A FAST MATCHING ALGORITHM FOR THE IMAGES WITH LARGE SCALE DISPARITY

SHICHU CHEN

Computer Science Department
Curtin University
Perth, Australia

ZHIQIANG WANG AND YAN REN

Automation College
Shenyang Aerospace University
Liaoning, China

(Communicated by Xiaodong Liu)

ABSTRACT. With the expansion of application areas of unmanned aerial vehicle (UAV) applications, there is a rising demand to realize UAV navigation by means of computer vision. Speeded-Up Robust Features (SURF) is an ideal image matching algorithm to be applied to solve the location for UAV. However, if there is a large scale difference between two images with the same scene taken by UAV and satellite respectively, it is difficult to apply SURF to complete the accurate image matching directly. In this paper, a fast image matching algorithm which can bridge the huge scale gap is proposed. The fast matching algorithm searches an optimal scaling ratio based on the ground distance represented by pixel. Meanwhile, a validity index for validating the performance of matching is given. The experimental results illustrate that the proposed algorithm performs better performance both on speed and accuracy. What’s more, the proposed algorithm can also obtain the correct matching results on the images with rotation. Therefore, the proposed algorithm could be applied to location and navigation for UAV in future.

1. Introduction. Object recognition and matching is a basic problem in the field of computer vision [3]. Given two images with the same scenery but photographed under different circumstances, such as illumination changes and camera variances, the matching needs to find out the correspondences of the same objects or scenes separately existing in the two images. This procedure is also applied in image retrieval, camera calibration and image registration etc.

Image matching is generally divided into three steps [12]. The first step is detection: some regions or structures with special characteristic are selected out such as edges, corners and blobs. Generally, these detecting algorithms, for instance, Hessian-Laplace and Harris-Laplace [6], etc. are called detectors. The next step is description: represent the selected parts with theirs surrounding pixels information. There are also some description algorithms such as LIOP [5] and Brief [13], which

2020 Mathematics Subject Classification. Primary: 58F15, 58F17; Secondary: 53C35.
Key words and phrases. UAV, SURF, matching, location, scale.
* Corresponding author: Shichu Chen.
are normally called descriptors. The final step is matching. There are many matching algorithms and the matching criteria may vary according to different types of descriptors and different purposes. A classic and widely accepted matching method is Mahalanobis or Euclidean distance [11].

In order to realize object recognition and matching, some classic algorithms are invented. D. Lowe used to propose the Scale Invariant Feature Transform (SIFT), which is famous on its accuracy and robustness. However, SIFT is time-consuming, so it cannot be applied in reality directly. Recently S. Katta and S. Pabboju etc., introduced PCA into SIFT which downgrades the dimension of SIFT descriptor in order to accelerate the matching [8, 9]. However, the SIFT detector remains low efficiency. Speeded-Up Robust Feature (SURF) is originally proposed by H Bay and the different versions of SURF also appeared [14], the principle of SURF is similar to SIFT but the former one utilizes the filtering to replace the convolution. Therefore SURF acquires great computation efficiency with sacrificing a little accuracy. Thus SURF gets widely used in reality. In this paper, a fast image matching algorithm is proposed in order to solve the matching between UAV images and tiles of Google maps.

This study contributes to the existing literature in two aspects: an optimal scaling ratio for UAV image is given, in order to perform the best matching between UAV image and Google tiles; the other is that a fast matching algorithm is proposed, which can satisfy the rotation invariance of UAV.

This paper is organized as follows: in Section II, the basic ideas of SURF and matching algorithm will be reviewed. In Section III, the matching difficulties accompanied by larger scale gap will be re-examined and an alternative method which is proposed by Ao [2] will be briefly discussed. Then a faster matching algorithm method is proposed. In Section IV, in order to verify the performance of the new algorithm, experimental results are compared with that of an alternative method.

2. Related work. SURF (Speeded-Up Robust Features) is famous for its speed. Meanwhile it claims to be scale invariance and rotation robust. Generally SURF is a combination of two parts: detector and descriptor. For the detector part, SURF utilizes Hessian determinant to find the place and scale of blob-like structures; for the descriptor part, SURF utilizes Haar wavelet to preserve the neighbour information. This kind feature makes SURF still suitable for significant changes of brightness [20] and it can be easily extended to fit for different demands [1].

2.1. Detection of interest point. The speed of SURF is acquired by substituting convolution operation carried on normal image with filtering operation applied on integral image thus greatly reduces the computation time and only sacrificing a slice of accuracy [16].

2.1.1. Integral image and Hessian matrix. Let $I(X)$ is a normal gray image then its integral image $I_{\Sigma}(X)$ is defined as below. Given a location $X(x,y)$, the pixel value at this point $\hat{X}$ on $I(X)$ equals to the summation of all pixel value which are confined by a rectangle region with the original point on the left-up and this point $\hat{X}$ on $I(X)$ and the formula is as follows 1. So for a given area $\Sigma$, the area value can be easily calculated by three additions $\sum = A - B - C + D$ as shown in Figure 1.

\[ I_{\Sigma}(X) = \sum_{i=1}^{x} \sum_{j=1}^{y} I(x, y) \] (1)
A FAST IMAGE MATCHING ALGORITHM

Figure 1. A region of intensities can be calculated in three additions on integral image.

Hessian matrix is used to find blob-like structures and it can also be used to select the appropriate scale. Given a point $X(x,y)$ on a grey image $I$, the Hessian matrix formula with $\sigma$ scale at $X$ is as follow 2: where $L_{xx}(X, \sigma)$, is the convolution of Gaussian second order derivative $\frac{\partial^2}{\partial x^2}g(\sigma)$ with the image $I$ at point $X$. $L_{xy}(X, \sigma)$ and $L_{yy}(X, \sigma)$ have the similar meanings.

$$H(X, \sigma) = \begin{bmatrix} L_{xx}(X, \sigma) & L_{xy}(X, \sigma) \\ L_{xy}(X, \sigma) & L_{yy}(X, \sigma) \end{bmatrix}$$ (2)

2.1.2. Box filters and approximate determinant of Hessian matrix. Box filters can be applied on integral image and it is the approximation of discretised and cropped the second order of Gaussian partial and mix-partial derivatives, thereby the convolution calculation could be reduced as filtering the integral image with box filters displayed in Figure 2. We denote $D_{xx}$, $D_{xy}$ and $D_{yy}$ as the filtering results which are separately corresponding to $L_{xx}(X, \sigma)$, $L_{xy}(X, \sigma)$ and $L_{yy}(X, \sigma)$ then the Hessian matrix 2 could be simplified to formula 3. The approximate determinant is acquired by 4.

The simplified Hessian matrix is

$$H_{approx} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$ (3)

$$\text{Det}(H_{approx}) = D_{xx}D_{yy} - w_2D_{xy}^2$$ (4)

The initial size of box filters is $9 \times 9$ and the corresponding scale is $\sigma = 1.2$. In order to obey energy conservation and balance the approximation error, there is a weight $w$ in formula 4 deriving from formula 5. Furthermore, the filter responses need to be normalised to balance different sizes of box filters.

$$w = \frac{|L_{xy}(1.2)|_F |D_{yy}(9)|_F}{|L_{yy}(1.2)|_F |D_{xy}(9)|_F}$$ (5)
After above computations on integral image $I_X(X)$, at any point $X(x, y)$ and at a given scale $\sigma$ we can figure out the approximation determination of point $X(x, y)$. By increasing the template size of box filters which is equivalent to increasing the scale $\sigma$ as shown in Figure 3, for the same integral image $I_X(X)$ we can get different Hessian determination at point $X(x, y)$ on a higher scale.

![Figure 3. Filters $D_{yy}$ (above) and $D_{xy}$ (below) with two size: 9 $\times$ 9 templates (left) and 15 $\times$ 15 templates (right)](image)

**2.1.3. Scale pyramid and interpolation.** According to filter template size, normally SURF groups these filters into three octaves and each octave contains four layers. The permutation is shown in Figure 4. The gradually increased filters are an analogue to increasing scale and downgrading the image as shown in Figure 5 [10]. The first octave is composed with 9 $\times$ 9, 15 $\times$ 15, 21 $\times$ 21 and 27 $\times$ 27 filters and the corresponding scale $\sigma$ is 1.2, 2.0, 2.8 and 3.6. The second octave starts from 15 $\times$ 15 and the third starts from 27 $\times$ 27 which guarantees the continuity of scale change and accuracy of SURF.

The determinant achieves its extreme – maximum or minimum - when the filter’s scale is comparable to the object’s scale, and the centre of filter is near to the interest point. As Figure 6 shown, we need to search for the extreme points in a 3 $\times$ 3 $\times$ 3 neighbourhood in the scale pyramid then use non-maximum suppression [7] to select out these points whose determinants are extreme among its surrounding points as interest points then we use interpolation to get the accurate location and scale space information [17].

![Figure 4. Filters and octaves permutation](image)
2.2. **Description of interest points.** The rotation invariance of SURF is acquired by assigning a reproducible dominant orientation for each interest point and the scale invariance is achieved by uniting computation unit to $s$ – the scale of current interest point.

2.2.1. **Dominant orientation.** Firstly, we compute Haar wavelet response in horizontal and vertical directions within a circular region. Take the interest point as the circle centre and select $6s$ as the radius. The sampling step is $s$. The side length of Haar wavelet template is $4s$ as shown in Figure 7. Secondly, all these responses are weighted by Gaussian $\sigma = 2s$ centred at the interest point. A sector window with angle of $\pi/3$ is gradually rotated around the interest point and within each sliding sector we sum up all the horizontal and vertical responses to get a local orientation vector and this procedure can be seen in Figure 8. Finally we choose the longest vector among all local vectors as the dominant orientation of current interest point.
2.2.2. Descriptor and normalization. Construct a rectangular with length $20s$, centred at the interest point and rotated to align with the dominant orientation. This rectangular is then divided into $4 \times 4$ sub-regions as shown in Figure 9. The sampling step is still $s$ but the Haar wavelet template side length in the descriptor is chosen to $2s$. Again we calculate the horizontal and vertical responses (here the horizontal and vertical means align with the dominant orientation) and the responses are then weighted with Gaussian $\sigma = 3.3s$ centred at the interest point. In each sub-region there is a four dimensional descriptor vector $v = (\sum dx, \sum dy, \sum |dx|, \sum |dy|)$. The summation of the horizontal responses is represented as $\sum dx$, the summation of the vertical responses is represented as $\sum dy$, the summation of the horizontal and vertical absolute responses is separately represented as $\sum |dx|$ and $\sum |dy|$ and Figure 10 displays that these responses represent different image local characteristic. Therefore $4 \times 4$ sub-regions and each sub-region contains four parameters totally $4 \times 4 \times 4 = 64$ dimensional vector is generated. Finally this vector is divided by its length to get unit descriptor vector for interest point.
2.3. Image matching algorithm based on SURF. SURF is a novel scale and rotation invariant detector and descriptor for images. It doesn’t contain the matching algorithm. Different matching algorithms can be combined with different detection and description algorithms under specific demands. For example, matching based on similarity: we can calculate the distance among description vector such as SSD (Sum of squared differences, formula 6) which provides more precise matching result and SAD (Sum of absolute differences, formula 7) which principle is like SSD, but provides faster matching result.

For two equations:

\[
SSD = \sqrt{\sum_{s=1}^{m} \sum_{t=1}^{n} (S(i + s - 1, j + t - 1) - T(s, t))^2}
\]  

\[
SAD = \sum_{s=1}^{m} \sum_{t=1}^{n} |S(i + s - 1, j + t - 1) - T(s, t)|
\]

\(S(i, j)\) and \(T(i, j)\) represent interest points separately located on two images; \(M\) and \(N\) are the components of descriptor vector. Matching based on speed: if the size of matching database is huge, you may prefer “Approximate” mode rather than “Exhaustive” mode to do matching [4]. In this paper the SAD is preferred and considering the amount of interest points is small the exhaustive mode is selected [2].

3. Fast matching algorithm for images with large scale disparity. Commonly, the flying height of UAV is a few hundred meters, but the track height of the satellite is much higher from a few kilometres to thousands of kilometres. This makes huge difference between the UAV aerial images and the satellite maps. Therefore, without any image pre-processing, it is not realistic to match UAV aerial images and the satellite maps by using SURF [15, 18]. Ao used to propose a method which traverses a range of ratios to search the optimal scaling ratio [2]. In his algorithm, an initial value scale \(\alpha_0\) is given which starts with the division of UAV flying height 500 meters to satellite vision height 9600 meters, i.e., \(\alpha_0 = \frac{H_{UAV}}{H_{satellite}} = \frac{500}{9600} \approx 0.05\). And the traversal range of \(\alpha_n\) is from \(\alpha_0\) to 0.2 with the step 0.01. In the process of gradually zooming out UAV image, it is equivalent to technically elevating the height of the UAV up to the height of satellite, to find the most suitable scaling ratio. Ao’s method is time consuming, which is the main disadvantage of that algorithm. In this paper, a fast matching algorithm for the images with large scale disparity is proposed. In the following, the detail algorithm will be given. Firstly, due to the large different of scales between UAV image and Google tiles, UAV image with large scale should be reduced in order to match with Google tiles [19]. Thus, firstly a suitable scaling ratio is defined as follows 8 to reduce the scales of UAV images,

\[
\alpha_{best} = \frac{D_{UAV}}{D_{Tile}} \times C
\]

where \(D_{UAV}\) is the ground distance represented by each pixel on UAV image, and \(D_{Tile}\) is the ground distance represented by each pixel on tile or satellite map. \(C\) is a coefficient used for improving the quality of tile and increasing the number of interest points detected. In this paper we double the size of tile, so we let \(C\) be 2. If the flight altitude of UAV is relatively stable, \(\alpha_{best}\) can be seen as a constant within a flying area. Now, the new UAV image \(I_{scaled}\) can be obtained by reducing the
dimensionality of $I_{UAV}$ with $\alpha_{\text{best}}$. Secondly, $I_{\text{scaled}}$ and each tile in satellite map will be matching by SURF algorithm. Meanwhile, in order to validate the results between the UAV image and tiles, a validity index $V_i$ will be defined as follows,

$$V_i = \frac{N_i}{AD_i} \quad (9)$$

where $i$ refers to the serial number of tiles, $N_i$ refers to the number of matched pairs and $AD_i$ is the total average matching distances for matched pairs on each tile. Table 1 and Figure 11 show the details of this proposed algorithm.

Notably, the proposed fast matching algorithm is suitable for two sceneries. One is the scenery in which UAV image is parallel to the corresponding ones of the Google satellite map or tile, and another scenery is UAV image are rotated a certain angle compared to the corresponding Google map or tile.

Table 1. Pseudo-code of fast matching algorithm

**Algorithm**: Fast matching algorithm for images with large scale disparity

**Input**: UAV aerial image $I_{UAV}$, satellite tiles $Tile_i$, $i = 1, 2, ..., n$ and $C = 2$.

**Output**: Best matching tile $Tile_b$ with $I_{\text{scaled}}$.

1: $\alpha_{\text{best}} = \frac{D_{UAV}}{D_{Tile}} \times C$;
2: Reduce $I_{UAV}$ with $\alpha_{\text{best}}$ to get $I_{\text{scaled}}$;
3: Let $Value_i$ represent the corresponding matching performance between $I_{\text{scaled}}$ and $Tile_i$;
4: for $i := 1$ to $n$ do
5: Double the size of $Tile_i$;
6: Do the matching between the doubled $Tile_i$ and $I_{\text{scaled}}$;
7: Matching performance is valued by $Value_i$;
8: end for
9: $b = \arg \max_i Value_i$ and $Tile_b$ is the best matching tile with $I_{\text{scaled}}$.

Stop.

**Figure 11.** Flowchart of fast matching algorithm

4. **Experiment result and analysis.** In this paper, UAV aerial images are captured by UAVs at about 500 meters high and the satellite tiles are $15^{th}$ level which is downloaded from Google with vision height 9600 meters (https://developers.
A FAST IMAGE MATCHING ALGORITHM

Google.com/maps/documentation/javascript/maptypes). Each pixel in UAV image captured at 500 meters represents about 0.21 meters. And the pixel in 15-level Google map represents about 3.51 meters. Here let $C$ be 2, so the best scaling ratio $\alpha_{best}$ should be \((0.21/3.51) \times 2 = 0.1196\). In order to verify the effectiveness of the fast matching method, we design the following two experiments. **Experiment 1:** The scenery is that UAV image is parallel to the corresponding Google map or tile. This condition is also verified by Ao’s method [2]: it is expected to see that the correct tile could be selected, i.e., the correct scenery is located. Meanwhile, the computation time could be reduced compared with Ao’s algorithm. **Experiment 2:** The scenery is that UAV images are rotated with a certain angle corresponding to the Google map or tile. In order to validate the efficacy of the proposed method, the UAV image will be rotated with different angles. And this is the common condition because the UAVs are always affected by airflow. Their angle or direction cannot be strictly parallel to the ground. While this condition has not been considered in Ao’s paper.

4.1. **Results of experiment 1.** It can be seen clearly that four UAV images A, B, C and D are matched with the correct tiles in Google satellite map respectively as shown in Figure 12-15. Figure 12 gives the matching result for UAV image A and its corresponding tile via Ao’s method and the proposed method. It can be seen that there are 3 matched pairs of interest points on (a) and (b) by using Ao’s method; and there are 7 matched pairs of interest points on (c) and (d) by the proposed fast algorithm. Similarly, Figure 13-15 are the results of UAV image B, C, D. Furthermore, the detailed comparisons between Ao’s method and the proposed fast matching method are illustrated in Table 2.

![Figure 12](image_url)

**Figure 12.** (b) and (d) is the UAV image A; (a) and (c) are the matched tiles with Image A via Ao’s method and the proposed method respectively.

In Table 2, the average matching time for each UAV image is 23.4 seconds by the proposed fast matching method compared with the average 722.3 seconds consumed by Ao’s method. The proposed method is approximate thirty times faster compared with Ao’s method, so this is one improvement of this paper. Moreover, the overall numbers of the matched pairs via the proposed fast method is apparently superior, especially on image A and image B. As shown in Figure 12 and Figure 13, the numbers of the matched pairs are doubled. And these matched pairs are more...
Figure 13. (b) and (d) is the UAV image B; (a) and (c) are the matched tiles with Image B via Ao’ method and the proposed method respectively.

Figure 14. (b) and (d) is the UAV image C; (a) and (c) are the matched tiles with Image A via Ao’ method and the proposed method respectively.

Table 2. Comparisons on time-consuming and numbers of matched pairs

| Image No.       | Matching time (second) | Numbers of matched pairs |
|-----------------|------------------------|--------------------------|
| Image A using fast method | 23.1                   | 7                        |
| Image A using Ao’s method    | 738.1                  | 3                        |
| Image B using fast method   | 23.0                   | 6                        |
| Image B using Ao’s method   | 703.7                  | 3                        |
| Image C using fast method   | 24.1                   | 6                        |
| Image C using Ao’s method   | 749.9                  | 7                        |
| Image D using fast method   | 23.3                   | 7                        |
| Image D using Ao’s method   | 697.5                  | 9                        |
centralized. Therefore, the proposed fast algorithm not only maintains the accuracy of matching, but also accelerates the matching speed greatly. It is expected to be applied in the practical application.

4.2. Results of experiment 2. In this experiment, the capacity of image matching on the images with the same scene and different directions will be identified. And the first work is to verify whether the correct matched tile will be selected by using the proposed algorithm when the given UAV image is not parallel to the tile. Another work is to estimate the direction of the given UAV image according to Google satellite map.

Given two points \( A(a_1, a_2) \) and \( B(b_1, b_2) \) on an image, the straight line across \( A \) and \( B \) and x-axis of the image form an angle as shown in Figure 16. The angle will change along with the image rotation and it is easy to be calculated by the arctangent value as formula 10. Because the Google satellite map is of fix orientation: the x axis is parallel to the latitude and the y axis is parallel to the longitude, the map could be used as base orientation to estimate the direction of UAV. For example: if we have two matched pairs \( A(a_1, a_2) \) and \( B(b_1, b_2) \) on UAV aerial image, and the \( \text{Angle} \) is 60 degrees, in the meantime the \( \text{Angle}' \) of the corresponding matched pairs \( A'(a'_1, a'_2), B'(b'_1, b'_2) \) on the best Google tile is 20 degrees, then the difference is the direction deviation of UAV, that is 40 degrees.

\[
\text{Angle} = \arctan \left( \frac{b_2 - a_2}{b_1 - a_1} \right) \tag{10}
\]

Apparently there are at least two couples of matched pairs for calculating the direction of UAV image. Furthermore, a more accurate direction could be computed by calculating the average of these directions with more than two couples of matched pairs. For example, we randomly select out two pairs out of total pairs and calculate one direction. Then the procedure is repeated for several times. Finally an averaged direction will be obtained. In the following experiment, three random groups of matched pairs are used to calculate the direction angle. The result is as shown in the third column of Table 3.

The second column is a serial of real rotation angles in degree. The last column is the corresponding calculated angle. It can be seen that the accuracy of calculated
Figure 16. Diagram of the angle between $\overrightarrow{AB}$ and Google map direction

| Image No. | Real image direction (degree) | Calculated image direction (degree) |
|-----------|-------------------------------|-------------------------------------|
| Image B   |                               |                                     |
| 15        | 16.44                         |                                     |
| 30        | 29.57                         |                                     |
| 45        | 49.10                         |                                     |
| 60        | 59.82                         |                                     |
| 75        | 76.25                         |                                     |
| 90        | 93.21                         |                                     |
| 110       | 109.07                        |                                     |
| 120       | 117.15                        |                                     |
| Image C   |                               |                                     |
| 130       | 130.09                        |                                     |
| 140       | 138.96                        |                                     |
| 150       | 149.41                        |                                     |
| 160       | 159.54                        |                                     |

rotation angle or direction is competitive close to the real image rotation angle: the average deviation in image B rotation group is 1.768 degree and the average deviation in image C rotation group is 0.993 degree. Then the matching results are obtained as shown in Figure 17 and Figure 18. Image B is rotated under a serial angles from 15 to 90 degrees in anti-clockwise direction with 15 degrees step. Figure 17 only displays the matching results of Image B with 15, 45 and 75 degrees rotations. In Figure 17, (a), (c) and (e) are the best matching tiles for (b), (d) and (f), respectively. Top group ((a) and (b)) displays the matching result of 15 rotation degrees, the middle group ((c) and (d)) is 45 rotation degrees, and the bottom group is 75 rotation degrees. Similarly, Image C is rotated from 110 to 160 degrees with 10 degrees step in anti-clockwise. Figure 18 shows the matching result of Image C with 120, 140 and 160 degrees rotations. It is clearly seen that the correct tile is selected and matched with any angle rotations. These matched groups illustrate that the fast matching method is invariant to rotation. It is expected to be applied in UAV autonomous location and navigation.
Figure 17. The matching results for Image B with rotations

Figure 18. The matching results for Image C with rotations
5. Conclusions. This paper proposes a fast matching algorithm, which can be applied on the images with huge scale difference. The proposed method first introduces a scaling factor to reduce the scale of the image, which can increase the matching accuracy. Furthermore, an angle evaluation index is presented, in order to validate the matching performance. The results show that the fast matching algorithm not only obtains the matching accuracy and the invariance to rotation, but also accelerates the matching speed greatly. And it is expected to be applied in UAV autonomous navigation.

Acknowledgments. We appreciate the efforts of two reviewers and Associate Editor for their critical comments on this manuscript and make the quality of this paper improved significantly. This work is supported by Aviation Science Foundation with No. 2017ZC54007 and 2017ZA54006.

REFERENCES

[1] W. An, R. Yu and Y. Wu, An image registration algorithm based on FAST, RANSAC and SURF, Information Technology and Applications, (2015), 329–333.
[2] T. C. Ao, X. D. Liu, Y. Ren, R. Luo and J. Xi, An approach to scene matching algorithm for UAV autonomous navigation, Chinese Control and Decision Conference, (2018), 996–1001.
[3] F. Bulbul, F. Badsha and R. Islam, Object detection by point feature matching using Matlab, Advances in Image and Video Processing, 6 (2018).
[4] M. B. Cohen, B. T. Fasy, G. L. Miller, A. Nayyeri, D. R. Sheehy and A. Velingker, Approximating nearest neighbor distances, Algorithms and Data Structures, Lecture Notes in Comput. Sci., Springer, Cham, 9214 (2015), 200–211.
[5] R. Dong, M. Lin and F. Li, Multiplayer convolutional feature aggregation algorithm for image retrieval, Mathematical Problems in Engineering, 2019 (2019), 1–12.
[6] A. S. Eltanany, M. Sady and A. S. Amein, Key point detection techniques, Advances in Intelligent Systems and Computing, 1058 (2019).
[7] H. Jan, B. Rodrigo and S. Bernt, Learning Non-maximum Suppression, Conference on Computer Vision and Pattern Recognition, (2017), 6469–6477.
[8] S. Katta, S. Pabboju, A. V. Baba and R. Akhil, CBIR using SIFT with LoG, DoG and PCA, Data Engineering and Communication Technology, (2020), 623–637.
[9] X. Li, J. Luo, C. Duan and P. Yin, Portrait matching based on sift features, IOP Conference Series: Materials Science and Engineering, 612 (2019), 032162.
[10] X. Li, X. Wang and C. Cheng, Application of scene recognition technology based on faster and surf algorithm in augmented reality, International Conference on Smart and Sustainable City, (2017), 23.
[11] L. Liberti and C. Lavor, Euclidean Distance Geometry: An introduction, Springer Undergraduate Texts in Mathematics and Technology, Springer, Cham, 2017.
[12] X. Liu, X. Tao and N. Ge, Fast remote-sensing image registration using priori information and robust feature extraction, Tsinghua Science and Technology, 21 (2016), 552–560.
[13] Y. L. Liu, H. Zhang, H. L. Guo and N. N. Xiong, A FAST-BRISK feature detector with depth information, Sensors, 18 (2018), 3908.
[14] L. Tong and X. Ying, 3D point cloud initial registration using surface curvature and SURF matching, 3D Research, 9 (2018).
[15] C. Tsai and Y. Lin, An accelerated image matching technique for UAV orthoimage registration, ISPRS Journal of Photogrammetry and Remote Sensing, 128 (2017), 130–145.
[16] C. Y. Wang, Z. Zhang, Q. W. Li and X. Zhou, An image copy-move forgery detection method based on SURF and PCET, Institute of Electrical and Electronics Engineers, 7 (2019), 170032–170047.
[17] D. Xiang, B. Zhong and K. Ma, A location-aware scale-space method for salient object detection, International Conference on Image Processing, (2015), 4195–4199.
[18] K. Yang, A. N. Pan, Y. Yang, S. Zhang, S. H. Ong and H. L. Tang, Remote sensing image registration using multiple image features, Remote Sensing, 9 (2017), 581.
[19] D. Zeng, L. D. Wu, B. Y. Chen and W. Shen, Slope-restricted multi-scale feature matching for geostationary satellite remote sensing images, Remote Sensing, 9 (2017), 576.
A FAST IMAGE MATCHING ALGORITHM

[20] L. Zhu, Y. Wang, B. Zhao and X. Z. Zhang, A fast image stitching algorithm based on improved SURF, International Conference on Computational Intelligence and Security, (2015), 171–175.

Received October 2019; revised March 2020.

E-mail address: shichuc@gmail.com
E-mail address: 1767743258@qq.com
E-mail address: renyan1108@hotmail.com