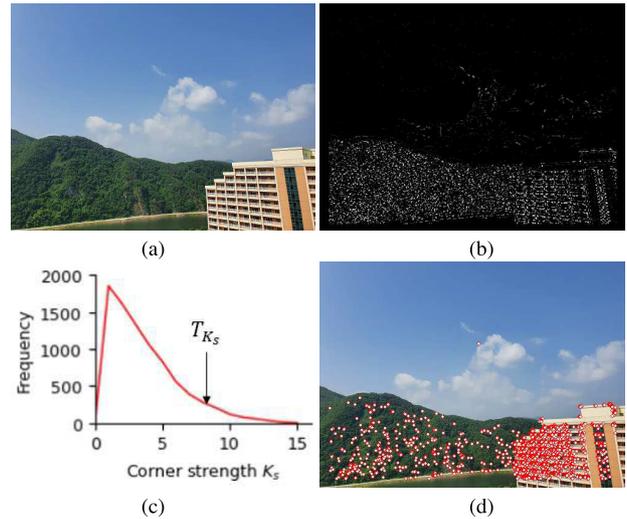


Fig. 3 (a) Typical natural scene image. (b) Corner strength image: brighter pixels have higher corner strengths which are not evenly distributed. (c) Given the desired number of features to be detected, threshold for feature detection is determined from a histogram of (b). (d) For example, 2,000 feature points (red colored circles) were detected from thresholding.

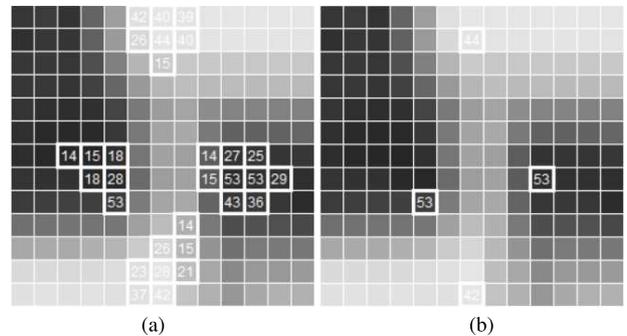


Fig. 4 (a) A portion of Fig. 3 (d) is enlarged for better view where small boxes represent pixels which are connected together, called a cluster. Feature points and their corner strengths are depicted as white thick rectangles and numbers, respectively. (b) A pixel with the largest corner strength within each cluster is chosen.

in Fig. 3 where the corner strength image reveals no even spatial distribution as shown in Fig. 3(b) and (d). And the more the number of feature points increases, the larger the occupied region of detected feature points is in the image.

Each feature within the cluster has different significance or meaningfulness. The features with high significance contribute to give better matching results. Therefore, a feature point with the highest significance in each cluster is chosen as a final feature point as shown in Fig. 4 and Fig. 5. The number of significant feature points is much less than the number of initial feature points as shown in Fig. 4(b), which allows matching time to be reduced. Cluster concept contributes to detect more evenly spatially distributed features in the image. Given the desired number of feature points, T_N , a set of final feature points F are detected as follows:

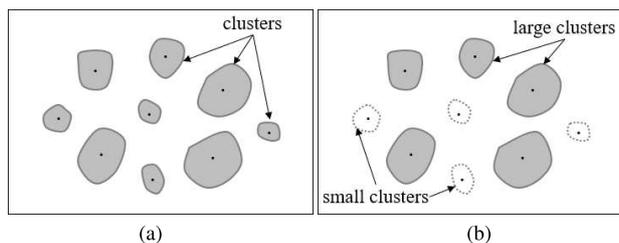


Fig. 5 (a) Clusters of detected feature points have different sizes in the second stage of feature detection. (b) Larger clusters are chosen and a pixel with the biggest strength of corner in each larger cluster is chosen as a final feature point.

$$n(C_1) > n(C_2) > n(C_3) > \dots > n(C_{T_N}) \dots > n(C_N) \quad (6)$$

$$F = \{q | \operatorname{argmax}_{q \in C_i} K_s, 1 \leq i \leq T_N\} \quad (7)$$

Although the number of feature points keep small, the method spreads spatially the feature points and provides better matching results.

3. Experimental Results and Discussion

For experiments, FAST detector and our algorithm for feature detection for natural scene images were implemented using Microsoft Visual C++ language on Windows 10 environment on a laptop computer with Intel i7-9750 CPU @ 2.60GHz. The methods were applied to detect features for natural scene images, 3000x4000 of image resolution, captured by a camera equipped in Samsung Galaxy S10 mobile device. The images were captured as its camera angle was slightly changed for the natural scenes.

In our method, strength of corner was calculated for thresholding image pixels to determine clusters of feature points, in which a point with the largest corner strength in each cluster was chosen as a final feature. The detected features were described by BRISK[6] algorithm and matched with nearest neighbor[6][7] and RANSAC[10] algorithm.

The two techniques were applied to Fig. 7 (e) and resulted in Fig. 8. Figure 8 and Fig. 10 show that our technique provides more spread feature correspondence points than the existing method although the number of the detected feature points are the same number of 400 feature points, respectively, because our method can help to detect better feature points for matching of two images.

To investigate effect of significance of features, FAST algorithm and our method were applied to five typical images for natural scenes as shown in Fig. 7 and their results were summarized in Table 1. N_f is the number of detected feature points satisfying lower bound of consensus or stable state in matching error between two images as shown in Fig. 9. Matching error means measurement of correspondences for detected feature points at the 9 golden reference points between two images as shown in Fig. 6. Table 1



Fig. 6 As a golden reference, this paper uses nine corresponding points (yellow colored crosses) between two images which were manually inputted.

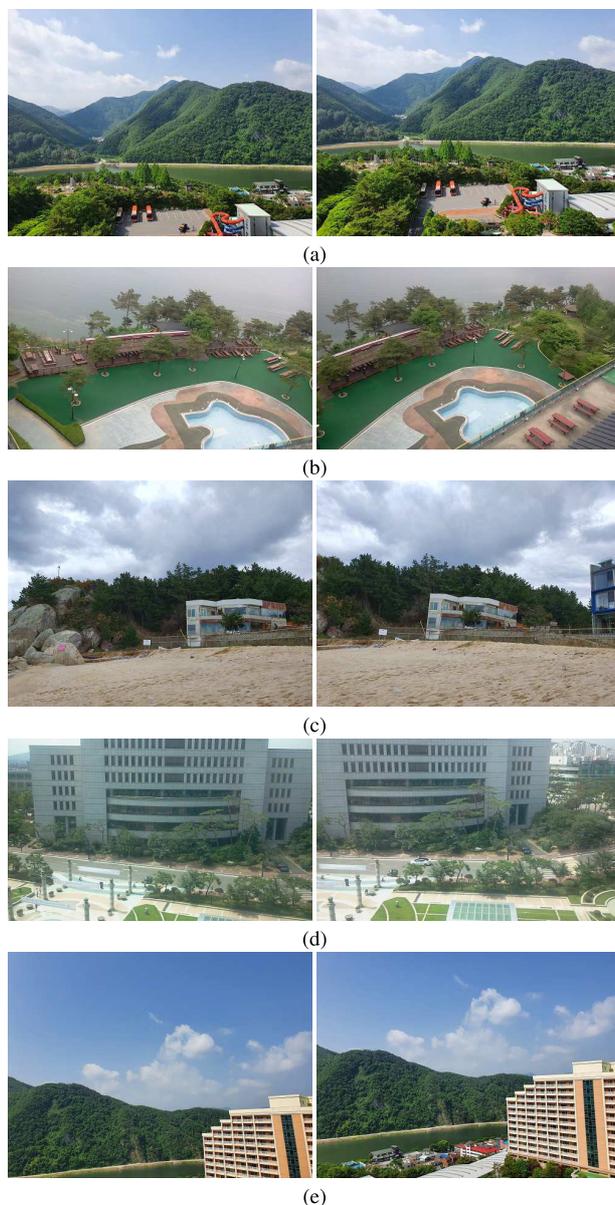


Fig. 7 Typical natural scene image pairs for experiments which have slight different characteristics.

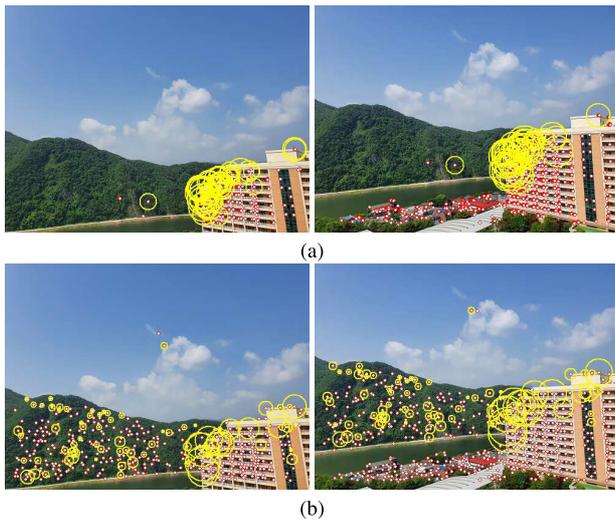


Fig. 8 Comparison of results of feature detection and matching between (a) FAST and (b) our method. Feature correspondences (yellow colored circles) were achieved by matching for 400 feature points (red colored circles) detected by both methods. Our method can detect feature points spatially spread and result in spread corresponding feature points.

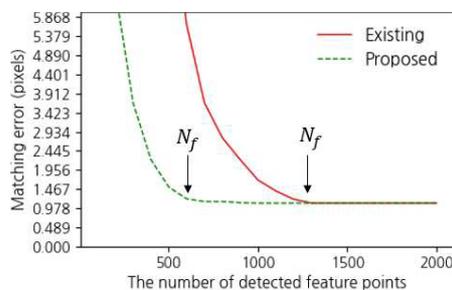


Fig. 9 The number of finally detected feature points, N_f , satisfying lower bound of consensus or stable state in matching error between two images. N_f of the proposed method is smaller than the existing method.



Fig. 10 Resampled images with homographies computed from matching results for two images of Fig. 8: (a) FAST and (b) proposed method for the number of 400 detected feature points.

shows that our technique results in better performance in the number of feature correspondence points and in processing time. Our approach provided faster results of about two times of average processing time than the existing feature detection method for typical natural scene images.

Table 1 Performance comparison for five pairs of natural scene images of Fig. 7, where matching error and processing time were measured as pixels and milliseconds, respectively.

image pair	(a)	(b)	(c)	(d)	(e)	Average
FAST						
N_f	446	895	830	907	1079	831 ± 234
Matching error	1.56	2.56	3.44	2.00	2.78	2.47 ± 0.72
Processing time	942	1517	1434	1527	1727	1429 ± 293
Proposed						
N_f	105	172	140	623	381	284 ± 217
Matching error	1.56	2.44	3.00	1.69	2.11	2.16 ± 0.59
Processing time	476	590	540	1194	855	731 ± 296

4. Conclusion

This paper proposed a feature detection method based on significance of local features for image matching. The algorithm detected initial features through thresholding from corner strengths computed from image. Final features were chosen from clusters of the initially detected features. Our algorithm based on significance of local features contributes to reduce the number of features to match two images and is faster about two times in average processing time than FAST detector for natural scene images.

Acknowledgments

This work was supported by the research fund of Hanyang Cyber University.

References

- [1] C. Leng, H. Zhang, B. Li, G. Cai, Z. Pei, and L. He, "Local Feature Descriptor for Image Matching: A Survey," *IEEE Access*, vol.7, pp.6424–6434, 2019.
- [2] E. Salahat and M. Qasaimeh, "Recent Advances in Features Extraction and Description Algorithms: A Comprehensive Survey," *2017 IEEE Int. Conf. Industrial Technology (ICIT)*, pp.1059–1063, March 2017.
- [3] D.G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol.60, no.2, pp.91–110, 2004.
- [4] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Understand.*, vol.110, no.3, pp.346–359, June 2008.
- [5] E. Rosen and T. Drummond, "Fusing points and lines for high performance tracking," In: *IEEE Int. Conf. Comput. Vis. (ICCV'05)*, vol.2, pp.1508–1511, 2005.
- [6] S. Leutenecker, M. Chli, and R.Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints," *Proc. IEEE Int. Conf. Comput. Vis.*, pp.2548–2555, Nov. 2011.
- [7] M. Muja and D.G. Lowe, "Fast matching of binary features," *Proc. 2012 Ninth Conf. Computer and Robot Vision, CRV '12*, pp.404–410, 2012.
- [8] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," *Proc. IEEE Int. Conf. Comput. Vis.*, pp.2564–2571, Nov. 2011.
- [9] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua, "BRIEF: Computing a local binary descriptor very fast," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.34, no.7, pp.1281–1298, July 2012.
- [10] M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol.24, no.6, pp.381–395, June 1981.