Journal of Systems Engineering and Electronics
Vol. 31, No. 6, December 2020, pp.1128 – 1139

A fast, accurate and dense feature matching algorithm for aerial images

LI Ying, GONG Guanghong*, and SUN Lin

School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China

Abstract: Three-dimensional (3D) reconstruction based on aerial images has broad prospects, and feature matching is an important step of it. However, for high-resolution aerial images, there are usually problems such as long time, mismatching and sparse feature pairs using traditional algorithms. Therefore, an algorithm is proposed to realize fast, accurate and dense feature matching. The algorithm consists of four steps. Firstly, we achieve a balance between the feature matching time and the number of matching pairs by appropriately reducing the image resolution. Secondly, to realize further screening of the mismatches, a feature screening algorithm based on similarity judgment or local optimization is proposed. Thirdly, to make the algorithm more widely applicable, we combine the results of different algorithms to get dense results. Finally, all matching feature pairs in the low-resolution images are restored to the original images. Comparisons between the original algorithms and our algorithm show that the proposed algorithm can effectively reduce the matching time, screen out the mismatches, and improve the number of matches.

Keywords: feature matching, feature screening, feature fusion, aerial image, three-dimensional (3D) reconstruction.

DOI: 10.23919/JSEE.2020.000085

1. Introduction

Three-dimensional (3D) reconstruction based on aerial images is a key research problem in photogrammetry and remote sensing, computer vision, and other fields. Pix4D [1] and ContextCapture3D [2] are common aerial photogrammetry softwares for 3D reconstruction problems.

The research on hardware and software of 3D reconstruction is a common concern. Nowadays, with the popularity of 3D reconstruction applications, more and more people want to realize 3D reconstruction in low-cost devices, such as laptop computers and tablet computers. These devices are usually equipped with low-level configuration, but they are easy to carry around and can provide services anytime and anywhere. In terms of software, by optimizing and improving algorithms, people keep studying fast and accurate methods to achieve real 3D reconstruction.

Taking the generation of dense 3D point clouds as an example, feature matching is an important step of 3D reconstruction. Feature number, matching time and the accuracy of feature pairs can directly affect the next steps such as the calculation of aerial triangles and the generation of sparse point clouds.

Moreover, aerial images usually have the characteristics of high resolution, large coverage areas, too many kinds of objects and repeated textures. They will lead to long matching time, uneven distribution of matching pairs and confusion of adjacent feature matching. Considering that the algorithms based on deep learning are usually limited by hardware and datasets, the existing 3D reconstruction methods usually use the traditional feature matching algorithms to directly perform feature matching and aerial triangulation calculation, resulting in too long time, few feature points or even reconstruction failures.

With the increasing requirements of time, accuracy and large-scale scenes, it is particularly important to find a fast, accurate and dense feature matching algorithm for aerial images and low-cost devices. Therefore, an algorithm is proposed in this paper. The main contributions of this work are as follows: (i) The algorithm can realize fast, accurate and dense feature matching of aerial images on low-cost devices. (ii) By finding the appropriate resolution of aerial images, feature matching can be realized quickly on low-cost devices, while ensuring the number and distribution of feature pairs. Moreover, the matching point pairs are restored to the original images by an effective algorithm in the end. (iii) Considering that aerial images are prone to mismatching due to their large coverage areas, a region-based feature screening algorithm is proposed to further realize feature screening.

Manuscript received July 18, 2019.

*Corresponding author.

This work was supported by the Equipment Pre-Research Foundation of China (6140001020310).

(iv) To ensure the quality of 3D reconstruction, it is necessary to further improve the number and distribution uniformity of feature pairs, and based on this, a feature fusion algorithm is proposed.

2. Related works

Feature matching algorithms usually include feature pairing and feature screening, in which feature pairing includes feature detection, feature description and feature matching. Local feature point detection and description are the most important problems. Lowe [3] proposed the scale-invariant feature transform (SIFT) algorithm, which has scale invariance but the main direction of features may not be accurate enough. Bay et al. [4] proposed the speeded up robust features (SURF) algorithm, whose variable is the size and the scale of Gaussian blur templates, so as to avoid the process of down-sampling and improve the processing speed. Using the features from accelerated segment test (FAST) [5] algorithm, the oriented FAST and rotated binary robust independent elementary features (BRIEF) (ORB) algorithm [6] extracts features, and obtains the main direction of features with intensity centroid. The ORB algorithm has rotation invariance, but does not have scale invariance. In addition, the binary robust invariant scalable keypoints (BRISK) algorithm [7], the fast retina keypoint (FREAK) algorithm [8], the KAZE (a Japanese word meaning wind) algorithm [9] and the accelerated-KAZE (AKAZE) algorithm [10] all achieve good results, but they also have some limitations. Tareen et al. [11] presented a comprehensive comparison of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK algorithms, and conducted experiments on diverse images taken from benchmark datasets.

Considering the limitations of traditional image matching algorithms, related improved algorithms have been proposed. Some researchers chose to improve a single algorithm [12–14]. For example, many researchers improved the SIFT algorithm [15]. Considering that the SIFT algorithm was sensitive to nonlinear radiation distortions, a radiation-variation insensitive feature transform (RIFT) was proposed [16]. The color information [17] and the scale-orientation joint restriction criteria [18] were also used to achieve robust feature matching.

Some researchers chose to combine multiple algorithms to improve their results. Ma et al. [19] used the FAST algorithm and the ORB algorithm to realize an improved oriented feature description. Combined with the decision tree theory, Wu [20] proposed a feature detection method based on image grayscale information-FAST operators and then used the BRIEF feature description method to describe the points. Aiming at the fact that the ORB descriptors do not have scale invariance, an improved feature point matching algorithm borrowing the idea of BRISK was proposed [21].

Considering different application situations, researchers put forward many new methods. In order to accommodate repetitive texture and unknown distortion, Li et al. [22] proposed a novel region descriptor formed by four feature points to improve the matching accuracy. Zhu et al. [23] combined the second-order characteristics of points with the Hessian matrix to detect more feature points.

However, the above algorithms are usually suitable for ordinary low-resolution images. Considering the time and texture characteristics of high-resolution images, further research is needed [24]. Xi et al. [25] used several different point feature extraction operators to extract features from the aerial images of different scenes and analyzed their performance and adaptability. Based on a competency criterion, and scale and location distribution constraints, Amin et al. [26] proposed a novel method to extract uniform and robust local feature for remote sensing images. Moreover, researchers adopted different algorithms for different applicable images, such as coastal remote sensing images [27], low contrast or homogeneous textures images [28, 29], and the aerial images with high-resolution and rich edges [30]. For aerial images, as one of the most important matching algorithm evaluation indices, time cost is also worth considering. To reduce computational complexity, Song et al. [31] achieved faster matching through an iterative transform simulation.

In addition, in order to improve accuracy, feature screening is usually carried out. There are three traditional screening algorithms: the nearest-next-neighbor ratio (the ratio algorithm), the crosscheck algorithm, and the random sample consensus (RANSAC) method [32]. Among them, the RANSAC algorithm sets objective functions according to specific problems and actual conditions, and distinguishes all feature points between inlier points and outlier points. The accuracy of RANSAC is high, but the number of retained feature points is not stable. There are also many attempts on improving these algorithms to remove mismatches. Based on the SURF bidirectional matching method, Gui et al. [33] presented a point-pattern matching method using the shape context. Xi et al. [34] improved the RANSAC algorithm by adding image gray level information. Li et al. [35] proposed a mismatch-removal algorithm called locality affine-invariant matching. Considering the viewpoint differences, Song et al. [31] proposed a homography matrix evaluation method based on a geometric approach. According to the scale-invariant feature transform matching algorithm, Gao et al. [36] proposed and compared several improved false matches elimination algorithms.

The evaluation of the feature matching algorithm is an important index to judge the advantages and disadvantages of the algorithm. Xi et al. [34] used the root mean square error (RMSE) to evaluate the quality of image matching. Li et al. [16] used the number of correct matches (NCM), RMSE, mean error (ME), and success rate (SR) as the evaluation metrics. Song et al. [31] used the NCM to evaluate their algorithm as well. Gesto-Diaz et al. [37] evaluated 28 different combinations of detectors and descriptors, and assessed the matching results based on the receiving operating characteristic curve associated to all combinations.

In conclusion, it is a key problem to obtain fast, accurate and dense feature pairs through algorithm improvement for high-resolution aerial images. In this work, we propose an algorithm with four steps, which will be described in detail in the following sections.

3. Algorithm

The traditional steps of matching features mainly include feature pairing and feature screening. In this paper, we implement a fast, accurate and dense feature matching algorithm which includes four steps: image down-sampling matching, region-based feature screening, fusion of feature pairs using different matching algorithms, and restoration of matching pairs to the original image location. The flow chart is shown in Fig. 1.

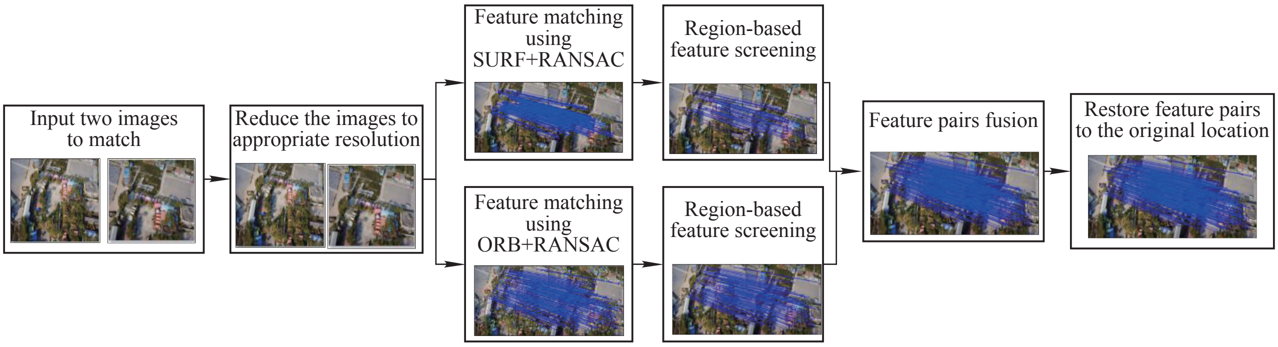

Input two images to match
Reduce the images to appropriate resolution
Feature matching using SURF+RANSAC
Feature matching using ORB+RANSAC
Region-based feature screening
Region-based feature screening
Feature pairs fusion
Restore feature pairs to the original location

Fig. 1 Algorithm flow chart

3.1 Down-sampling matching of aerial images

Searching the tie points to calculate air triangulation is the first main step for 3D reconstruction using aerial images. As shown in Table 1, when we use ContextCapture3D [2] to reconstruct sparse point clouds of 133 aerial images with different resolutions. It is found that as the resolution of the images increases, the number of the tie points decreases.

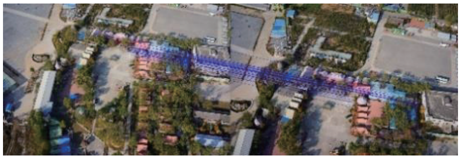
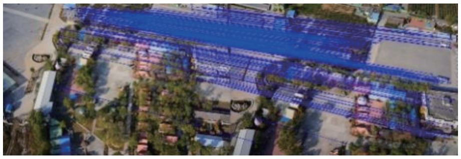
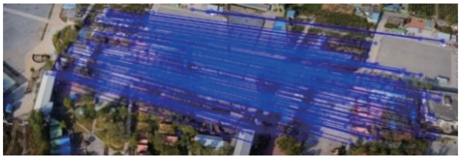
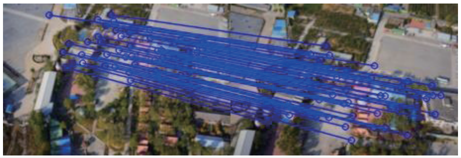
Considering that the SURF [4] algorithm and the RANSAC [32] algorithm are usually used for image feature matching, in order to verify the influence of image resolution on feature matching, we use aerial images to perform down-sampling matching experiments. For an image of the resolution of $N \times M$ pixels, if the downsampling coefficient is k, then a new image is formed by taking one pixel every k pixels in each row and column in the original image. We separately calculate the image results and data results, including (i) number of matching pairs before screening, (ii) number of matching pairs after screening, (iii) pairing time, (iv) screening time, as shown in Table 2.

Table 1 3D reconstruction results with different resolution images

Image resolution/ (pixels × pixels)	Key points' number of a single picture (median)	Number of all tie points	Tie points' number of a single picture (median)
1 840 × 1 228	6 693	35 388	1 229
3 680 × 2 456	24 254	32 609	1 007
7 360 × 4 912	43 455	23 924	828
14 720 × 9 824	43 549	13 859	400

In Table 2, the higher the image resolution, the more the feature points before screening, the longer the pairing time and the longer the screening time. However, the number of feature points after screening will become larger at first and then smaller. The results are consistent with those obtained by ContextCapture3D.

Table 2 Aerial image down-sampling matching experiment

Image resolution/(pixels × pixels)	Image result	Number	Time/s
3 680 × 2 456		(i) 71 751 (ii) 36	(iii) 335.27 (iv) 4.65
1 840 × 1 228		(i) 15 166 (ii) 36	(iii) 17.33 (iv) 1.21
920 × 614		(i) 3 848 (ii) 207	(iii) 2.93 (iv) 0.45
460 × 307		(i) 944 (ii) 164	(iii) 1.68 (iv) 0.28
230 × 153		(i) 232 (ii) 48	(iii) 1.21 (iv) 0.24

Therefore, in order to ensure real-time and quantitative balance of feature matching, the resolution of aerial images can be appropriately reduced according to the actual situation. In Section 4.1, experiments and analyses will be carried out on ten different aerial image pairs to obtain the relationship between the resolution of aerial images and the matching pairs and time. Furthermore, we will give the appropriate resolution of aerial images.

3.2 Region-based feature pair screening

In the above experimental results, although the RANSAC algorithm has been used for point pair screening, there are still some mismatched point pairs. Taking the image result of 460×307 pixels above as an example, the feature matching is not completely accurate, as shown by the circles in Fig. 2. It can be found that when the feature points are close, mismatching is likely to occur. Sometimes even one point corresponds to multiple points. At the same time, due to the local similar texture of different feature points, the general screening algorithm cannot completely remove the mismatching pairs. This problem is particularly evident in aerial images with repeating textures, such as buildings or forests.

The number of the feature pairs after the RANSAC [32] screening is 164. At this time, if the ratio algorithm or the crosscheck algorithm is continued to be used for matching screening, the number of remaining feature pairs may decrease sharply. Therefore, we propose a region-based feature pair screening algorithm, which is designed for the characteristics of high similarity between adjacent aerial images. It includes a screening algorithm based on similarity judgment and a screening algorithm based on local optimization.

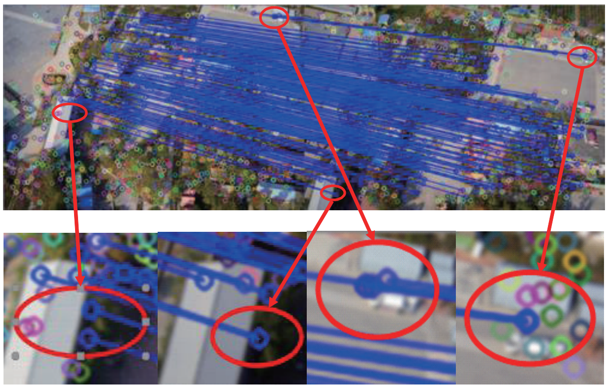


Fig. 2 Mismatching result

3.2.1 Similarity judgment based screening algorithm

The screening algorithm based on similarity judgment uses image similarity indices such as peak signal to noise ratio (PSNR), structural similarity (SSIM) index and perceptual hash to further screen the matched pairs. By comparing the image similarity of the same matching pairs' surrounding areas, we can judge whether it is the same area, and then eliminate the point pairs that do not belong to the same area.

PSNR can be used to judge the similarity of the images. The larger the PSNR value between the two images is, the more similar the two images are. The PSNR value P can be calculated by the following equation:

$$P = 10 \times \lg \frac{(2^n - 1)^2}{\mathrm{MSE}} \tag{1}$$

where MSE is the mean square error between the two surrounding areas, n is the bits' number of the sampled value. For example, the corresponding n value of a 0–255 grayscale image is 8.

SSIM is an indicator to measure the similarity between two images. It is designed with the visual characteristics of human eyes, which is more in line with the human visual perception than the traditional methods. The larger the SSIM value, the more similar the two images. The SSIM value $S(x,y)$ can be calculated based on different windows, if the size of the window is $n \times n$ pixels, then $S(x,y)$ can be calculated by the following equation:

$$S(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{2}$$

where (x,y) is the coordinate of pixels in the image; μ_x and μ_y represent the means of x and y; σ_x and σ_y represent the standard deviations of x and y; σ_{xy} represents the covariance of x and y. c_1 and c_2 are constants to avoid the denominator being 0.

The calculation of the perceptual hash value is to generate a "fingerprint" string for each image and then compare the fingerprints of different images. The closer the results are, the more similar the images are. Mean hash is a kind of perceptual hash, which typically scales down photos and simplifies colors to make the complex information easier. In this work, we improve the mean hash calculation. In order to preserve image information, we do not make size and color change. The procedures of images' similarity comparison using the improved mean hash value H is given as follows.

Step 1 Calculate the grayscale average G_avg of all pixels in the image.

Step 2 For each pixel in the image, compare the gray level G with G_avg. If $G \geqslant G$_avg, the result R is 1. If $G < G$_avg, R is 0.

Step 3 Combine R of each pixel to form a string, which is the fingerprint of this image. Ensure that all images are in the same combination order.

Step 4 Compare the fingerprints of different images and calculate the difference value H. The bigger the difference, the more different the images.

Using the above three similarity judgment indices, we can screen out the mismatches. The screening algorithm based on similarity judgment contains the following steps.

Step 1 Obtain the feature pairs to be screened and the related data of them in the two images.

Step 2 Set $i = 1$ at first, and obtain the feature points of the number i pair in both two images.

Step 3 Select square regions of $n \times n$ pixels centering on the two feature points.

Step 4 Calculate $P, S(x,y)$ or H of the two square regions, and then set a threshold T corresponding to different indices.

Step 5 Compare $P, S(x,y)$ or H with the corresponding T. If the image similarity is high enough, retain the matching pair. Otherwise, regard the number i pair as a mismatch and delete it.

Step 6 Set $i = i + 1$, repeat Step 2 to Step 5 until $i = N$, where N is the number of the matched pairs. Finish the loop.

3.2.2 Local optimization based screening algorithm

When a local area has a large number of feature points, mismatches will occur, making it difficult to screen. Therefore, in order to improve the matching accuracy, we use the screening algorithm based on local optimization to choose the best pairs in local areas. The algorithm mainly determines the most similar feature points according to the distance of feature points and the degree of similarity. The screening algorithm based on local optimization contains the following steps.

Step 1 Select a distance threshold D.

Step 2 Set $i = 1$.

Step 3 Obtain the feature points of the number i pair in both two images.

Step 4 Set $j = i + 1$.

Step 5 Obtain the feature points of the number j pair in both two images.

Step 6 Calculate the distance d between the point in number i pair and the point in number j pair in the first image.

Step 7 Compare D with d. If $D \geqslant d$, calculate the matching degree of these two pairs, retain the more similar one, and turn to Step 8. If $D < d$, turn to Step 9.

Step 8 Determine whether the retained pair is the number i pair. If yes, turn to Step 9. If no, turn to Step 11.

Step 9 Set $j = j + 1$.

Step 10 Determine whether the number $j - 1$ pair is the last pair. If yes, turn to Step 11. If no, turn to Step 5.

Step 11 Set $i = i + 1$.

Step 12 Determine whether the number *i* pair is the last pair. If yes, finish the loop. If no, turn to Step 3.

3.3 Fusion of different matching algorithms

Different feature algorithms have different applicability, so the matching feature pairs of different algorithms will have a big difference. For example, for the same two images, the SURF algorithm [4] and the ORB algorithm [5] will result in different matches. In order to increase the number of feature pairs and make the algorithm more applicable, we can use two or more traditional image matching algorithms in Section 2 to match the images and fuse the matching results together. Therefore, we put forward the fusion algorithm for feature pairs of different matching algorithms to make feature pairs dense.

To give an example, we use SURF [4] + RANSAC [32] and ORB [5] + RANSAC [32] to fuse. The process is shown in Fig. 3 and the steps are shown as follows.

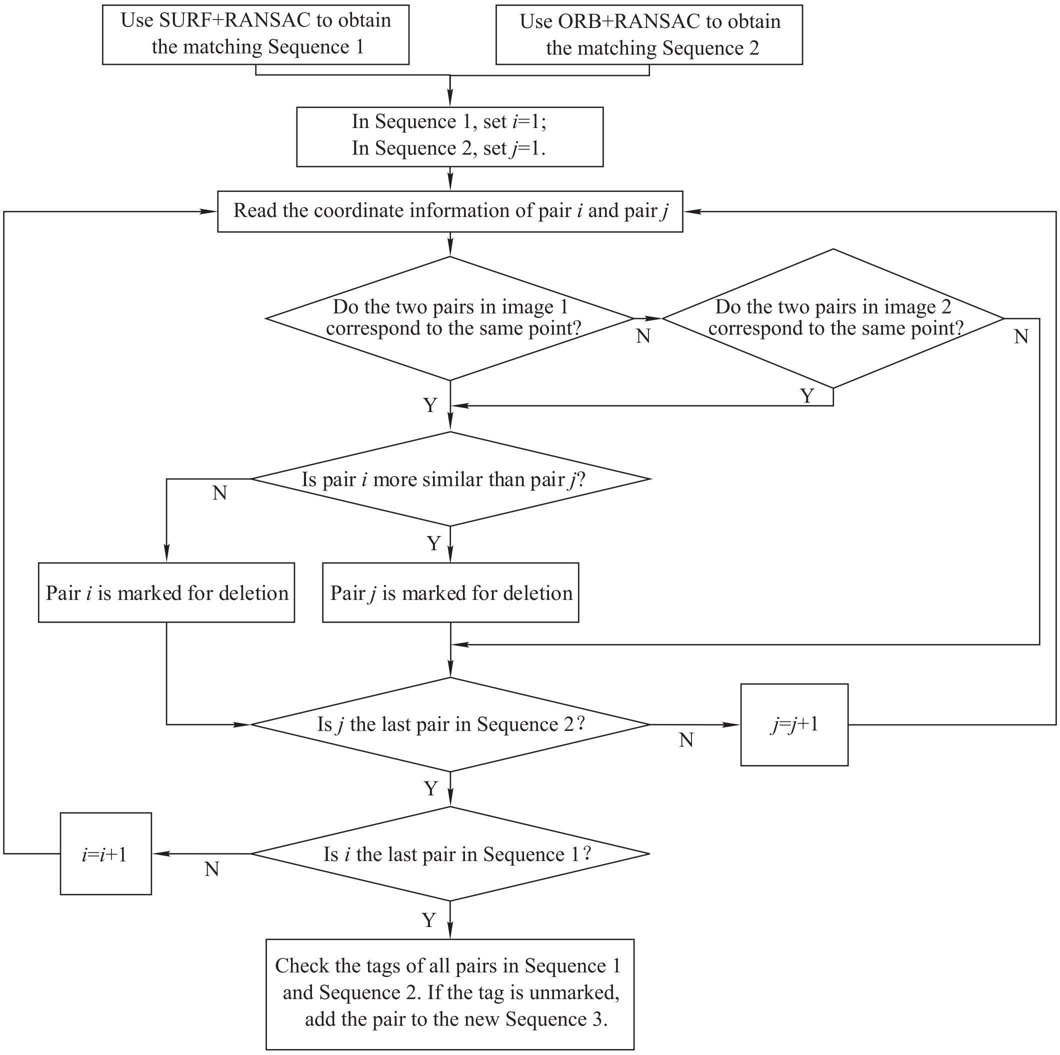

Use SURF+RANSAC to obtain the matching Sequence 1
Use ORB+RANSAC to obtain the matching Sequence 2
In Sequence 1, set *i*=1; In Sequence 2, set *j*=1.
Read the coordinate information of pair *i* and pair *j*
Do the two pairs in image 1 correspond to the same point?
N
Do the two pairs in image 2 correspond to the same point?
N
Y
Y
Is pair *i* more similar than pair *j*?
N
Y
Pair *i* is marked for deletion
Pair *j* is marked for deletion
Is *j* the last pair in Sequence 2?
N
j=*j*+1
Y
i=*i*+1
N
Is *i* the last pair in Sequence 1?
Y
Check the tags of all pairs in Sequence 1 and Sequence 2. If the tag is unmarked, add the pair to the new Sequence 3.

Fig. 3 Fusion of the feature pairs of SURF and ORB

Step 1 Use two algorithms to obtain two sets of different matching pairs called Sequence 1 and Sequence 2.

Step 2 Successively extract the feature pairs in Sequence 1 and Sequence 2, and read the coordinate information of the point pairs.

Step 3 Compare the coordinates to know if there is any coincident pair or coincident point.

Step 4 If there are coincident pairs or points, find the two pairs' information and judge which pair has higher similarity. Then, mark the less similar pair.

Step 5 Check the tags of all pairs in Sequence 1 and Sequence 2. If the tag is unmarked, add the pair to Sequence 3. Sequence 3 is the fusion result.

3.4 Matching pairs restoration

The following operations have been completed after the above process: fast feature matching after down-sampling, error pairs elimination through screening, and point pairs increasing through fusion of the methods. Finally, the matched pairs need to be restored to the original images to achieve high-resolution feature pair matching. Since each point actually corresponds to a small surrounding area after restoration (the specific size is related to k), the most similar pair in the surrounding area can be used as the matching pair after restoration. The specific steps include the following parts.

Step 1 In Section 3.1, for an image of $N \times M$ pixels resolution, if the down-sampling coefficient is k, we take one pixel every k pixels in each row and column to form a down-sampled image. Therefore, in this part, multiply the coordinates of each point in pair i by k to achieve a preliminary restore operation, where $i = 1$ at first.

Step 2 Take each restored feature point as the center, and take the range of $(k+1)\times(k+1)$ pixels around it as the region to be matched.

Step 3 Calculate the descriptors of each point in the surrounding region, and the points in the surrounding region are matched in turn with the points in the corresponding region.

Step 4 Find the most similar point pairs in the two regions and reserve them as the final matching pair i.

Step 5 If pair i is not the last pair, set $i = i+1$, and repeat the restore operation from Step 1 to Step 4.

4. Experiments

Experimental results and data analyses are provided in this section. Section 4.1 introduces the influence of image down-sampling. Section 4.2 presents the region-based feature pair screening. Section 4.3 provides the fusion algorithm results. Section 4.4 describes the restoration to the original resolution. Section 4.5 summarizes the experimental results of all steps.

4.1 Influence of image down-sampling

To quantitatively analyze the impact of image resolution on matching points and time, we define the average time spent on each point pair as AToEP, and the AToEP value A can be calculated as

$$A = \frac{\mathrm{TT}}{\mathrm{NOM}} \tag{3}$$

where TT is the total time, NOM is the number of output matching pairs.

Ten aerial image pairs taken from different angles and different scenes are selected. They are processed into images with different resolutions to calculate A and NOM. The results are shown in Fig. 4 and Fig. 5. A small A and a large NOM are required.

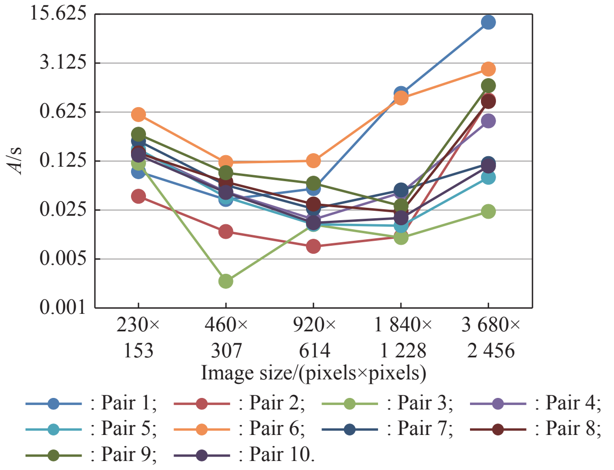


Fig. 4 AToEP changes of different image pairs

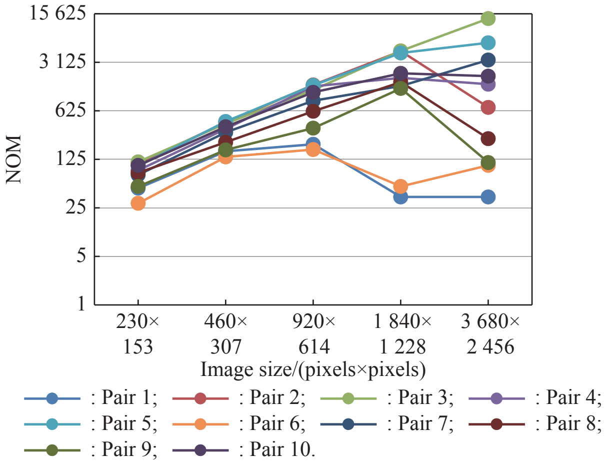


Fig. 5 Matching pair number changes of different image pairs

From the results shown above, the images with the resolution between 500×500 pixels and $2\,000 \times 2\,000$ pixels can achieve a good balance between time and the number of point pairs. They not only have low AToEP values, but also have a large number of matching pairs. Therefore, the images can be reduced to a certain range such as 500 × 500 pixels to 2 000 × 2 000 pixels. After that, feature matching and feature screening can be done.

4.2 Region-based feature pair screening

The high threshold value of PSNR or SSIM or the low threshold value of mean hash indicates the high requirements on image similarity and few feature pairs are retained through final screening. Meanwhile, when selecting the size of local areas to match, due to the different angles of different images, there will be some parallax occlusion. As a result, when the edge size of the local area is large, the time spent is long, and the final reserved pairs are sparse. When the area is too small, it will result in a large number of correct pairs being screened out. The same is true for the screening algorithm based on local optimization.

After experimental comparison, three aspects are comprehensively considered: quantity of retained features, quality of retained features and time of error pairs removal.

For the screening algorithm based on similarity judgment, the local area side length is 20 pixels, the threshold value of PSNR is 8, the threshold value of SSIM is 0.1, and the threshold value of mean hash is 200. For the screening algorithm based on local optimization, the distance *D* between the points is 8 pixels.

We analyze the effect of the algorithm qualitatively and quantitatively.

The qualitative analysis is to compare the experimental image results from the visual angle, and judge whether the wrong matching point pairs are removed. We still choose to conduct qualitative experimental comparison with Fig. 3, from which we can see the mismatching pairs in the red circles obviously. If the screening algorithm proposed is effective, we can see the reduction of mismatches from the images.

Meanwhile, we use the NCM pairs to carry out the quantitative analysis of accuracy. The method to calculate the matching accuracy is as follows.

Step 1 Obtain the correspondence between two images by calculating the homography matrix.

Step 2 Use affine transformation to get the corresponding positions of the feature points in image 1.

Step 3 Calculate the distance between the corresponding point and the matching point in image 2.

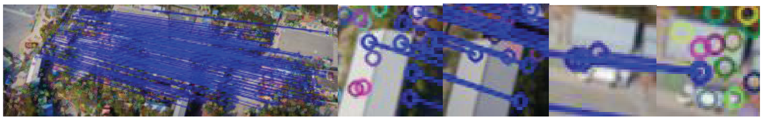
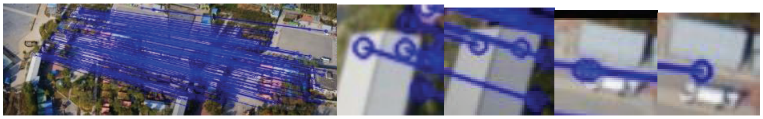
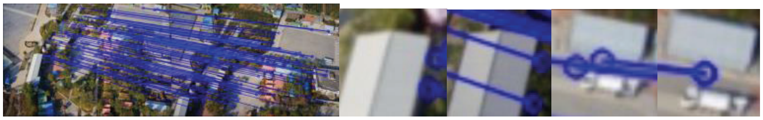
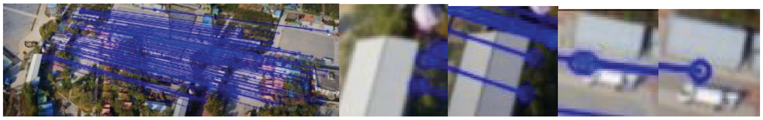
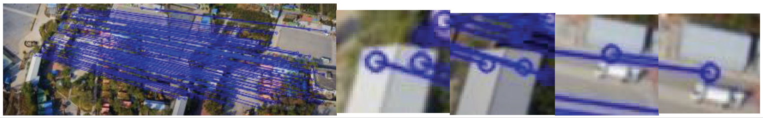
Step 4 Considering the occlusion between the images and the calculation error of the homography matrix, when the distance is less than 9 pixels (two percent of the image length), we think it is the correct matching pair.

In Section 4.1, we set the number of output matching pairs as NOM. Here, we set the number of correct matching pairs as NCM, and the number of wrong matching pairs as NWM. The above three indexes satisfy the following relation:

$$\mathrm{NOM} = \mathrm{NCM} + \mathrm{NWM}. \tag{4}$$

We carry out the screening experiments based on the similarity judgment algorithm and the local optimization algorithm, in which the similarity judgement experiments use the PSNR value, the SSIM value, and the improved mean hash value. In order to verify the accuracy, the feature pairs quantity and the screening time results, we compare the image results and the experimental data results, including (i) the number of matching pairs before screening, (ii) NOM, (iii) NCM, (iv) NWM, (v) screening time, as shown in Table 3.

Table 3 Region-based feature pair screening experiment

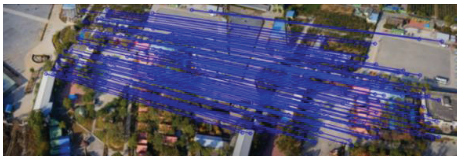
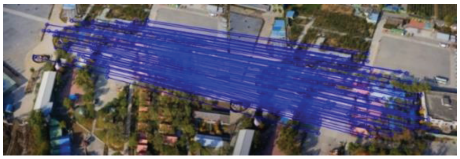
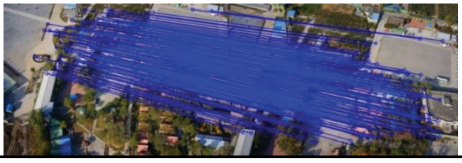
Algorithm	Image result	Number	Time/s
Original result		(i) 164 (ii) 164 (iii) 161 (iv) 3	–
PSNR		(i) 164 (ii) 151 (iii) 150 (iv) 1	(v) 0.691
SSIM		(i) 164 (ii) 87 (iii) 85 (iv) 2	(v) 0.676
Mean hash		(i) 164 (ii) 107 (iii) 105 (iv) 2	(v) 0.205
Local optimization		(i) 164 (ii) 118 (iii) 118 (iv) 0	(v) 1.355

According to the experimental results, the above algorithms can effectively screen out the mismatching pairs and improve the accuracy of matching in a relatively short time. Through the comparison of the results, it can be seen that the proposed screening algorithms have the relative consistency of the results. It is worth noting that the pixel threshold used here in calculating NCM can be adjusted according to the occlusion and the shooting distance of the adjacent images. Moreover, we can adjust the threshold of algorithms or select different algorithms according to the specific requirements such as real-time or feature pair quantity.

4.3 Fusion algorithm results

Since both the ORB algorithm and the SURF algorithm have the advantages of short matching time and stable matching effect, they are selected to do the fusion experiments. Meanwhile, in order to improve the speed and the accuracy, the down-sampled images in Fig. 3 and the screening algorithm based on local optimization are adopted. We perform experiments on the SURF algorithm, the ORB algorithm, and the fusion of the SURF algorithm and the ORB algorithm. Table 4 shows the statistics of image results, number of feature pairs and time.

Table 4 Experimental results of the algorithm based on feature pair fusion

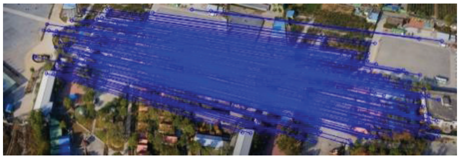
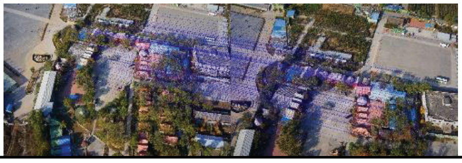
Algorithm	Image result	Number of pairs	Time/s
SURF		118	4.353
ORB		107	4.758
SURF + ORB		221	10.446

In the above experiment, the number of SURF matching pairs is 118, and the number of ORB matching pairs is 107. In the end, a total of 221 pairs are obtained. It can be seen that, after the fusion of the two algorithms, the final feature pairs cover a wider area and are much denser than that of only one algorithm.

4.4 Restoration to the original resolution

We restore the results from the fusion of the SURF algorithm and the ORB algorithm in Section 4.3 to the original images, and the comparison is shown in Table 5.

Table 5 Experimental results of restoration

Image resolution/(pixels × pixels)	Image result
460 × 307	
3 680 × 2 456	

4.5 Experimental results of all steps

In order to verify the feasibility of the algorithm, we carry out experiments on the aerial image pairs of three different regions as shown in Fig. 6.

The initial resolution of the aerial image pairs is 3 680 × 2 456 pixels. Using the conclusion obtained in Section 4.1, we down-sample the images to 920 × 614 pixels at first. Then we use the traditional algorithm (SURF+RANSAC) and the proposed algorithm to perform feature point matching. In order to observe the application of aerial image feature matching results in 3D reconstruction, a structure from motion (SfM) algorithm is carried out and the sparse point cloud reconstruction results are obtained. The experimental results are shown in Table 6, including image results, SfM results, time results, (i) NOM, and (ii) NCM.

From the above experimental verification, we can see that the number and the distribution of feature point pairs will directly affect the generation result of sparse point clouds. Compared with the traditional algorithm, the proposed algorithm can quickly obtain accurate, dense and uniformly distributed feature point pairs, which is very helpful for 3D reconstruction.

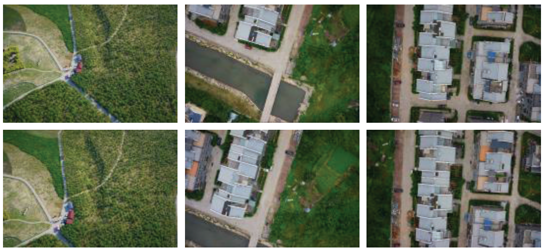
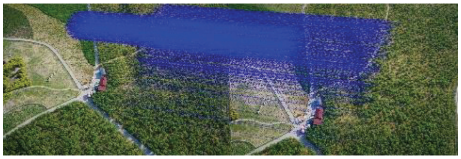
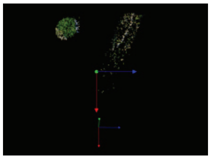
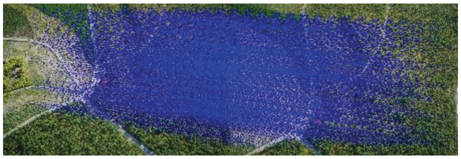
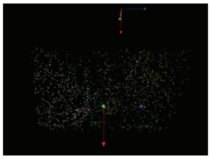
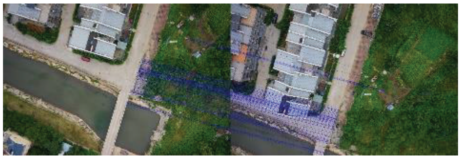
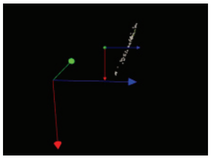
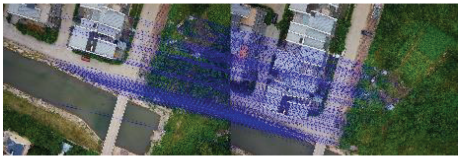
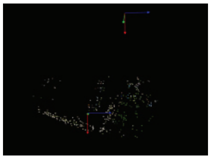
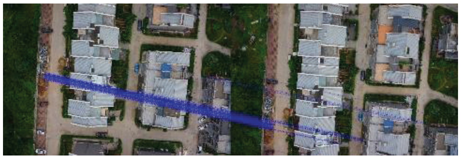
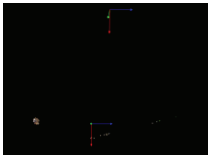
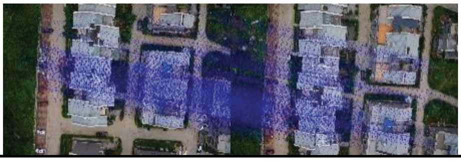
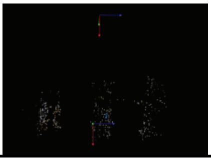


Fig. 6 Three groups of tested aerial images

Table 6 Comparison between SURF+RANSAC and our algorithm

Method	Image result	SfM result	Time/s	Number
SURF + RANSAC			697	(i) 1 501 (ii) 1 500
The proposed algorithm			344	(i) 1 862 (ii) 1 862
SURF + RANSAC			262	(i) 103 (ii) 103
The proposed algorithm			113	(i) 447 (ii) 446
SURF + RANSAC			167	(i) 113 (ii) 113
The proposed algorithm			129	(i) 486 (ii) 486

5. Conclusions

In this paper, a fast, accurate and dense feature matching algorithm is realized by down-sampling and matching the aerial images, screening out mismatches based on the region, fusing the pairs of different matching algorithms, and restoring the matching pairs to the original image. This algorithm can solve the feature matching problems of high-resolution aerial images using low-cost devices, such as long time, few matching points and uneven distribution. Our experiments show that the matching time is greatly reduced, the mismatches are removed and the matching pairs are increased and evenly distributed.

In the future, we will conduct the followup research, for example, further automatic selection of fusion algorithms for different image textures. In terms of matching time, the time pressure brought by two or more algorithms will be further studied.

References

[1] LIU T. Comparison of software Inpho, PhotoScan and Pix4D for UAV orthophoto processing. Technology and Industry

Across the Straits, 2017, 221(11): 82–84. (in Chinese)

[2] BENTLEY. Context capture: create 3D models from simple photographs. https://www.bentley.com/en/products/brands/contextcapture.

[3] LOWE D G. Distinctive image feature from scale-invariant keypoints. International Journal of Computer Vision, 2004, 60(2): 91–110.

[4] BAY H, TUYTELAARS T, GOOL L J V. SURF: speeded up robust features. Proc. of the European Conference on Computer Vision, 2006: 404–417.

[5] ROSTEN E, DRUMMOND T. Machine learning for high-speed corner detection. Proc. of the European Conference on Computer Vision, 2006: 430–443.

[6] RUBLEE E, RABAUD V, KONOLIGE K, et al. ORB: an efficient alternative to SIFT or SURF. Proc. of the IEEE International Conference on Computer Vision, 2011: 2564–2571.

[7] LEUTENEGGER S, CHLI M, SIEGWART R Y. Brisk: binary robust invariant scalable keypoints. Proc. of the IEEE International Conference on Computer Vision, 2011, 58(11): 2548–2555.

[8] ALAHI A, ORTIZ R, VANDERGHEYNST P. FREAK: fast retina keypoint. Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2012: 510–517.

[9] PABLO F A, ADRIEN B, ANDREW J D. KAZE features. Proc. of the European Conference on Computer Vision, 2012: 214–227.

[10] PABLO F A, JESUS N, ADRIEN B. Fast explicit diffusion for accelerated features in nonlinear scale spaces. Proc. of the British Machine Vision Conference, 2013: 1–12.

[11] TAREEN S A K, SALEEM Z. A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK. Proc. of the IEEE International Conference on Computing, Mathematics and Engineering Technologies, 2018: 1–10.

[12] LAM S K, LIM T C, WU M, et al. Data-path unrolling with logic folding for area-time-efficient FPGA-based FAST corner detector. Journal of Real Time Image Processing, 2019, 16(6): 2147–2158.

[13] CHEN S J, ZHENG S Z, XU Z G, et al. An improved image matching method based on SURF algorithm. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2018, 42(3): 179–184.

[14] ZHU J T, GONG C F, ZHAO M X, et al. Image mosaic algorithm based on PCA-ORB feature matching. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2020, 42(3): 83–89.

[15] GUO S L, HAN L N, HAO X T. GW28-e0419 image detection scale-invariant feature transform algorithm based on feature matching improves image matching accuracy. Journal of the American College of Cardiology, 2017, 70(16): C10.

[16] LI J Y, HU Q W, AI M Y. RIFT: multi-modal image matching based on radiation-variation insensitive feature transform. IEEE Trans. on Image Processing, 2020, 29: 3296–3310.

[17] YU X Y, LEI H, GAO W, et al. Image matching algorithm with color information based on SIFT. Proc. of the 10th International Conference on Digital Image Processing, 2018. DOI:10.1117/12.2502820.

[18] LI Q L, WANG G Y, LIU J G, et al. Robust scale-invariant feature matching for remote sensing image registration. IEEE Geoscience and Remote Sensing Letters, 2009, 6(2): 287–291.

[19] MA C, HU X, XIAO J, et al. Improved ORB algorithm using three-patch method and local gray difference. Sensors, 2020, 20(4): 975.

[20] WU Y. Research on feature point extraction and matching machine learning method based on light field imaging. Neural Computing and Applications, 2019, 31(12): 8157–8169.

[21] ZHANG Y, LI C, CAO C Q, et al. An improved ORB feature point matching algorithm. Proc. of the 2nd International Conference on Computer Science and Artificial Intelligence, 2018: 207–211.

[22] LI J Y, HU Q W, AI M Y. 4FP-structure: a robust local region feature descriptor. Photogrammetric Engineering and Remote Sensing, 2017, 83(12): 813–826.

[23] ZHU S Z, LIU L Z, CHEN S. Image feature detection algorithm based on the spread of Hessian source. Multimedia Systems, 2017, 23(1): 105–117.

[24] HARTMANN W, HAVLENA M, SCHINDLER K. Recent developments in large-scale tie-point matching. ISPRS Journal of Photogrammetry and Remote Sensing, 2016, 115: 47–62.

[25] XI W F, SHI Z T, LI D S. Comparisons of feature extraction algorithm based on unmanned aerial vehicle image. Open Physics, 2017, 15(1): 472–478.

[26] AMIN S, NAIILA M. Uniform competency-based local feature extraction for remote sensing images. ISPRS Journal of Photogrammetry and Remote Sensing, 2018, 135: 142–157.

[27] LI M M, XIE W F. Remote sensing image matching algorithm for coastal zone based on local features. Journal of Coastal Research, 2019, 93(sp1): 723–728.

[28] YUAN X X, CHEN S Y, YUAN W, et al. Poor textural image tie point matching via graph theory. ISPRS Journal of Photogrammetry and Remote Sensing, 2017, 129: 21–31.

[29] WU Y D, MING Y, TAN H. An efficient hyper-graph based tie point matching method for aerial imagery. Remote Sensing Letters, 2020, 11(4): 313–322.

[30] CHEN S T, FENG R, ZHANG Y Y, et al. Aerial image matching method based on HSI hash learning. Pattern Recognition Letters, 2019, 117: 131–139.

[31] SONG W H, JUNG H G, GWAK I Y, et al. Oblique aerial image matching based on iterative simulation and homography evaluation. Pattern Recognition, 2019, 87: 317–331.

[32] MARTIN A F, ROBERT C B. Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography. Communications of Association for Computing Machinery, 1981, 24(6): 381–395.

[33] GUI Y, SU A, DU J. Point-pattern matching method using SURF and shape context. Optik International Journal for Light and Electron Optics, 2013, 124(14): 1869–1873.

[34] XI W F, LI D S, LI J S, et al. Research on an improved algorithm for eliminating gross error in UAV image mosaic. Proc. of the 26th International Conference on Geoinformatics, 2018: 1–5.

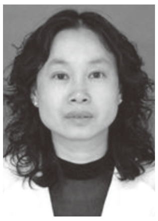
[35] LI J Y, HU Q W, AI M Y. LAM: locality affine-invariant feature matching. ISPRS Journal of Photogrammetry and Remote Sensing, 2019, 154: 28–40.

[36] GAO H W, JIANG Y Q, LIU J G, et al. Zooming image based false matches elimination algorithms for robot navigation. Advances in Mechanical Engineering, 2017, 9(12): 168781401773815.

[37] GESTO-DIAZ M, TOMBARI F, GONZALEZAGUILERA D, et al. Feature matching evaluation for multimodal correspondence. ISPRS Journal of Photogrammetry and Remote Sensing, 2017, 129: 179–188.

Biographies

LI Ying was born in 1995. She received her B.S. degree from Beijing Forestry University in 2017. Then she entered Beihang University for her M.S. degree and became a doctoral candidate in 2019. She is pursuing her Ph.D. degree in navigation, guidance and control from Beihang University, China. Her research interests include image processing, environment modeling and simulation, virtual reality and augmented reality.
E-mail: 1193126485@qq.com



GONG Guanghong was born in 1968. She received her B.S., M.S. and Ph.D. degrees from Beihang University, in 1990, 1993 and 1997, respectively. She is currently a professor in the Key Laboratory of Advanced Simulation Aeronautical Technology, Beihang University. She is also an executive director of the China Simulation Society and a director of the Environmental Modeling and Simulation Professional Committee. She has won the Science and Technology Progress Award of the Ministry of Education and the National Defense Science and Technology Progress Award for many times. Her research interests are virtual reality and distributed interactive simulation.
E-mail: ggh@buaa.edu.cn

SUN Lin was born in 1995. He received his B.S. and M.S. degrees from Beihang University. During his master’s degree program, his main research direction is terrain 3D reconstruction. His research interests include virtual reality and environment simulation.
E-mail: sunlin 2019@163.com