IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received 19 July 2024, accepted 8 August 2024, date of publication 13 August 2024, date of current version 13 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3443111

RESEARCH ARTICLE

A Fast Image Stitching Algorithm Based on Texture Classification and Improved SIFT

ZETIAN TANG[1], ZEMIN ZHANG[2], JUNJIE FENG[1], WEI CHEN[3], KUN ZHU[1], WENTAO YANG[1], AND WENJUAN HAN[1]

[1]School of Physics and Electrical Engineering, Liupanshui Normal University, Liupanshui 553004, China
[2]School of Chemistry and Materials Engineering, Liupanshui Normal University, Liupanshui 553004, China
[3]School of Electronics and Communication Engineering, Sun Yat-sen University, Shenzhen 518107, China

Corresponding author: Junjie Feng (2651034399@qq.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 62065003, in part by the Natural Science Research Project of Guizhou Provincial Education Office under Grant Qianjiaohe KY Zi [2022] 046, in part by the Scientific Research Project of Liupanshui Normal University under Grant LPSSYZK202016, in part by Liupanshui Normal University Discipline Team under Grant LPSSY2023XKTD12, and in part by the Science and Technology Innovation Talent Team of Liupanshui Science and Technology Bureau under Grant 52020-2023-0-2-20 and Grant 52020-2023-0-2-19.

ABSTRACT A fast and improved scale-invariant feature transform (SIFT) image stitching algorithm is proposed based on texture classification to solve the problem of huge computational complexity. In the preprocessing stage, the phase correlation algorithm is used to calculate the overlapping regions of the images, and the structural similarity (SSIM) of the overlapping regions are calculated to avoid the impact caused by inaccurate calculation of the phase correlation algorithm. Meanwhile, gradient based texture classification method is used to avoid ineffective calculations in weak texture regions. A circular eight regions descriptor structure was designed in the descriptor generation stage. And the sum of gradients in five directions within each region was calculated to obtain a feature descriptor with a dimension of only 40. The time of feature point matching was reduced due to the descriptors of lower dimensions. Further, a twice matching method was proposed based on extreme value classification to reduce the time cost of feature point matching. The experimental results show that compared to existing algorithms, this algorithm has the best performance in terms of time cost and stitching quality in two datasets. Compared to the SIFT algorithm, the time was reduced by 73.24% and 47.58%, the root mean square error (RMSE) was reduced by 94.87% and 84.36%, the number of images with failed stitching has decreased by 93.35%. The proposed algorithm significantly reduces the time cost and improves the quality of image stitching. The proposed algorithm has certain application value in the field of real-time image stitching.

INDEX TERMS Scale-invariant feature transform (SIFT), texture classification, extreme value classification, twice matching.

I. INTRODUCTION

Image stitching is a technique that calculates the transformation relationship between two or more images and fuses them to obtain a larger field of scene. Image stitching is one of the important research projects in the field of image processing, widely used in super-resolution reconstruction, medical detection, remote sensing, and other fields [1], [2], [3].

At present, feature-based image stitching algorithms have better performance, such as Harris [4], features from accelerated segment test (FAST) [5] oriented FAST and rotated BRIEF (ORB) [6], binary robust invariant scalable keypoints (BRISK) [7], scale-invariant feature transform (SIFT) [8]. The SIFT algorithm has good invariance in rotation, scaling, and affine transformations. The SIFT algorithm also achieves good performance in accuracy and robustness, which can effectively ensure the quality of image stitching. Therefore, the SIFT algorithm has attracted the attention of many

The associate editor coordinating the review of this manuscript and approving it for publication was Orazio Gambino.

researchers. However, the computational complexity of the SIFT algorithm is enormous, making it difficult to meet the real-time requirements in engineering.

In recent years, numerous researchers have proposed solutions from various stages of image stitching. Cai et al. [9] combined edge detection to segment subregions with rich edge information in the preprocessing stage, limiting the region for feature point extraction. Li et al. [10] divided the image to be stitched into four sub-blocks and calculated similar sub-blocks through the normalized cross-correlation coefficient of image energy. Liu et al. [11] and Li et al. [12] divided image regions based on the similarity of shared information between images. Shi et al. [13] calculated the feature blocks of overlapping regions using the fuzzy C-means algorithm. Liu et al. [14] determined the overlapping regions of images using a binary tree model and the BRISK algorithm. Li et al. [10], Liu et al. [11], Li et al. [12], Shi et al. [13], and Liu et al. [14] all restricted the algorithm to similar regions, reducing the computational region of the algorithm. The methods of calculating similar sub-blocks and overlapping regions can effectively reduce the computational regions required by the algorithm. However, the image stitching effect will decrease when the calculation results are inaccurate.

In the feature point matching stage, Yang et al. [15] used the HSI color model to constrain the random sample consensus (RANSAC) algorithm. Ma et al. [16] proposed a guided local preservation matching method. Wang et al. [17] used RANSAC algorithm for coarse matching and least squares matching for fine matching. Kupfer et al. [18] limited the matching range of feature points by using their information. Zhou et al. [19] reduced the dimensionality of SIFT descriptor by using convolutional neural networks, thereby reducing the data dimensionality in the matching stage. Li et al. [20] have also limited the matching space of feature points at the low-level pyramids started matching from the high-level pyramids to achieve the matching from rough to fine. Wang et al. [21] calculated the feature blocks of images using the fuzzy C-Means algorithm and combined the information of the feature blocks during the matching stage to reduce the search space for feature point matching. Gao et al. [22] the gridded BRISK method was used to improve the efficiency of feature point matching. Deng et al. [23] improved the accuracy and efficiency of feature point matching by introducing block link constraints and shape distortion constraints. Wang et al. [24] limited the space for feature point search during matching by calculating the coordinates of the image, thus accelerating the speed of stitching. Zhang et al. [25] added genetic algorithm in the feature point matching stage, which improved the efficiency of feature point matching. An et al. [26] used cosine similarity and bidirectional consistency detection for feature point matching, reducing the time cost of the algorithm. At this stage, in order to improve the efficiency of the matching stage, existing algorithms usually combine the feature information of the image to constrain the space of feature point search. However, the efficiency of feature point matching will be affected when the number of feature points is small and the calculation of constraint relationships between feature points is too complex.

The above researches have focused on improving the speed of image stitching algorithms in one stage, so there are certain limitations to their improvement. To further improve the speed of the algorithm, some researchers have made improvements from multiple aspects. Wang et al. [27] used phase correlation algorithms and texture classification to calculate image overlapping regions and texture classification. They also use the results of texture classification to limit the space for feature point matching search. The speed of image stitching has been accelerated in the preprocessing stage and feature point matching stage. Chen et al. [28] used the Canny edge detection operator to limit the range of feature point detection, and designed an 18 dimensional circular descriptor. Liu et al. [29] detected overlapping regions in images and used the descriptor based on gradient normalization for feature point matching. The above two teams have reduced the time cost of the algorithm from two aspects preprocessing and descriptor. Zhao et al. [30] optimized the number of feature point extraction by changing the contrast threshold of feature point extraction. Meanwhile, they eliminated mismatches based on location information and RANSAC. Liang et al. [31] made a design based on the characteristics of FAST, weighted angular diffusion radial sampling (WADRS) and matching method. This method can achieve fast registration while ensuring registration accuracy. Liu et al. [32] downsampled the image to reduce the computational complexity and the number of feature points. Meanwhile, using MN-SIFT feature descriptors for feature point matching reduces the time cost of the matching stage. Wu et al. [33] used the SIFT-OCT algorithm for feature point extraction, which reduced the number of feature points. In addition, the efficiency and quality of matching had been improved using arccos and fast sample consensus (FSC) algorithm [34] for feature point matching. Zhao et al. [30], Liang et al. [31], Liu et al. [32], and Wu et al. [33] have improved the speed of the algorithm through two stages: feature point extraction and matching. Through of Wang Yuhao and coworker's [35] efforts, a mask method was used for the algorithm to limit the space for feature point extraction in the preprocessing stage. Furthermore, they structured a smaller dimensional descriptor and proposed a feature point matching method for extreme value classification in the stages of descriptor and feature point matching. This algorithm has achieved good accuracy and efficiency. Tang et al. [36] calculate the overlapping regions of images by phase correlation algorithm. They improved the feature point extraction method of the SIFT algorithm, and limited the number of feature point extraction. They also structured a feature descriptor with only 56 dimensions. This algorithm greatly reduces the time cost of high-resolution image stitching. Wang et al. [35] and

Tang et al. [36] improved the efficiency of image stitching in three stages. The improvements made in each stage of image stitching in the above literature are shown in table 1.

TABLE 1. Improvement of image stitching algorithms by various algorithms.

Reference number	Preprocessing stage	Feature point extraction stage	Descriptor generation stage	Feature point matching stage
[9-14]	√			
[15-26]				√
[27]	√			√
[28-29]	√		√	
[30-33]		√		√
[35]	√		√	√
[36]	√	√	√	

The image stitching algorithm's speed has been better enhanced from many aspect improvements. In the preprocessing stage, current research typically requires independent calculations for regions of interest or texture in images. However, the SIFT algorithm also requires calculating gradients, which can reflect the strength of image textures. Therefore, we would reduce the time cost of the preprocessing stage by using image gradients for texture classification. Regarding the construction of descriptors, current research is considering how to construct lower dimensions while ensuring the accuracy of feature point matching. A descriptor with good accuracy and lower dimensions was designed. In the feature point matching stage, the research focus is on how to have good matching accuracy and less time cost. On the basis of reference [35], we propose a simple and fast feature point matching method, and use FSC algorithm for fine matching, achieving good stitching quality and efficiency. We compared it with the current researches in the three stages of our improvement. The experimental results show that the proposed algorithm has the lowest time cost and good performance in each stage. In summary, the SIFT algorithm has been improved in three stages, and the main work is as follows:

(1) In the preprocessing stage, the phase correlation algorithm was used to calculate the overlapping regions of the images. Then, the structural similarity (SSIM) [37] of the overlapping regions calculated to ensure the accuracy of the phase correlation algorithm calculation. Next, the overlapping regions are divided into complex texture regions and weak texture regions based on gradients. And the computational space of the SIFT algorithm was limited to complex texture regions, which thereby reducing the computation of useless regions.

(2) In the descriptor generation stage, we designed a circular descriptor with only 40 dimensions. This descriptor has good matching performance, which effectively reduces the time cost of feature point matching stage.

(3) In the feature matching stage, a twice matching method based on extreme value classification was proposed, and the FSC algorithm was used for fine matching. This method achieved better matching results and the time overhead was reduced.

This paper is organized as follows. Section II shows the steps and principles of the SIFT algorithm. Section III introduces the proposed algorithm, detailing the preprocessing method, descriptor structure, and feature point matching method. Section IV introduces the experimental datasets, platform, and analyzes the parameter settings of the algorithm. Section V presents objective evaluation indicators and analyzes algorithm performance. Finally, Section VI makes a summary of this paper.

II. SIFT IMAGE STITCHING ALGORITHM

The flowchart of the SIFT image stitching algorithm is shown in figure 1. Firstly, the SIFT algorithm extracts spatial extremum points in the scale space. Then, unstable edge response points and extreme points of poorer contrast are removed to obtain the keypoints of the image. Next, generate descriptors for keypoints to represent the image information around them. Finally, match the feature points and calculate the correspondence between images for image fusion.

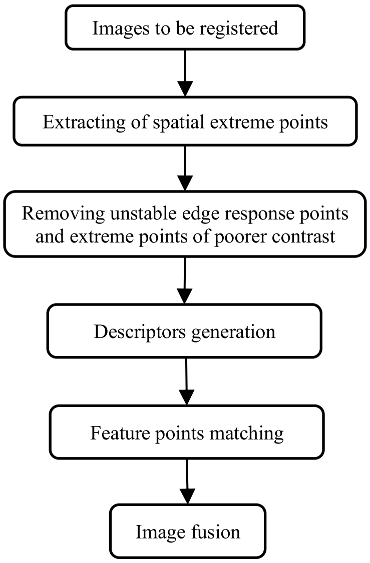

Images to be registered
Extracting of spatial extreme points
Removing unstable edge response points and extreme points of poorer contrast
Descriptors generation
Feature points matching
Image fusion

FIGURE 1. Flow chart of SIFT image stitching algorithm.

A. EXTRACTING OF SPATIAL EXTREME POINTS

The image is downsampled to obtain images of different sizes. And convolve the images using a Gaussian filter to obtain a Gaussian pyramid, as follows:

$$L(x, y, \sigma) = G(x, y, \sigma) \otimes I(x, y) \quad (1)$$

where, $I(x, y)$ represents the input image, $G(x, y, \sigma)$ represents the Gaussian filter. The Gaussian filter is as follows:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x-m/2)^2+(y-n/2)^2}{2\sigma^2}} \quad (2)$$

where, σ denotes the standard deviation of the normal distribution, m and n are the size of the Gaussian filter, and x and y are the positions of the corresponding elements of the Gaussian filter.

Then, subtract between two adjacent layers of an octave of Gaussian pyramids to obtain the difference of Gaussian (DOG), as follows:

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \tag{3}$$

where, $L(x, y, k\sigma)$ from formula (1), k represents the scale factor of two scale spaces. The DOG model is shown in figure 2.

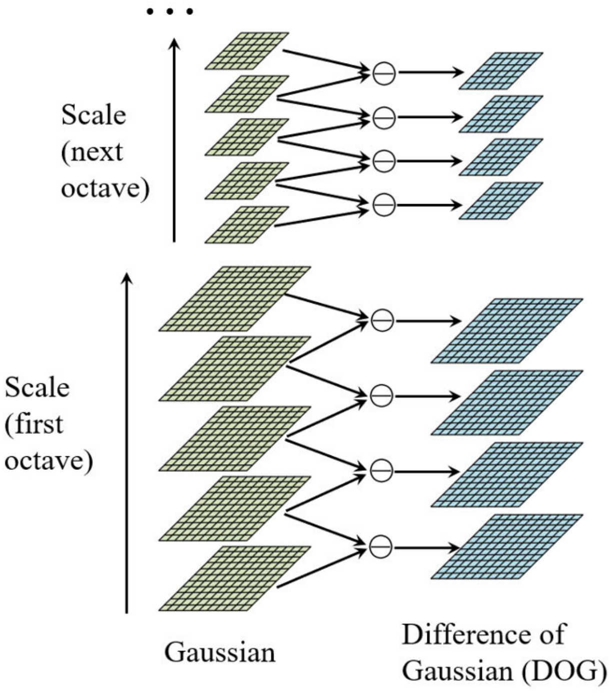


FIGURE 2. DOG model.

Next, search for local extremum points after establishing the DOG. As shown in figure 3, compare any point (x, y) with all points in a $3 \times 3 \times 3$ region centered on that point, it is a local extreme point when the value of point (x, y) is the maximum or minimum value.

B. REMOVING UNSTABLE EDGE RESPONSE POINTS AND EXTREME POINTS OF POORER CONTRAST

The extreme points of low and the edge response points of poor stability contrast need to be further removed after the extreme points are obtained.

Calculate the $D(\hat{X})$ value of the extreme point, and the calculation formula is as follows:

$$D(\hat{X}) = D + \frac{1}{2}\frac{\partial D^T}{\partial X_0}\hat{X} \tag{4}$$

where, $\hat{X} = (x, y, \sigma)^T$ denotes the center offset of the relative interpolation, D denotes the first term of the Taylor expansion of the spatial scale function $D(x, y,\sigma)$ at the extreme point. Then, all extreme with a value of $D(\hat{X})$ less than 0.03 were discarded to obtain more stable extreme points.

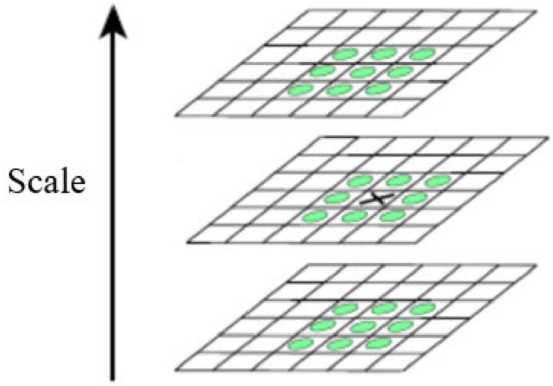


FIGURE 3. Search for local extremum points.

The location and scale of a feature point can be pinpointed accurately by a Hessian matrix, which can be expressed as:

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \tag{5}$$

Then, the stability of the point is presented by the following formula:

$$stability = \frac{(D_{xx} + D_{yy})^2}{D_{xx}D_{yy} - D_{xy}^2} < \frac{(r+1)^2}{r} \tag{6}$$

where, r represents the parameter which controls feature value. Remove all points that do not conform to formula (6).

C. DESCRIPTOR GENERATION

For the SIFT operator to be rotated invariant, the main direction of the keypoints needs to be determined by the following formula:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) – L(x, y-1))^2} \tag{7}$$

$$\theta(x, y) = \arctan(\frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)}) \tag{8}$$

where L represents the scale space value of the keypoints, and formulas (7) and (8) are the corresponding gradient modulus value and direction, respectively. The direction of the maximum is the main direction. In addition, if the direction is greater than 80% of the maximum value, set them as secondary directions for the keypoints. All main and secondary directions will be retained as separate feature points.

Then, the pixels around the feature points are rotated to the corresponding direction to ensure the invariance of the descriptor direction. As shown in figure 4, the rotated region is divided and the cumulative gradient is calculated in eight directions of each subregion to form a 128 dimensional feature descriptor.

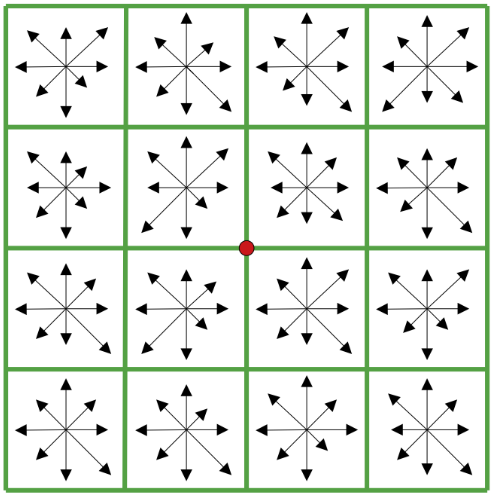


FIGURE 4. Descriptor of SIFT.

D. FEATURE POINTS MATCHING

Feature points need to be matched after generating descriptors. The SIFT algorithm needs to calculate the distance between feature point descriptors between two images during matching, and the calculation formula is as follows:

$$d(i,j) = \sqrt{\sum_{k=1}^{n} (x_i(k) - y_j(k))^2} \tag{9}$$

where $d(i, j)$ represents the distance between the i-th feature point in the first image and the j-th feature point in the second image, n represents the dimension of the descriptor, and $x_i(k)$ and $y_j(k)$ denote the descriptors of the i-th and j-th feature points in the two images, respectively.

III. PROPOSED METHOD

The main work is to improve the efficiency of the SIFT algorithm through multiple stages of improvement. The algorithm flowchart is shown in figure 5. Firstly, in order to reduce unnecessary calculations in non-overlapping regions in subsequent processes, phase correlation algorithm was used to calculate the overlapping regions between the images to be stitched. But the SSIM of the overlapping regions ($\mathrm{SSIM_o}$) were calculated to ensure the accuracy of the phase correlation algorithm. When the $\mathrm{SSIM_o}$ is greater than or equal to the set threshold T_p, it indicates that the phase correlation algorithm calculates accurately. And subsequent algorithms use the images of the overlapping regions. Otherwise, if the phase correlation algorithm calculates incorrectly, subsequent algorithms will use the original images. Then, the image is divided into complex texture regions and weak texture regions based on the gradient of the image. And further limit the SIFT algorithm's calculation in complex texture regions, which thereby avoids the SIFT algorithm from performing calculations in regions where feature points cannot be effectively extracted. Subsequently, feature point detection and descriptor generation were performed on complex texture regions. In the descriptor generation stage, a circular eight regions descriptor structure was designed, and the sum of gradients in five directions within each region was calculated to obtain a feature descriptor with a dimension of only 40. Descriptors of lower dimensions reduce the time of feature point matching. Next, in the feature point matching stage, a twice matching method based on extreme value classification is proposed, which effectively reduces the time cost of the feature point matching stage by performing extreme value classification and twice matching on the feature points. Finally, use the FSC algorithm for fine matching and calculate the projection transformation matrix to complete image fusion.

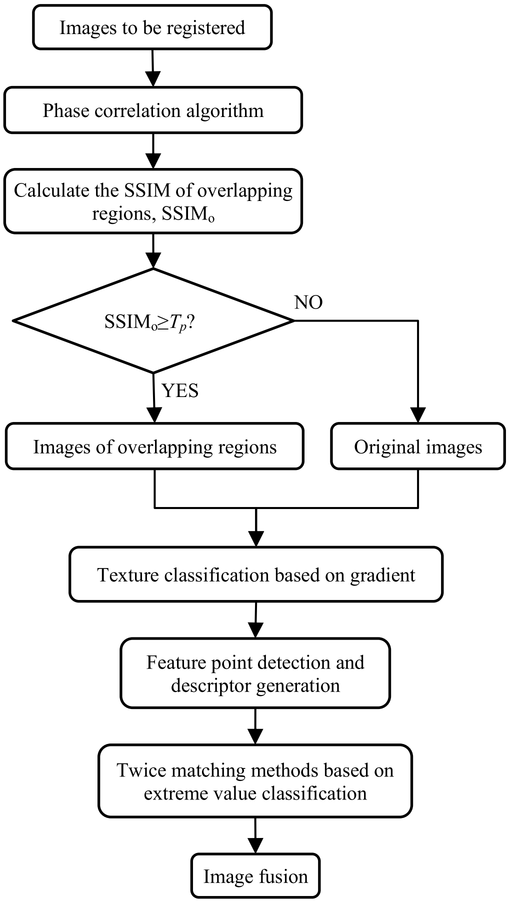

Images to be registered
Phase correlation algorithm
Calculate the SSIM of overlapping regions, SSIM$_o$
SSIM$_o$≥T_p?
NO
YES
Images of overlapping regions
Original images
Texture classification based on gradient
Feature point detection and descriptor generation
Twice matching methods based on extreme value classification
Image fusion

FIGURE 5. Algorithm flow chart.

A. PHASE CORRELATION ALGORITHM AND SSIM

The phase correlation algorithm can effectively calculate the overlapping regions between images, which can reduce ineffective calculations in non-overlapping regions. However, phase correlation algorithms can only calculate the offset between images. The calculation results of phase correlation algorithms may be inaccurate when there are complex image transformations between images. Therefore, in order to determine the images to be used by subsequent algorithms, we calculate the SSIM of the overlapping regions

($SSIM_o$) after calculating the phase correlation algorithm. When SSIMo is greater than or equal to the set threshold $T_p(0.52)$, subsequent algorithms use images of overlapping regions. Otherwise, subsequent algorithms will use the original image. This method can effectively avoid the impact of inaccurate calculation by phase correlation algorithms. The formula for this method is described as follows:

$$f_s = \begin{cases} f_{ov} & O_{ov} \geq T_p \\ f_{or} & O_{ov} < T_p \end{cases} \tag{10}$$

where, f_s is the images used by subsequent algorithms, f_{ov} represents the images of the overlapping regions, f_{or} represents the original images, T_p is the set threshold.

1) PHASE CORRELATION ALGORITHM

The principle of phase correlation algorithm is as follows. The phase correlation algorithm transforms two images into the frequency domain, and then uses the normalized cross power spectrum to calculate the translation parameters between the images. If there is shift (x_0, y_0) in part of the two images, that is,

$$f_2(x, y) = f_1(x - x_0, y - y_0) \tag{11}$$

Then, use Fourier transform to transform the two images into the frequency domain to obtain $F_1(\mu, v)$ and $F_2(\mu, v)$. At this time, the relationship between images is:

$$F_2(\mu, v) = e^{-j(\mu x_0 - v y_0)} F_1(\mu, v) \tag{12}$$

The cross-power spectra of F_1 and F_2 is:

$$\frac{F_1^*(\mu, v)F_2(\mu, v)}{\left|F_1^*(\mu, v)F_2(\mu, v)\right|} = e^{-j(\mu x_0 - v y_0)} \tag{13}$$

Inverse Fourier transform is performed on the right side of the above formula to obtain the impact response function $\delta(x - x_0, y\text{-}y_0)$, search for the point (x_0, y_0) corresponding to the maximum value of A. (x_0, y_0) is the optimal translation amount between images, thus obtaining the range of overlapping regions.

2) SSIM

The calculation formula for SSIM is as follows:

$$SSIM = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{14}$$

where μ_x and μ_y represent the average grayscale values of the two images, σ_x and σ_y represent the standard deviation of the grayscale values of the two images, σ_{xy} denotes the covariance of the two images, C_1 and C_2 are constants. The larger the SSIM, the greater the similarity of the image and the smaller the difference. The setting of the threshold for SSIM of overlapping regions will be discussed in the next section.

3) TEXTURE CLASSIFICATION BASED ON GRADIENT

Feature points would be extracted in the overlapping regions after obtaining the overlapping regions of the images. However, not all overlapping regions can effectively extract feature points. Typically, feature points are distributed in complex texture regions, while it is difficult to extract feature points in weak texture regions [27], [28], [35], [38], [39]. Therefore, texture classification of images can further reduce the useless computational regions of subsequent algorithms. The current calculation methods for texture classification require separate algorithm design, which adds some additional time overhead. However, the SIFT algorithm calculates corresponding gradient during the calculation process, and gradient can effectively reflect the texture changes of the image. Therefore, we will use gradient for texture classification to reduce unnecessary time overhead. The calculation formula for image gradient is as follows:

$$G = \sqrt{(f(x, y) \otimes S_x)^2 + (f(x, y) \otimes S_y)^2} \tag{15}$$

where, $f(x, y)$ represents the original image, ⊗ represents convolution, S_x and S_y denote the horizontal and vertical directions of the Sobel operator, respectively. The Sobel operator is shown in formulas (16) and (17).

$$S_x = \begin{bmatrix} -2 & 0 & 2 \\ -1 & 0 & 1 \\ -2 & 0 & 2 \end{bmatrix} \tag{16}$$

$$S_y = \begin{bmatrix} -2 & -1 & -2 \\ 0 & 0 & 0 \\ 2 & 1 & 2 \end{bmatrix} \tag{17}$$

The gradient corresponding to figure 6 (a) is shown in figure 6 (b). It can be seen from figure 6 (b) that regions with larger gradients in the image correspond to complex texture regions in figure 6 (a), while regions with smaller gradients correspond to weak texture regions. This indicates that the gradient changes of the image can effectively reflect the changes of image texture. Therefore, the method of texture classification based on gradient is feasible. We divide the image into complex texture and weak texture regions using the threshold T_t. The formula of texture classification based on gradient is described as follows:

$$f_{tc}(x, y) = \begin{cases} 1 & G(x, y) \geq T_t \\ 0 & G(x, y) < T_t \end{cases} \tag{18}$$

where, f_{tc} is the image for texture classification, $G(x, y)$ is the gradient in the corresponding position of the image, and T_t is the set threshold. 1 and 0 represent complex texture regions and weak texture regions, respectively.

The image for texture classification is shown in figure 6 (c) (white represents complex texture regions, and black represents weak texture regions), the result of using threshold directly for texture classification exists independently in pixel units. Due to the fact that the SIFT algorithm needs to detect a region of $3 \times 3 \times 3$ for extreme point detection, directly establishing a Gaussian pyramid based on this classification result

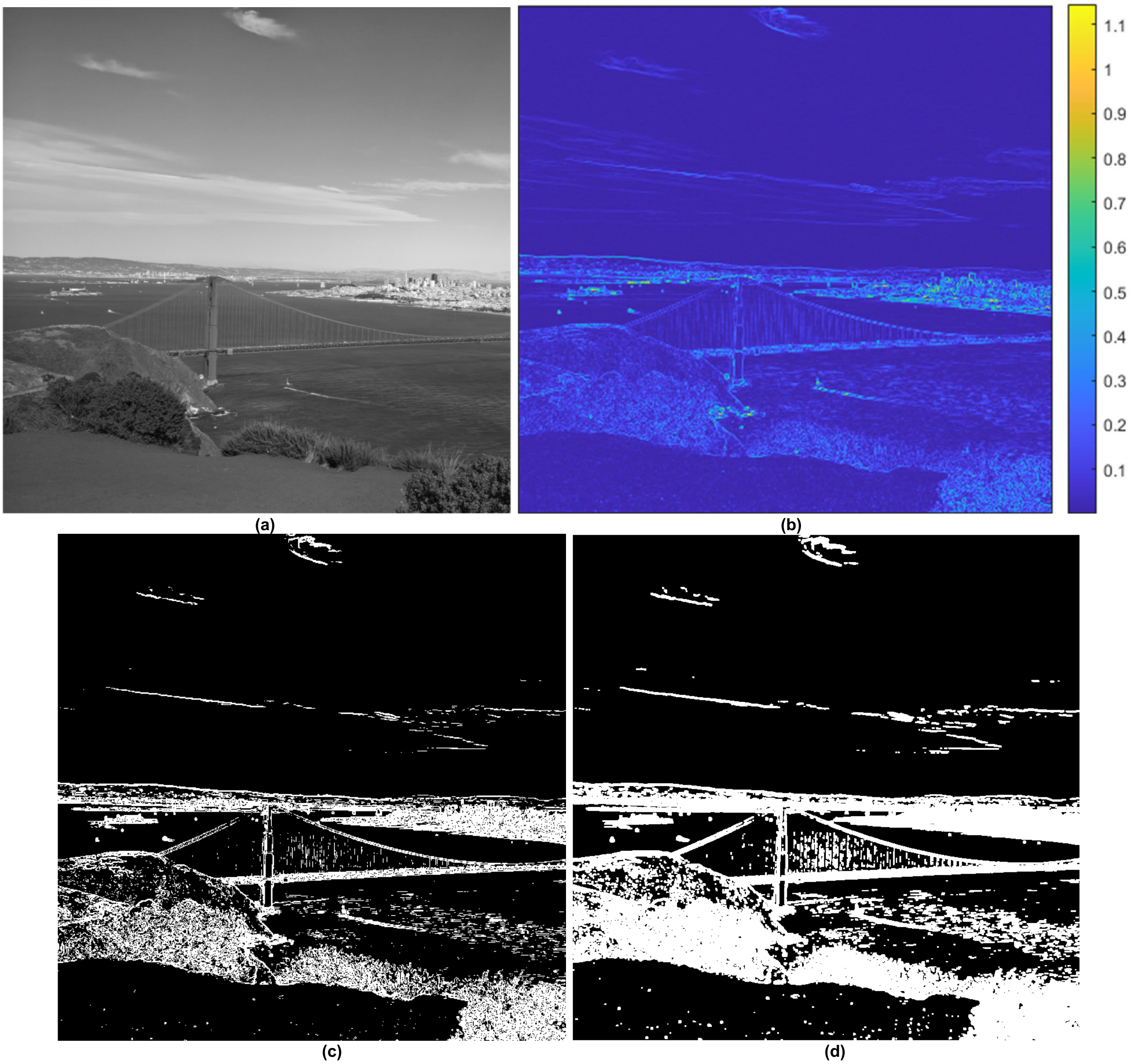


FIGURE 6. Results of texture classification: (a) original image; (b) corresponding gradient image; (c) the results of texture classification; (d) the result of texture classification after dilation operation.

will not be conducive to subsequent extreme point detection. In addition, the methods in references [38], [39] all employ texture classification based on 5 pixels × 5 pixels sub-blocks. Therefore, we apply dilation processing on complex texture regions on the basis of threshold classification. The dilation process is as follows:

$$f \oplus se = \{x : se(x) \cap f \neq \emptyset\} \tag{19}$$

where, f is the dilated image, se is the structural element, and the structural element used is a 5 pixels × 5 pixels square. The result of dilatation operation is shown in figure 6 (d), where there are no isolated points in the dilated region, and it is more continuous overall. After texture classification is completed, feature points will be extracted in complex texture regions. The threshold setting for texture classification will be discussed in the next section.

4) IMPROVED DESCRIPTOR

The corresponding descriptor needs to be calculated after the feature point extraction is completed. The SIFT algorithm needs to calculate the cumulative sum of gradients in 8 directions (45^{o} being one direction) of a 4 × 4 region to obtain a descriptor with 128 dimensions. Descriptor with larger dimensions result in a higher time cost for feature point matching. The calculation range of the descriptor in the SIFT algorithm is a square region. When the SIFT algorithm rotates the calculation region to the main direction of the feature points, there will be a problem of inconsistent calculation range of the descriptor [40]. In response to the above problem, this paper proposes a descriptor with a circular structure (radius of 6σ), as shown in figure 7. And we only calculated the cumulative sum of gradients in 5 directions for each region to further reduce the dimension of the descriptor. The construction method of descriptors is as follows: Establish double concentric circles with feature point as pole point and R_1, R_2 as polar radius, the ratio of R_1 to R_2 is 0.6. Then, the two concentric circles are divided into 4 equal parts to generate 8 regions. Finally, the cumulative sum of gradients in 5 directions for each region is calculated as descriptors for feature points. The number of directions for the cumulative sum of gradients, the ratio of R_1 and R_2, and the radius of the circle will be discussed in the next section.

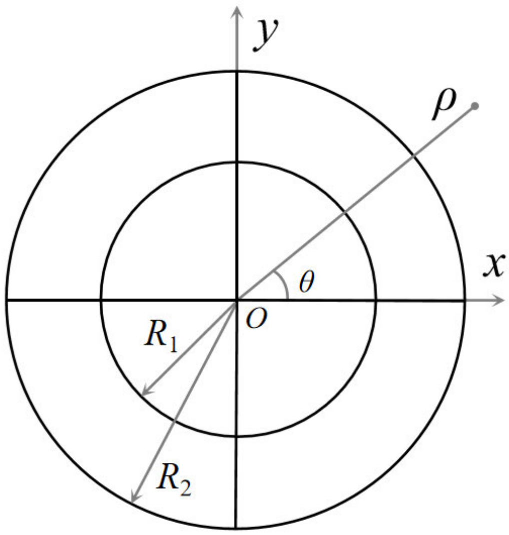


FIGURE 7. The structure of descriptor.

5) TWICE MATCHING METHOD BASED ON EXTREME VALUE CLASSIFICATION

In response to the problem of high time cost in feature point matching stage, reference [35] divides feature points into two categories based on whether they are maxima or minima during feature point extraction, and restricts the matching search space to the same category during the matching stage. Simple labeling achieved good matching accuracy and efficiency. According to formula (9), it is necessary to calculate the distance between the complete descriptors when calculating the distance between two feature points, which inevitably leads to an increase in the time cost of the matching stage.

To further improve the efficiency of the matching stage, we proposed a twice matching method based on reference [35]. The specific method is as follows: firstly, divide the descriptors into two equal parts (corresponding to the inner and outer ring regions of the double ring descriptors), and calculate the distance between the first part for the descriptors of the two feature points. If it is greater than or equal to the set threshold, the difference between the currently calculated feature points is significant, and the current feature points do not match. If the distance is less than the set threshold, calculate the distance between the second part of the descriptors. Then, add up the descriptor distances of the two parts to obtain the final descriptor distance. Finally, determine the matching of feature points based on the nearest neighbor distance ratio (NNDR) and FSC algorithm. The formula for calculating the distance between descriptors is as follows:

$$d_1(i,j) = \sqrt{\sum_{k=1}^{n/2} (x_i(k) - y_j(k))^2} \tag{20}$$

$$d(i,j) = \sqrt{\sum_{k=n/2+1}^{n} (x_i(k) - y_j(k))^2} + d_1(i,j)d_1(i,j) < T_d \tag{21}$$

where, $d_1(i,j)$ represents the distance between the first half of the descriptor, and T_d denotes the set threshold. The setting of thresholds will be discussed in the next section.

IV. DATASETS AND ANALYSIS OF ALGORITHM PARAMETERS

A. DATASETS AND PLATFORM

Dataset 1: The dataset we collected, this dataset contains 150 pairs of images from mobile phones and digital cameras, including scenes of buildings, mountains, cities, rivers, farmland, etc. The images have rigid or affine transformations, and the image sizes range from 1400 pixels × 1700 pixels to 2040 pixels × 2040 pixels.

Dataset 2: This dataset is sourced from the Kaggle website [41]. This dataset contains 1582 pairs of images, including scenes such as buildings, roads, and vehicles. The images have rigid or affine transformations, and the images size are 1920 pixels × 1080 pixels.

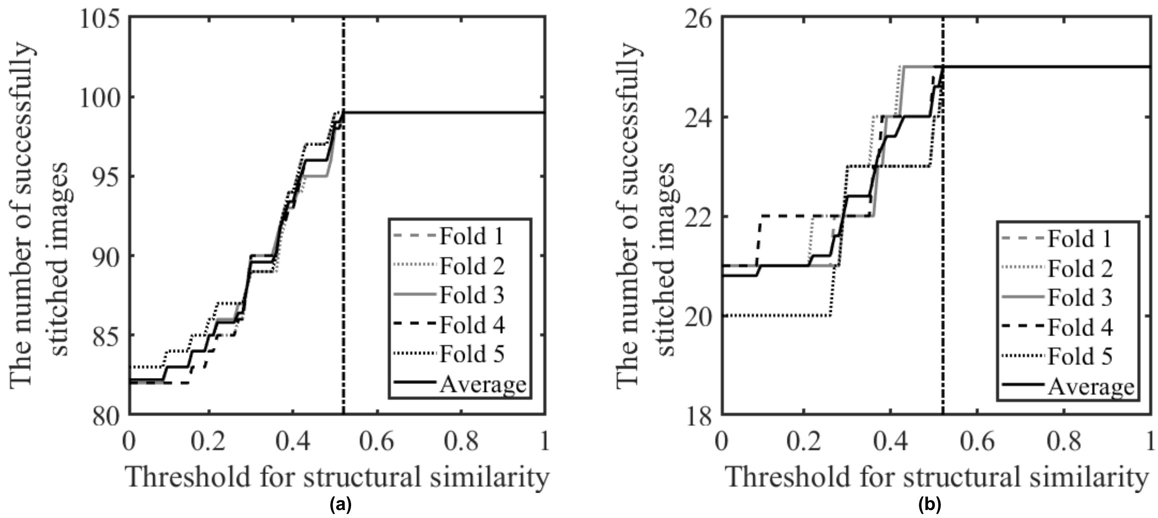
We randomly selected 80% of the images in dataset 1 as the training set, and used 5-fold cross validation method to calculate the relevant parameters in the training set. The remaining 20% of images in dataset 1 and the images in dataset 2 are used as the testing set.

In addition, the implementation platform for the proposed algorithm and all compared algorithms is a 64 bit Windows 11 operating system with a CPU of Intel(R) Core (TM) i7-12700F 2.10 GHz and 16 GB RAM.

B. ANALYSIS OF ALGORITHM PARAMETERS

1) ANALYSIS OF THE THRESHOLD FOR SSIM IN OVERLAPPING REGIONS

In order to obtain the threshold for SSIM in the overlapping regions, different thresholds are set to determine whether subsequent algorithms use images of the overlapping regions (the images of the overlapping regions are used when greater than or equal to the threshold). And calculate the number of successfully stitched images, as shown in figure 8. It can be seen from figure 8 that as the threshold increases, the number



(a)

(b)

FIGURE 8. The effect of the threshold for SSIM on image stitching: (a) training set; (b) validation set.

of successfully stitched images gradually increases on both the training and validation sets. The results calculated by the phase correlation algorithm are inaccurate when the SSIM of the overlapping regions is small, which results in fewer successful image stitching. As the threshold increases, the requirement for SSIM in the overlapping regions of the image gradually increases, which effectively reduce the impact of inaccurate phase correlation algorithm calculations. When the threshold is greater than or equal to 0.52, the number of successfully stitched images remains unchanged. When the SSIM of the overlapping regions is greater than or equal to 0.52, the result calculated using the phase correlation algorithm is accurate. Therefore, we set the threshold for SSIM of the overlapping regions to 0.52.

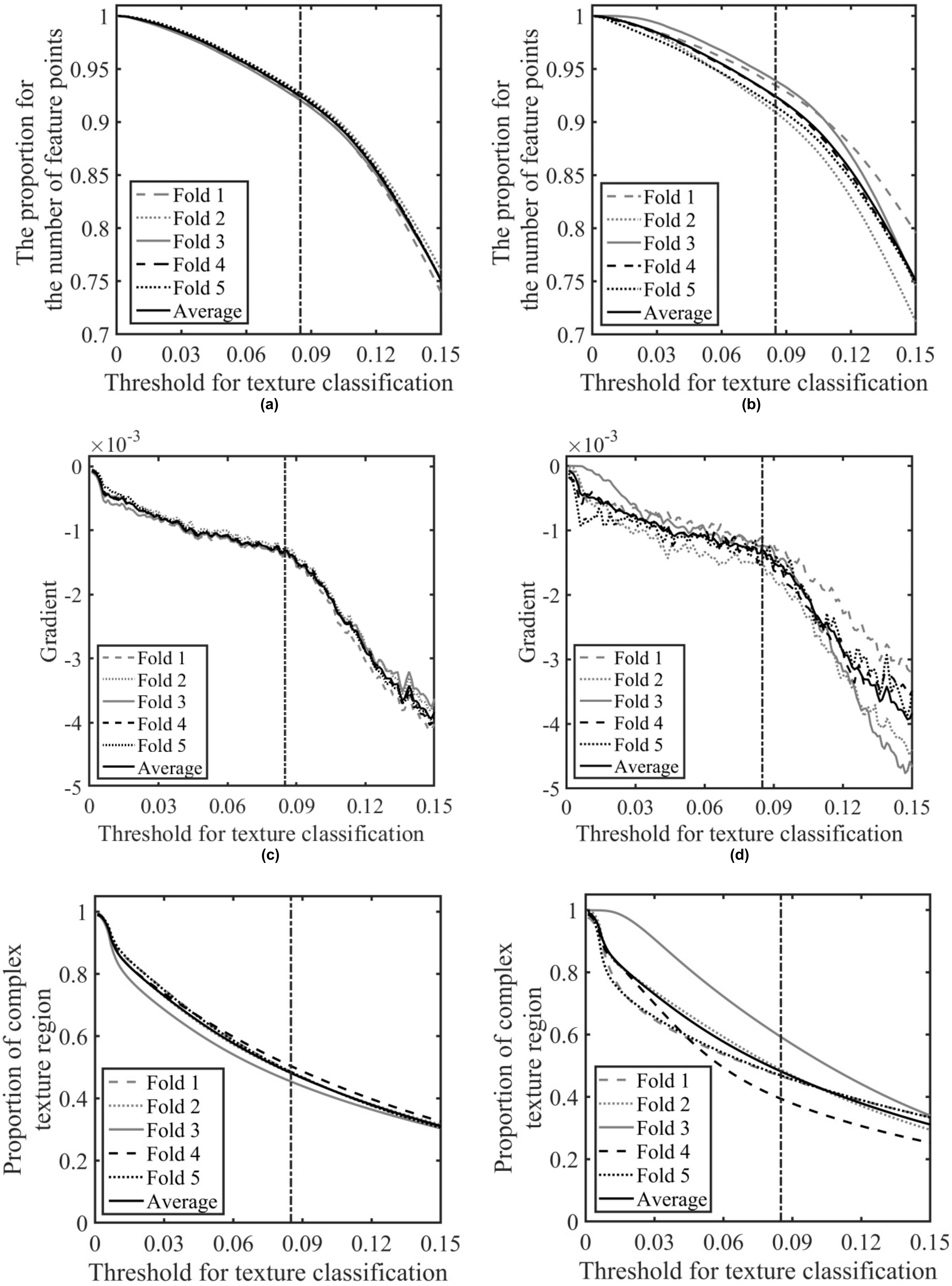
2) ANALYSIS OF THRESHOLD FOR GRADIENT BASED TEXTURE CLASSIFICATION

To obtain the specific threshold for texture classification, different thresholds are set for texture classification after calculating the overlapping regions. And we calculate the proportion of complex texture regions, the proportion of feature points in complex texture regions and the corresponding gradient, as shown in figure 9. According to figures 9 (a) and (b), it can be seen that on both the training and validation sets, the proportion for the number of feature points gradually decreases with the increase of threshold, and the decreasing trend gradually increases. Figures 9 (c) and (d) show the gradient corresponding to figures 9 (a) and (b). It can be seen from figures 9 (c) and (d) that the overall gradient shows a decreasing trend, and the trend gradually increases. The gradient decreases more significantly when the threshold is greater than or equal to 0.085. It can be seen from figures 9 (e) and (f) that the proportion of complex texture regions gradually decreases as the threshold increases, and the decreasing trend gradually slows down. When the threshold is 0.085, the proportions of five cross validations on the training and validation sets are higher than 0.9 in terms for the proportion of feature points, and the average value are higher than 0.92. In terms of the proportion of complex texture regions, all five cross validations were less than 0.5 on the training set. On the validation set, four cross validations were less than 0.5, one cross validation was 0.58, and the average of five cross validations was below 0.5. When the threshold is 0.085, it reduces a few feature points, but also greatly reduces the complex texture regions. For the SIFT algorithm, reducing the number of feature points has minimal impact on image stitching, but reducing the computational region by half has a significant impact on algorithm speed. Therefore, we set the threshold for texture classification to 0.085.

3) ANALYSIS OF DESCRIPTOR STRUCTURE PARAMETERS

The impact of different number of ring divisions and the ratio of inner ring radius (ratio of R_1 to R_2) on the average matching rate was calculated, as shown in figure 10. It can be seen from figure 10 that on both the training and validation sets, the five cross validations and their average values perform poorly when the number of ring divisions is 2 and 3. The results are close and with a small difference when the number of ring divisions reaches 4-8. Therefore, we set the number of ring divisions to 4. Observing the line segments with a number of ring divisions, only the fifth cross validation on the validation set reaches its maximum value when the ratio of R_1 to R_2 is 0.7. The remaining 9 cross validations all reached their maximum value when the ratio of R_1 to R_2 was 0.6, and the average values of the validation set and training set reached their maximum value when the ratio of R_1 to R_2 was 0.6. Therefore, we set the ratio of R_1 to R_2 to 0.6.

After determining the structure of the descriptor, the influence of the number of directions on the matching rate



(a) (b) (c) (d) (e) (f)

FIGURE 9. The effect of gradient threshold on the region of complex texture regions: (a), (c) and (e) training set; (b), (d) and (f) validation set.

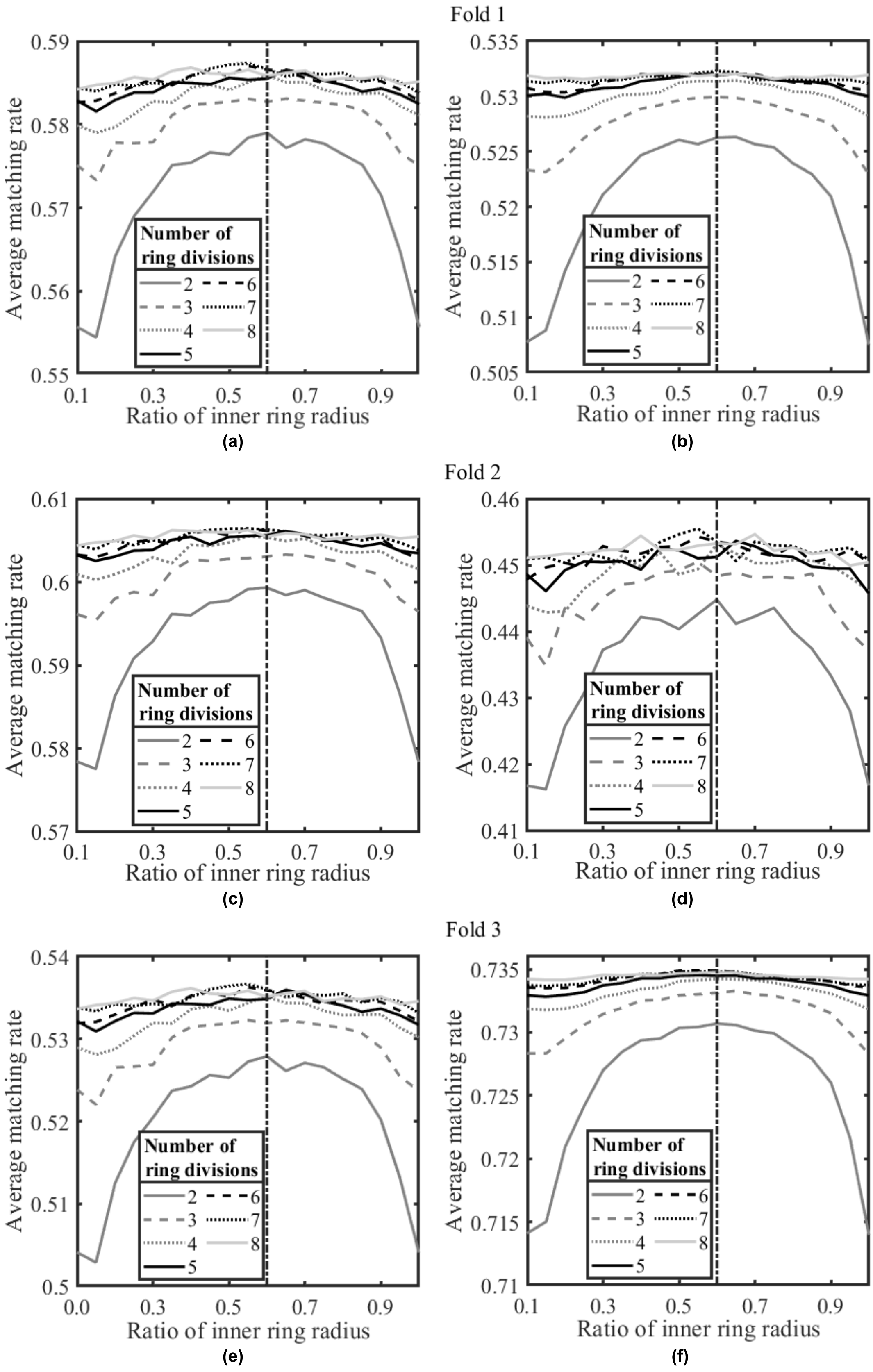
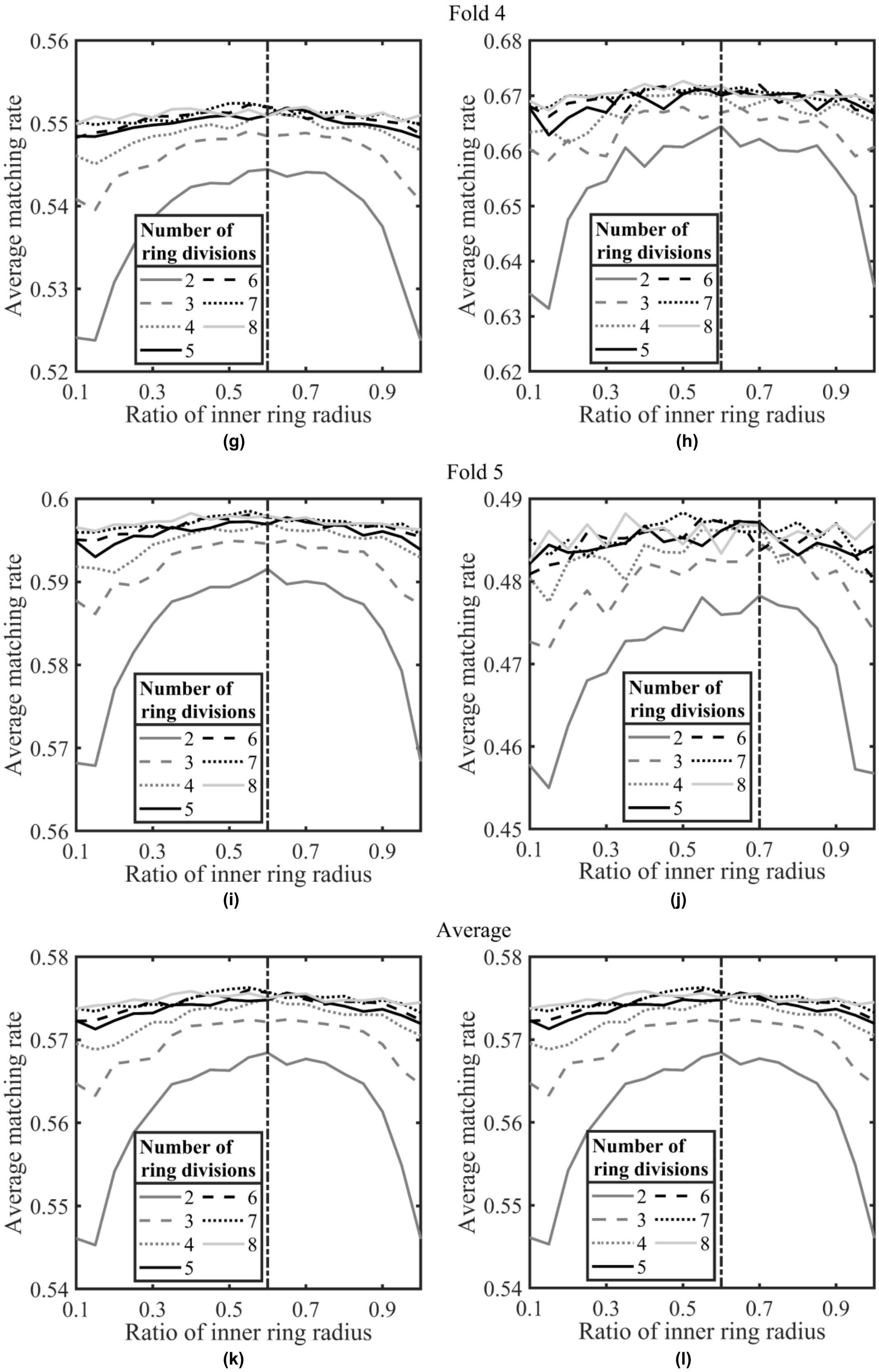


FIGURE 10. The influence of the equal fraction of the ring and the ratio of R_1 and R_2: (a), (c), (e), (g), (i) and (k) training set; (b), (d), (f), (h), (j) and (l) validation set.



(g)

(h)

(i)

(j)

(k)

(l)

FIGURE 10. *(Continued.)* The influence of the equal fraction of the ring and the ratio of R_1 and R_2: (a), (c), (e), (g), (i) and (k) training set; (b), (d), (f), (h), (j) and (l) validation set.

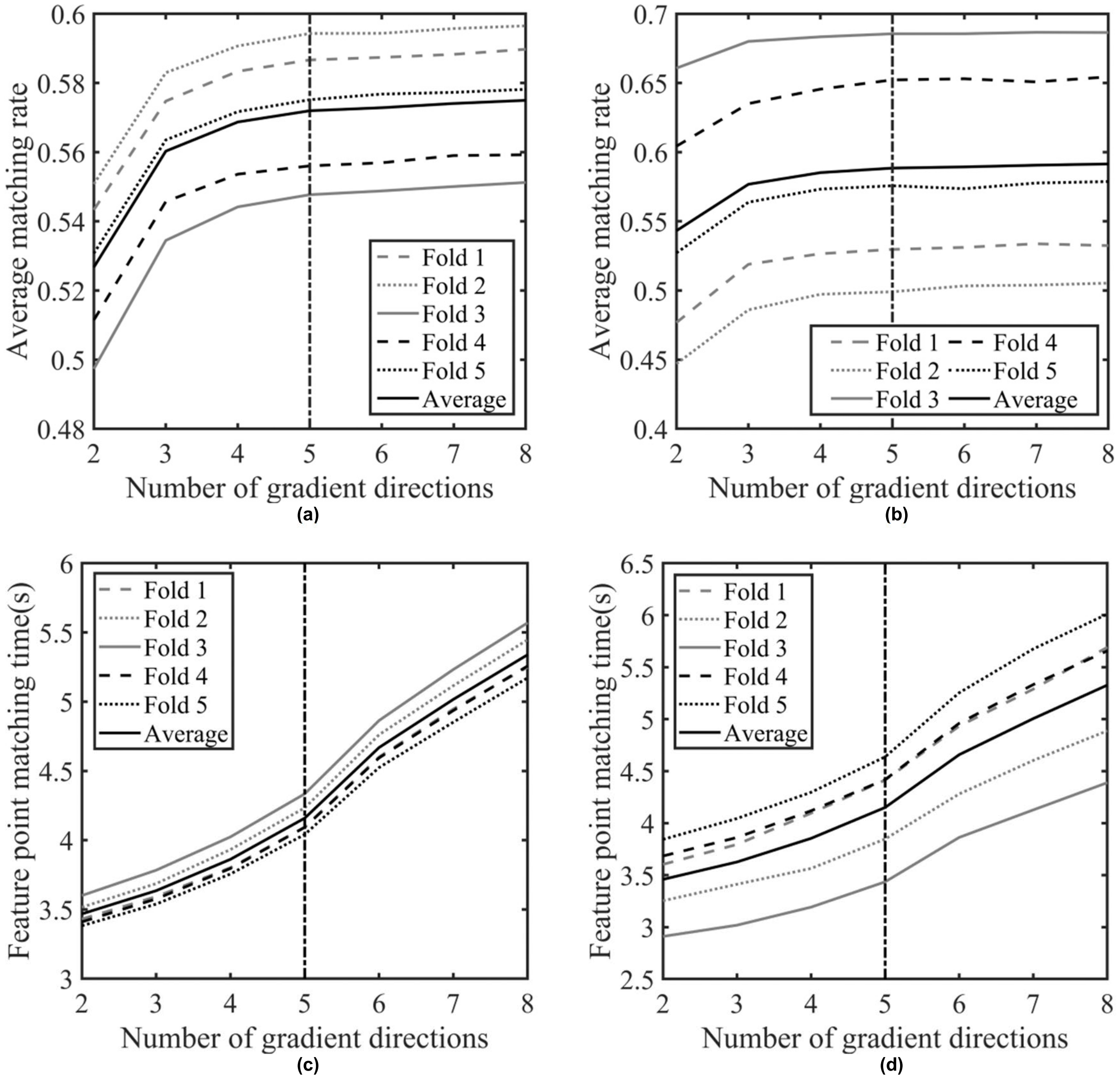
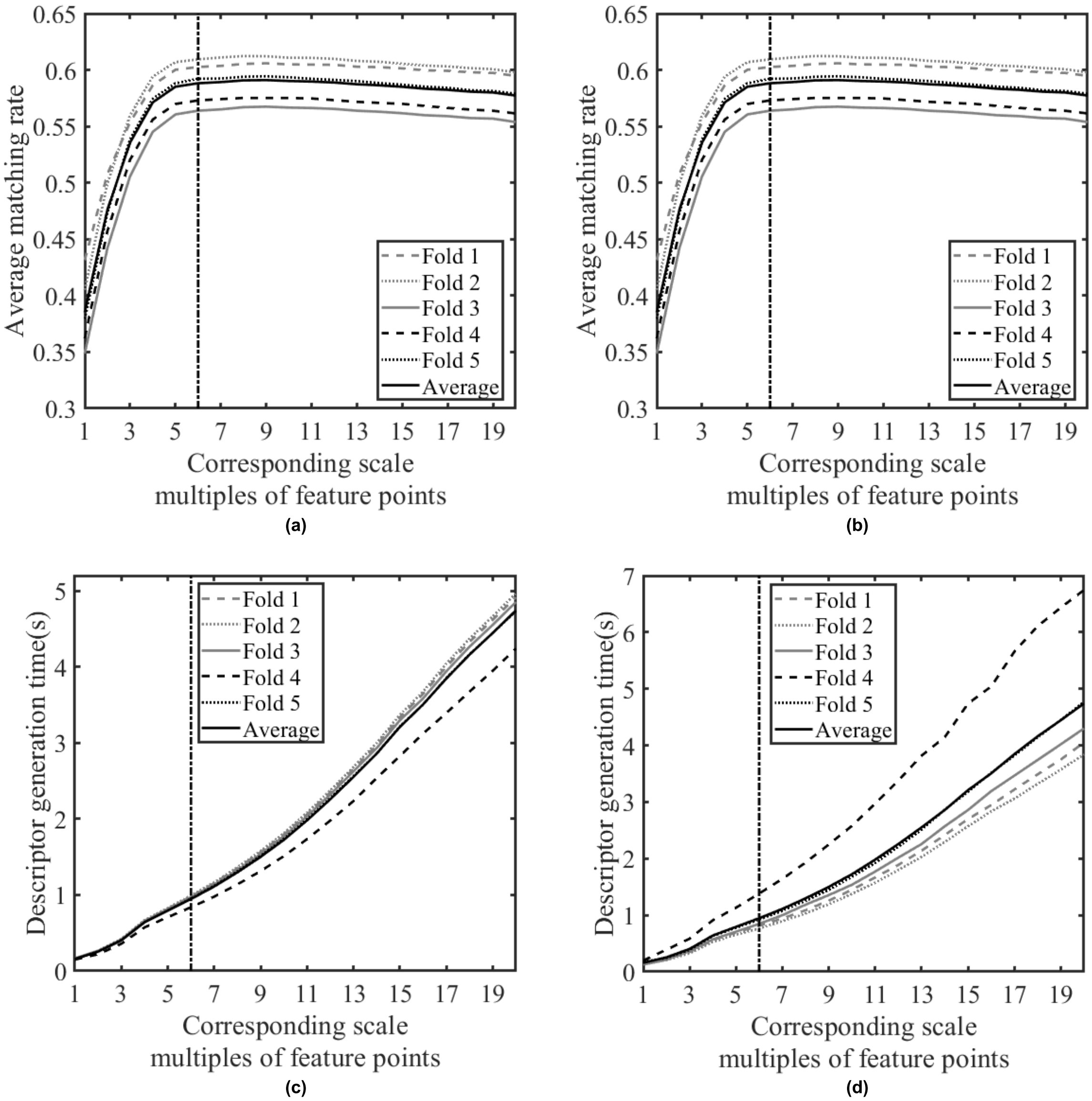


FIGURE 11. The influence of the number of directions in the cumulative sum of statistical gradients: (a) and (c) training set; (b) and (d) validation set.

and feature point matching time was analyzed to obtain the number of directions for the cumulative gradient sum, and the results are shown in figure 11. As shown in figures 11 (a) and (b), the average matching rate of all cross validations shows a trend of first increasing and then stabilizing with the increase of the number of directions, and tends to stabilize after the number of directions reaches 5. It indicates that the improvement in feature point matching rate is relatively small after the number of directions reaches 5. Figures 11 (c) and (d) show that the time for all cross validation increases with the increase of the number of directions. This indicates that the larger the dimension of the feature descriptor, the greater the time cost of feature point matching. Therefore, we set the number of directions for the cumulative gradient sum to 5. At this time, the feature point matching rate is good, and the subsequent increase in the number of directions has a smaller impact on the feature point matching rate, and its time cost is relatively small.

In addition, it is necessary to further determine the calculation range of the descriptor. Figure 12 shows the impact of different corresponding scale multiples of feature points on matching rate and descriptor generation time. It can be seen from figures 12 (a) and (b) that as the calculation range increases, the average matching rate of all cross validations first increases and then decreases. The average matching rate achieves good results when the corresponding scale multiples of feature points reaches 6. Figures 12 (c) and (d) show that as the calculation range increases, the descriptor generation time significantly increases. Therefore, we set the descriptor calculation range to sixfold the corresponding scale of feature points, which has a better matching rate and smaller time cost.



(a)

(b)

(c)

(d)

FIGURE 12. The impact of descriptor calculation range: (a) and (c) training set; (b) and (d) validation set.

4) ANALYSIS OF THRESHOLD FOR TWICE MATCHING METHOD BASED ON EXTREME VALUE CLASSIFICATION

To obtain the threshold for initial matching in twice matching methods, we analyzed the influence of different thresholds on the average matching rate and feature point matching time, and the results are shown in figure 13. It can be seen from figures 13 (a) and (b) that as the threshold increases, the average matching of all cross validations first increases and then stabilizes. When the threshold for initial matching reached 0.33, the average matching rate of all cross validations remained stable, with minimal subsequent changes. This indicates that the impact of the threshold on feature point matching is minimal when the threshold reaches 0.33. Figures 13 (c) and (d) show that as the threshold increases, its time cost gradually increases. It indicates that the larger the threshold, the fewer erroneous matches are removed, resulting in an increase in the number of second matches. In summary, we set the threshold for initial matching to 0.33, which results in a better matching rate and lower time cost.

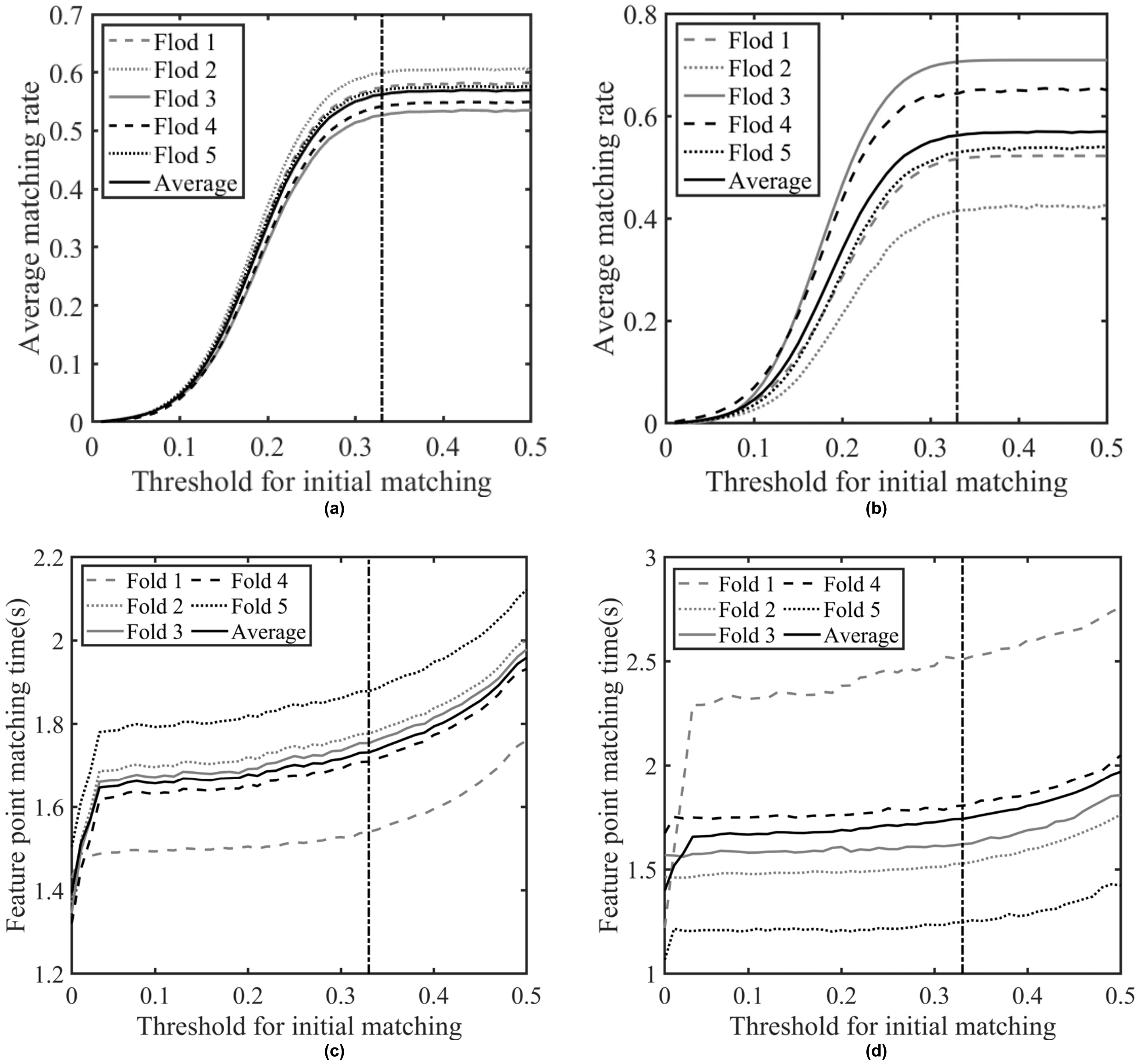
V. EXPERIMENT RESULTS AND DISCUSSION

A. OBJECTIVE EVALUATION INDICATORS

To evaluate the quality of image stitching, we used SSIM, peak signal to noise ratio (PSNR) and root mean squared error (RMSE) [42] for evaluation.

The calculation formula for PSNR is as follows:

$$PSNR = 10\log_{10}(\frac{2^{bits}-1}{MSE}) \tag{22}$$



(a)

(b)

(c)

(d)

FIGURE 13. The influence of different initial matching thresholds: (a) and (c) training set; (b) and (d) validation set.

$$MSE = \frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N}[f_1(x, y) - f_2(x, y)]^2 \tag{23}$$

In formula (22), *bits* represents the bit depth of the pixel value (its value is 8), *MSE* denotes the mean square error of two images, and the calculation formula for *MSE* is shown in formula (23). In formula (23), M and N represent the size of the image, f_1 and f_2 represent two images. The larger the SSIM and PSNR, the greater the similarity of the image and the smaller the difference.

The calculation formula for RMSE is as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\Delta x_i)^2 + (\Delta y_i)^2} \tag{24}$$

where N is the number of evaluated points, and Δx_i and Δy_i are the residual differences of the i-thcheckpoint pair in the x and y directions. The smaller the RMSE, the smaller the error in image registration.

B. ANALYSIS OF PREPROCESSING METHODS

We conducted experiments on the testing set to analyze the effectiveness of the proposed algorithm in the preprocessing stage. We used SIFT algorithm, SIFT + phase correlation algorithm, the algorithm of reference [27] (phase correlation algorithm + 5 × 5 sub-block texture classification), and the proposed algorithm for feature point extraction. Then, we used the descriptor of SIFT algorithm and the NNDR + RANSAC method for feature point matching. It is worth noting that images that fail to be stitched are not

TABLE 2. Comparative analysis of feature point extraction methods in dataset 1.

Method of feature point extraction	Number of failures	Time of preprocessing (s)		Time of feature point extraction (s)		Number of feature points		Matching rate	
		Mean	Std.Dev	Mean	Std.Dev	Mean	Std.Dev	Mean	Std.Dev
SIFT	0	0	0	1.6847	0.2965	5687.18	3201.93	0.3483	**0.2214**
SIFT + phase correlation algorithm	0	**0.1113**	0.0201	1.2037	0.2948	3314.34	1285.31	0.5345	0.3105
Reference [27]	0	0.2375	0.0418	1.1109	0.2440	3189.82	1289.50	0.5323	0.3090
The proposed	0	0.1567	**0.0197**	**1.0910**	**0.2164**	**3053.03**	**1245.01**	**0.5364**	0.3022

TABLE 3. Comparative analysis of feature point extraction methods in dataset 2.

Method of feature point extraction	Number of failures	Time of preprocessing (s)		Time of feature point extraction (s)		Number of feature points		Matching rate	
		Mean	Std.Dev	Mean	Std.Dev	Mean	Std.Dev	Mean	Std.Dev
SIFT	**1053**	0	0	1.4666	**0.1706**	5690.92	2178.53	0.0366	**0.0396**
SIFT + phase correlation algorithm	1092	**0.0900**	0.0109	1.3147	0.1763	5082.74	1933.94	0.0407	0.0470
Reference [27]	1092	0.2249	0.0218	1.4293	0.1912	4952.94	1933.57	**0.0408**	0.0454
The proposed	**1053**	0.1290	**0.0106**	**1.2430**	0.1762	**4733.75**	**1901.63**	0.0402	0.0464

TABLE 4. Comparison of descriptors in dataset 1.

Descriptor	Number of failures	Time of descriptor generation (s)		Time of feature point matching (s)		Matching rate		SSIM		PSNR	
		Mean	Std.Dev	Mean	Std.Dev	Mean	Std.Dev	Mean	Std.Dev	Mean	Std.Dev
SIFT	0	**0.3278**	**0.1207**	2.6719	1.5708	**0.5364**	**0.3022**	0.8389	0.3074	29.4208	14.2566
Reference [28]	0	1.7938	0.6777	**1.3128**	**0.7911**	0.4496	0.3438	0.7727	0.3984	27.8071	16.8186
Reference [35]	0	0.4298	0.1551	1.9754	1.1050	0.5257	0.3065	0.8677	0.2716	**30.8118**	**13.2054**
Reference [36]	0	1.0326	0.3720	1.8913	1.1683	0.5167	0.3159	0.8669	**0.2708**	30.6076	13.3535
The proposed	0	0.4298	0.1555	1.6733	0.9088	0.5188	0.3112	**0.8678**	0.2711	30.8048	13.2953

TABLE 5. Comparison of descriptors in dataset 2.

Descriptor	Number of failures	Time of descriptor generation (s)		Time of feature point matching (s)		Matching rate		SSIM		PSNR	
		Mean	Std.Dev	Mean	Std.Dev	Mean	Std.Dev	Mean	Std.Dev	Mean	Std.Dev
SIFT	1053	0.4738	**0.1843**	5.1405	3.6574	**0.0402**	0.0464	0.4684	0.1719	14.0040	3.5289
Reference [28]	1303	2.3496	0.8771	**2.0579**	**1.1271**	0.0360	**0.0316**	**0.5133**	0.1557	**14.7830**	**2.9973**
Reference [35]	739	0.6837	0.2578	3.6996	2.2478	0.0294	0.0352	0.4492	0.1605	13.7212	3.3520
Reference [36]	720	1.6746	0.6615	3.8396	2.4426	0.0311	0.0374	0.4570	0.1578	13.8720	3.2598
The proposed	**710**	**0.6784**	0.2527	3.0453	1.7640	0.0282	0.0321	0.4515	**0.1515**	13.8854	3.0411

subjected to statistical calculations, and the results are shown in tables 2 and 3.

It can be seen from tables 2 and 3 that the proposed algorithm and SIFT algorithm have the least number of

TABLE 6. Comparison of matching methods in dataset 1.

Method of feature point matching	Number of failures	Time of feature point extraction (s)		Time of feature point matching (s)		Matching rate		SSIM		PSNR		RMSE	
		Mean	Std.Dev	Mean	Std.Dev	Mean	Std.Dev	Mean	Std.Dev	Mean	Std.Dev	Mean	Std.Dev
NNDR + RANSAC	0	**1.0910**	**0.2164**	1.6733	0.9088	**0.5188**	0.3112	0.8678	0.2711	30.8048	13.2953	5.3084	8.5985
NNDR + FSC	0	**1.0910**	**0.2164**	1.8509	0.9952	0.3824	0.2447	0.9037	0.2130	32.2859	12.7290	0.3065	0.1919
Reference [35]	0	1.0952	0.2285	1.5464	0.7540	0.3965	0.2391	**0.9039**	**0.2118**	**32.4081**	**12.3601**	**0.2812**	0.1876
Proposed algorithm + RANSAC	0	1.0952	0.2285	**1.2497**	0.6722	0.5052	0.3011	0.8718	0.2668	30.5393	13.6002	5.4954	8.6169
Proposed algorithm + FSC	0	1.0952	0.2285	1.3208	**0.6321**	0.3931	**0.2328**	0.9002	0.2170	32.4022	12.4727	0.2946	**0.1875**

TABLE 7. Comparison of matching methods in dataset 2.

Method of feature point matching	Number of failures	Time of feature point extraction (s)		Time of feature point matching (s)		Matching rate		SSIM		PSNR		RMSE	
		Mean	Std.Dev	Mean	Std.Dev	Mean	Std.Dev	Mean	Std.Dev	Mean	Std.Dev	Mean	Std.Dev
NNDR+RANSAC	710	**1.2430**	**0.1685**	3.0453	1.7640	**0.0282**	0.0321	0.4515	0.1515	13.8854	3.0411	2.8788	2.8700
NNDR+FSC	64	**1.2430**	**0.1685**	6.6919	4.0308	0.0110	0.0167	0.4401	0.1491	13.9453	2.3422	0.4930	**0.1265**
Reference [35]	75	1.2465	0.1762	2.7479	**1.6296**	0.0105	0.0163	0.4486	0.1536	14.0116	2.5172	0.4925	0.2437
Proposed algorithm + RANSAC	708	1.2465	0.1762	**2.6075**	2.1592	0.0278	0.0300	**0.4550**	0.1483	**14.0150**	3.0996	2.7331	2.7902
Proposed algorithm + FSC	**70**	1.2465	0.1762	2.6868	1.6358	0.0106	**0.0163**	0.4452	**0.1474**	13.9857	**2.2803**	**0.4828**	0.1545

failures, indicating that the proposed algorithm effectively avoids the impact of inaccurate phase correlation calculation by calculating the SSIM of the overlapping regions. In terms of preprocessing time, the average values of the algorithms proposed on the two datasets increased by 0.0454 seconds and 0.0390 seconds respectively compared to the SIFT + phase correlation algorithm. The proposed algorithm only takes 0.0454 seconds and 0.0390 seconds to complete texture classification, which is only 35.97% and 28.91% of the algorithms of reference [27]. This indicates that the proposed algorithm effectively utilizes the gradients in the SIFT algorithm calculation process, reducing the time cost of texture classification. Meanwhile, the proposed algorithm has the minimum standard deviation, indicating that the time required for classifying different image textures is relatively stable. The proposed algorithm has the minimum mean and standard deviation in terms of feature point extraction time. It is proved that the proposed algorithm effectively reduces the computational region of the SIFT algorithm and has good stability. In terms of the average number of feature points, the SIFT algorithm has the highest number, followed by the SIFT + phase correlation algorithm. It is indicated that the phase correlation algorithm effectively reduces the number of feature points in non-overlapping regions. The algorithm of reference [27] and the proposed algorithm have fewer feature points, indicating that although the texture classification method reduces the region calculated by the SIFT algorithm. However, the slightly reduces the number of extractable feature points. Combining the two datasets, the algorithm proposed has the best performance in terms of average matching rate. This indicates that the proposed algorithm extracts more accurate and stable feature points. In terms of its standard deviation, the SIFT algorithm performs better, while the proposed algorithm has relatively large numerical values. However, the proposed algorithm is significantly better than the SIFT algorithm in terms of mean, so the feature point matching rate extracted by the proposed algorithm performs the best. In summary, in the preprocessing stage, the proposed algorithm effectively utilizes the gradients of the image, completing texture classification in only 0.0454 seconds and 0.0390 seconds. The feature point extraction time is shorter, and the extracted feature points have the best matching rate. This indicates that the proposed algorithm is more accurate, fast and effective.

C. ANALYSIS OF DESCRIPTORS

We conducted experiments on the testing set to analyze the effectiveness of the proposed descriptor. On the basis of using the proposed algorithm to extract feature points, the

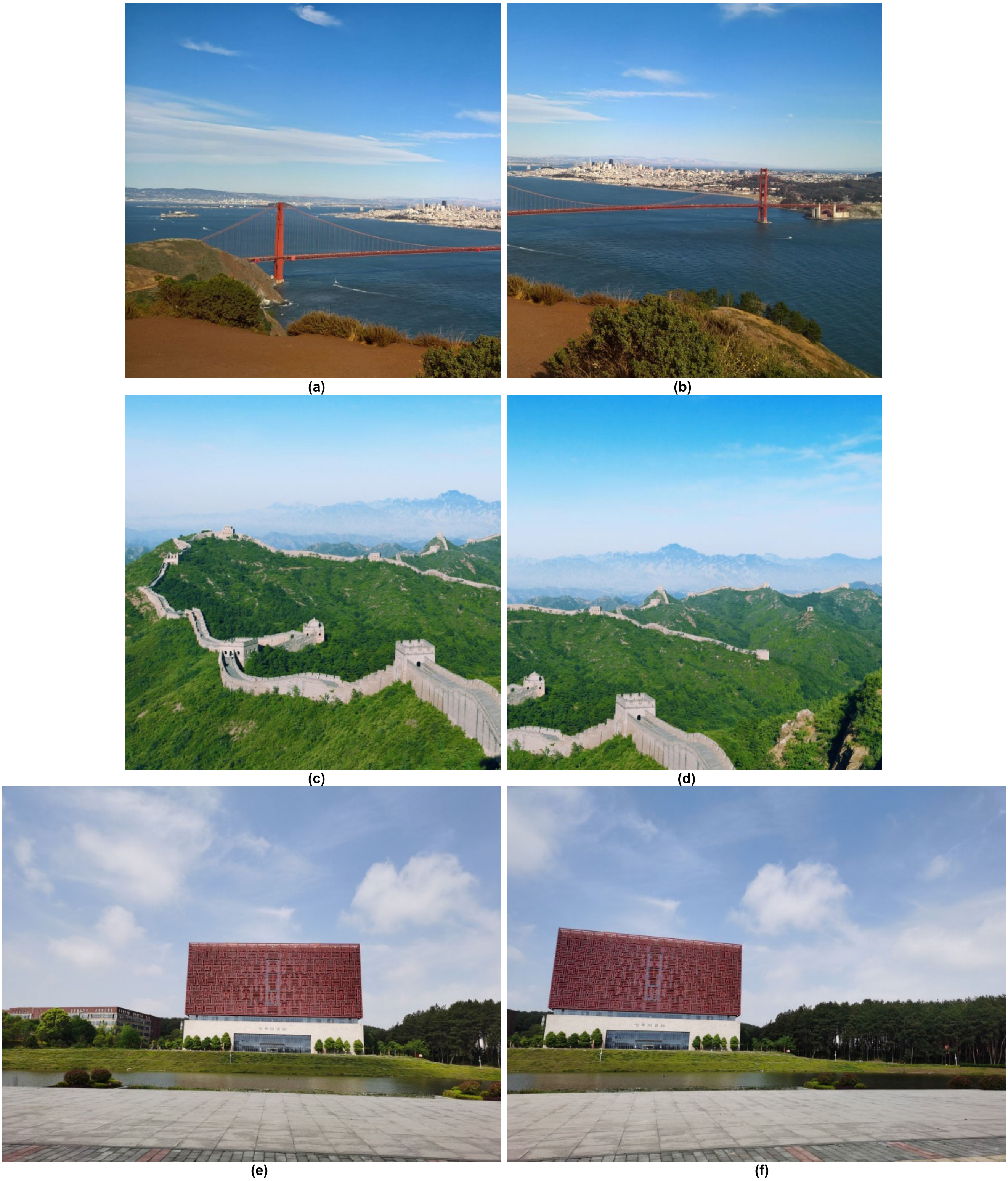


FIGURE 14. Images to be stitched: (a) and (b) bridge; (c) and (d) mountain range;(e) and (f) building.

descriptors of SIFT, reference [28], [35], [36] and the proposed algorithm were used for descriptor generation. Then, the NNDR + RANSAC method was used for feature point matching. Similarly, images that failed to be stitched were not subjected to statistical calculations, and the results are shown in tables 4 and 5.

TABLE 8. Comparison of algorithm times.

Image	Algorithm	Number of feature points	Time of preprocessing (s)	Time of feature point extraction (s)	Time of descriptor generation (s)	Time of feature point matching (s)	Total time (s)
Figures 14(a) and (b)	SIFT	3788,6019		1.9403	0.5579	4.4485	6.9467
	Reference [27]	2099,2398	0.2095	0.8842	0.2076	1.3412	2.6425
	Reference [28]	3390,5314	0.1799	1.8229	1.9322	1.9004	5.8354
	Reference [33]	1117,1826		1.3468	**0.1702**	1.5437	3.0607
	Reference [35]	2802,4478	0.1399	1.6369	0.4332	1.9669	4.1769
	Reference [36]	**872,921**	**0.0972**	3.3057	0.2932	**0.5081**	4.2042
	The proposed	2062,2368	0.1566	**0.7974**	0.2417	0.8023	**1.9980**
Figures 14(c) and (d)	SIFT	12659,8249		2.5065	1.1109	18.2547	21.8721
	Reference [27]	6356,3066	0.2124	1.0117	0.4393	4.1632	5.8266
	Reference [28]	10910,7267	0.1678	2.0804	3.9815	6.7461	12.9758
	Reference [33]	3383,2406		1.3726	**0.3006**	4.6438	6.3170
	Reference [35]	8692,6357	0.1446	1.9834	0.8822	7.1129	10.1231
	Reference [36]	**958,936**	**0.1006**	2.4712	0.3394	**0.5437**	3.4549
	The proposed	6192,2945	0.1603	**0.9691**	0.5189	1.7709	**3.4192**
Figures 14(e) and (f)	SIFT	3719,3602		1.3753	0.3999	3.0343	4.8095
	Reference [27]	2762,2779	0.2135	1.1060	0.2858	2.0383	3.6436
	Reference [28]	2091,1867	0.1478	1.5140	0.9027	0.8216	3.3861
	Reference [33]	1456,1268		1.1330	**0.1567**	1.6002	2.8899
	Reference [35]	2093,1956	0.1212	1.2847	0.2348	1.0809	2.7216
	Reference [36]	**887,879**	**0.0822**	5.6209	0.3113	**0.6219**	6.6363
	The proposed	2684,2686	0.1542	**1.0106**	0.2947	1.0348	**2.4943**

According to tables 4 and 5, the proposed descriptor has the least number of failures, indicating that it can better describe image information. In terms of the time for descriptor generation, the generation time is longer due to the fact that all descriptors except SIFT are circular descriptors. The descriptor generation time of references [28] and [36] is longer due to the larger calculation range and construction method of more complex. The generation time of reference [35] and the proposed descriptor is close, with both having shorter descriptor generation times. In terms of the standard deviation for descriptor generation time, the small standard deviation of the proposed algorithm indicates that the fluctuation of descriptor generation time for different images is relatively small. In terms of feature point matching time, the mean and standard deviation are the smallest for descriptor of reference [28] due to the smallest dimension. The proposed descriptor is only slightly higher than that of reference [28]. Combining the times of descriptor generation and feature point matching, the proposed descriptor has the least comprehensive time. This indicates that the proposed descriptor is structurally simple and has a smaller dimension. Combining the matching rate, SSIM, PSNR, and the number of failures. The descriptor of SIFT algorithm has the best matching rate, but it has a higher number of failures, poorer SSIM and PSNR. This indicates that there are many incorrect matches in the descriptor of the SIFT algorithm, resulting in an increase in the number of failures and a decrease in image quality. The proposed descriptor has good matching rate, SSIM and PSNR, as well as the lowest number of failures. This indicates that the proposed descriptor can accurately describe image information and avoid incorrect matching. In summary, the proposed descriptor has a simple construction and small dimensions. It can accurately describe the information of images, which is more conducive to feature

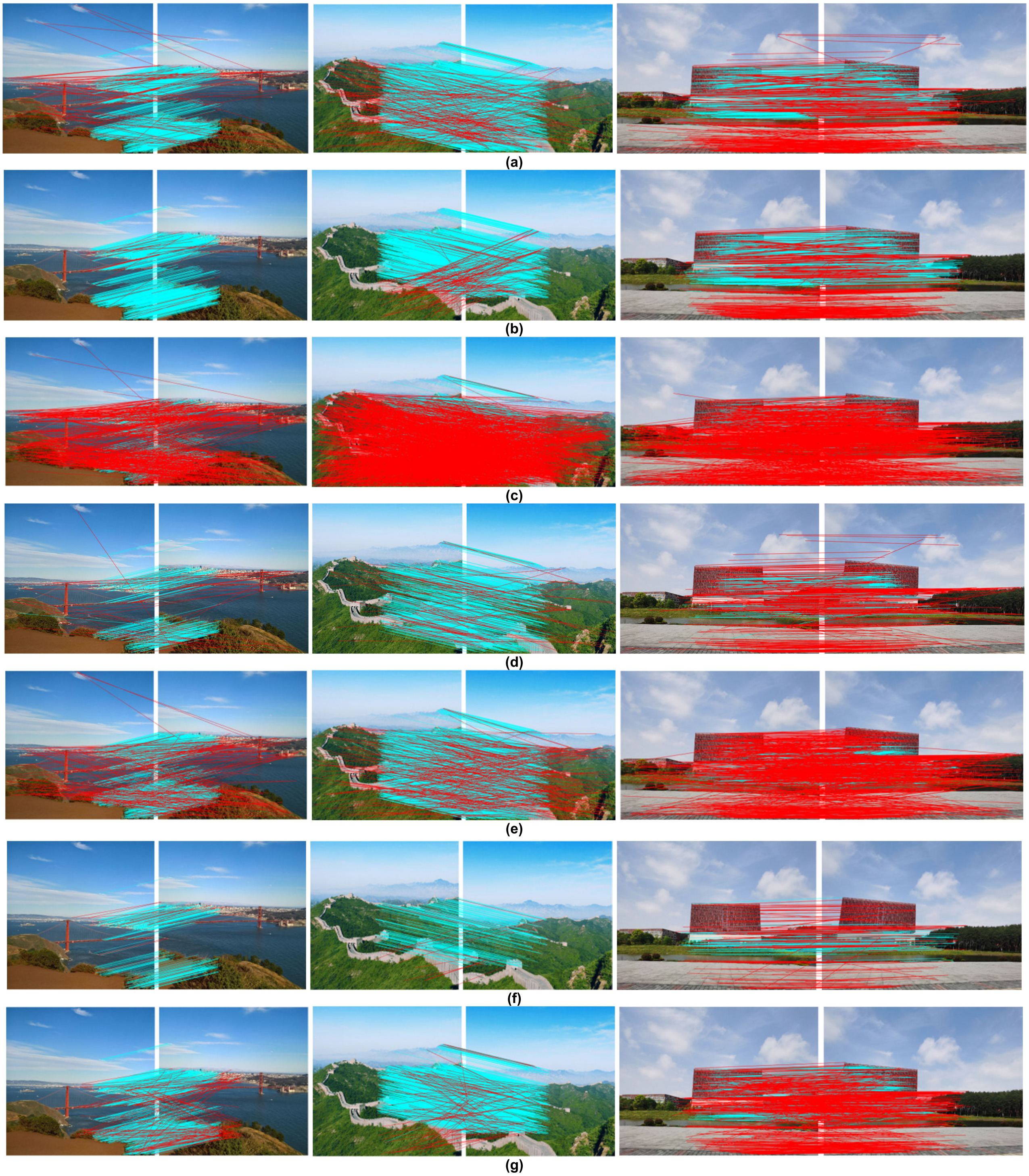


FIGURE 15. The result of feature point matching (The red line indicates incorrect matches, while the blue line indicates correct matches): (a) SIFT; (b) reference [27];(c) reference [28]; (d) reference [33]; (e) reference [35]; (f) reference [36]; (g) Proposed algorithm.

point matching and has good matching performance and efficiency.

D. ANALYSIS OF MATCHING METHODS

On the basis of extracting feature points and generating descriptors using the proposed algorithm, feature points matching was performed using NNDR + RANSAC, NNDR + FSC, reference [35] (extreme value classification + FSC), proposed algorithm + RANSAC, and proposed algorithm + FSC, respectively. Similarly, images that failed to be stitched were not subjected to statistical calculations, the experimental results of the testing set are shown in tables 6 and 7.

Tables 6 and 7 shows that the NNDR + FSC algorithm has a longer feature point matching time and lower matching

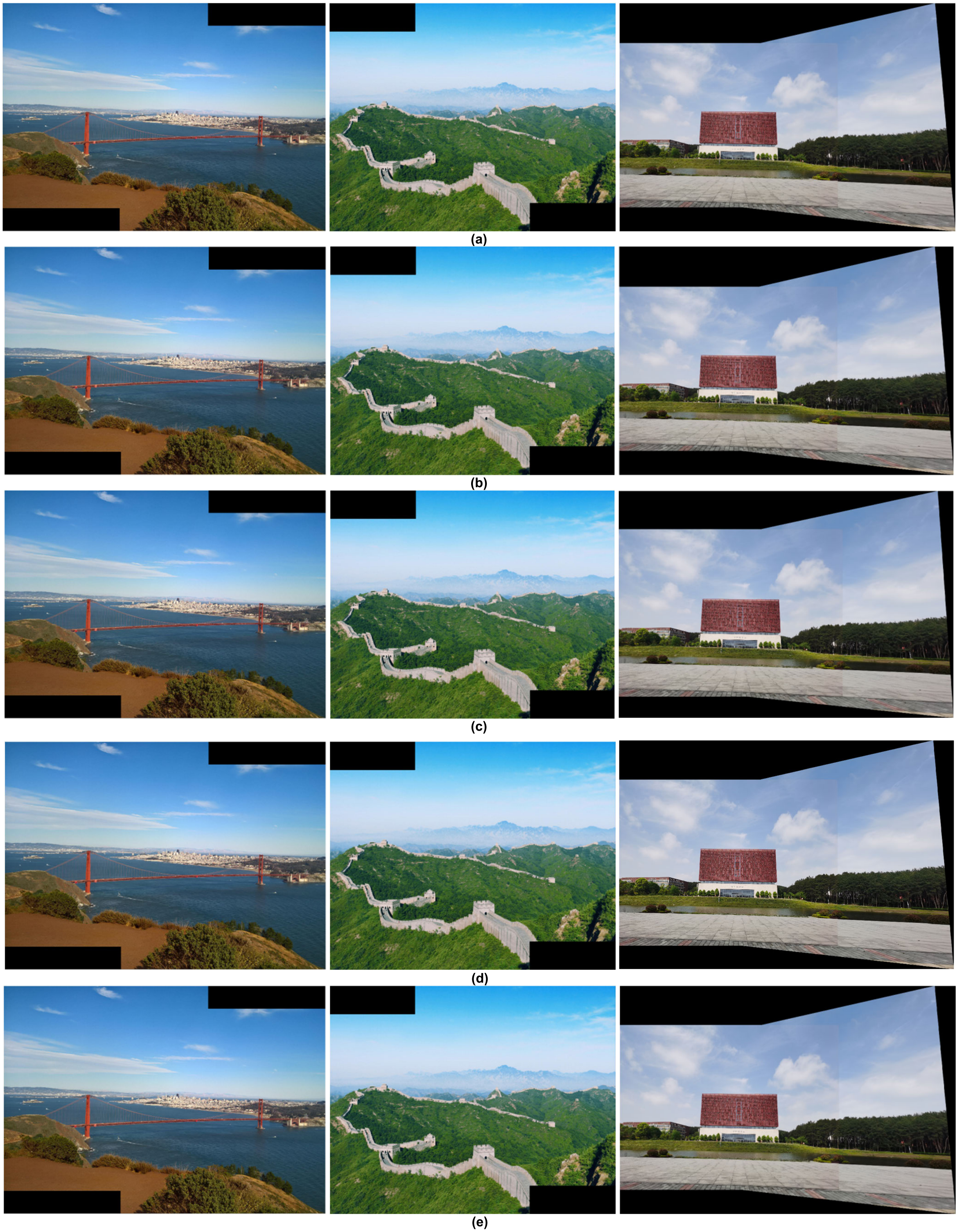
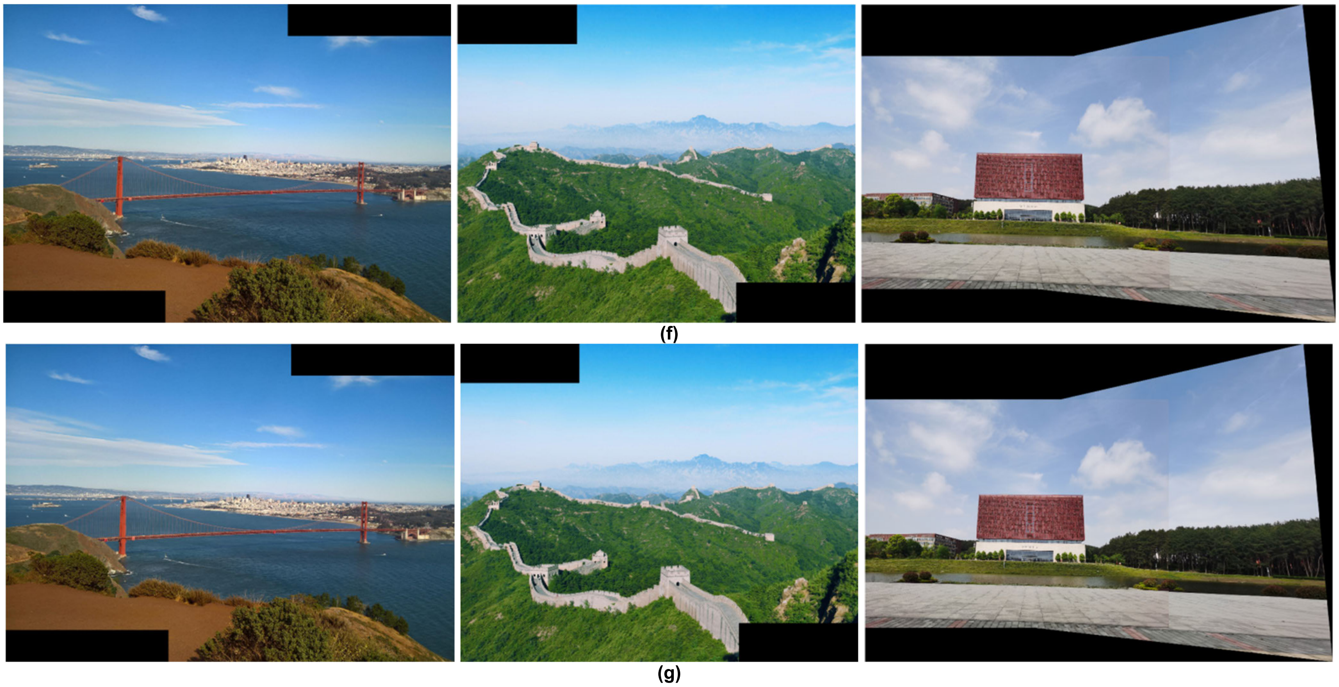


FIGURE 16. **Results of image stitching: (a) SIFT; (b) reference [27];(c) reference [28]; (d) reference [33]; (e) reference [35]; (f) reference [36]; (g) Proposed algorithm.**


(f)

(g)

FIGURE 16. ***(Continued.)*** **Results of image stitching: (a) SIFT; (b) reference [27]; (c) reference [28]; (d) reference [33]; (e) reference [35]; (f) reference [36]; (g) Proposed algorithm.**

rate compared to the NNDR + RANSAC algorithm. But the performances of the number of failures, SSIM, PSNR, and RMSE are significantly better. This indicates that FSC algorithm has a higher time cost, but it can extract more accurate feature point matching results, thereby improving the quality of image stitching. Compared to the NNDR + FSC algorithm, the algorithm of reference [35] needs to classify the feature points in the feature point extraction stage. Therefore, there will be additional time overhead, but it only increases the time by 0.0035-0.0042 seconds. In terms of feature point matching time, the algorithm in reference [35] has significantly lower time overhead than the NNDR + FSC algorithm, which indicated that the simple extreme value classification can effectively reduce the time for feature point matching. In addition, in terms of matching rate, SSIM, PSNR and RMSE, the algorithm of reference [35] has similar performance to NNRD + FSC, indicating that the algorithm of reference [35] also has better stitching quality. Therefore, the algorithm of reference [35] has faster efficiency and better image stitching quality. Due to the fact that the proposed algorithm is also based on extreme value classification, it is consistent with the algorithm of reference [35] in terms of feature point extraction time. The proposed algorithm + RANSAC has the minimum time cost, indicating that the proposed twice matching method based on extreme value classification can effectively reduce the time cost of feature point matching stage. The matching rate of SSIM, PSNR and RMSE is similar to NNDR + RANSAC. Therefore, the proposed algorithm + RANSAC can effectively reduce the time cost of feature point matching stage and has good stitching quality. The time cost of the proposed algorithm + FSC is slightly higher than that of the proposed algorithm + RANSAC, indicating that this method also effectively reduces the time cost of the feature point matching stage. The matching rate of SSIM, PSNR and RMSE is close to reference [35], proving that this method can achieve more accurate feature point matching and effectively improve the quality of image stitching. The proposed algorithm + FSC has better stitching quality and is close in time compared to the proposed algorithm + RANSAC. The proposed algorithm + FSC can effectively reduce the time cost of feature point matching stage. Meanwhile, it can match feature points more accurately and has better stitching quality.

E. ANALYSIS OF ALGORITHM TIME

We selected three representative pairs of images from the testing set, as shown in figure 14. The sizes of figures 14 (a), (b), (c), and (d) are 1750 pixels × 1750 pixels. The sizes of figures 14 (e) and (f) are 1368 pixels × 1824 pixels.

For the image in figure 14, we used SIFT algorithm, the proposed algorithm, the algorithms of references [27], [28], [33], [35], and [36] for image stitching. And the time for each stage were calculated, as shown in table 8.

It can be seen from table 8 that the time cost of feature point matching is highest because of the SIFT algorithm has a huge number of feature points, thereby resulting in the highest overall time cost. The algorithm of reference [27] effectively reduced the time of feature point extraction and number of feature points using preprocessing methods, which results in relatively smaller time costs for subsequent stages and total time. The algorithm of reference [28] reduces the time of feature point extraction through preprocessing. Although its descriptor calculation is relatively complex, its dimensions are small. Therefore, its descriptor generation time is longer, but the feature point matching time is smaller. The total time is relatively small. The algorithm of reference [33] reduces the number of feature points by restricting their extraction and optimizes the matching stage, thereby reducing the time of the three stages of the algorithm. The algorithm of reference [35] reduces the computational region of the SIFT algorithm and number of feature points through preprocessing, and its descriptor dimension is smaller. Therefore, the time cost for the subsequent three stages is relatively smaller, and the overall speed is fast. The algorithm of reference [36] obtained the minimum number of feature points through preprocessing and special feature point extraction methods. The time of feature point extraction is longer because the feature point extraction method is more complex. However, due to its minimum number of feature points and smaller descriptor dimensions, the times of descriptor generation and feature point matching are very small, resulting in a smaller total time. The proposed algorithm effectively reduces the computational region of the SIFT algorithm through phase correlation and texture classification. The time of feature point extraction is minimized. In the stage of descriptor generation, the time of descriptor generation is relatively short due to the small number of feature points and the relatively simple calculation of descriptors. In the feature point matching stage, the time cost of the feature point matching stage is relatively small due to its smaller descriptor dimensions and faster efficiency of the twice matching method based on extreme value classification. In terms of total time, the proposed algorithm is only 28.76%, 15.63%, and 51.86% of SIFT, respectively. And the proposed algorithm has the best performance compared to other algorithms.

F. ANALYSIS OF IMAGE STITCHING QUALITY

The results of feature point matching and stitching using the above algorithm are shown in figures 15 and 16. As shown in figure 15, the algorithm of reference [28] has the highest number of mismatches due to its poor descriptor performance. Due to the fact that the algorithm of reference [36] has the fewest number of feature points, it also has the fewest number of mismatches. Both the SIFT algorithm and the algorithm in reference [27] have a relatively small number of mismatches. Because the FSC algorithm can more accurately remove the mismatched feature points of the NNRD algorithm, the algorithm of reference [35] has relatively larger number of mismatches. Although the algorithm of reference [33] also uses the FSC algorithm, due to its fewer feature points, the number of mismatches is relatively small. Compared to the algorithm of reference [35], the proposed algorithm also uses the FSC algorithm, but it has fewer number of mismatches. This indicates that the proposed algorithm extracts more stable feature points and the proposed descriptor has better performance.

TABLE 9. Comparison of algorithm stitching quality.

Images	Algorithm	Matching rate	SSIM	PSNR	RMSE
Figures 14(a) and (b)	SIFT	0.3200,0.2014	0.9947	34.4827	0.2440
	Reference [27]	**0.5936,0.5196**	0.9904	34.4811	0.2412
	Reference [28]	0.2209,0.1409	0.9881	32.5297	0.1783
	Reference [33]	0.2408,0.1473	0.9946	34.2815	0.2527
	Reference [35]	0.2359,0.1476	**0.9950**	34.7100	0.1845
	Reference [36]	0.4427,0.4191	**0.9950**	**34.8001**	**0.1527**
	The proposed	0.4219,0.3674	**0.9950**	34.6864	0.1797
Figures 14(c) and (d)	SIFT	0.2573,0.3948	0.9933	34.5568	0.2704
	Reference [27]	**0.3499,0.7254**	0.9934	33.6841	0.2666
	Reference [28]	0.1932,0.2901	0.9718	28.9722	0.2100
	Reference [33]	0.2296,0.3229	**0.9941**	**35.5245**	0.2805
	Reference [35]	0.1803,0.2465	0.9931	33.3533	0.2070
	Reference [36]	0.3497,0.3579	0.9937	34.2042	0.2227
	The proposed	0.2494,0.5239	0.9940	35.2634	**0.2038**
Figures 14(e) and (f)	SIFT	0.2291,0.2365	0.8774	23.3283	26.1872
	Reference [27]	**0.3088,0.3069**	0.8777	23.1471	26.5912
	Reference [28]	0.0832,0.0932	0.8849	23.7788	10.3519
	Reference [33]	0.1332,0.1529	0.8860	23.1713	0.5457
	Reference [35]	0.1304,0.1396	0.8844	23.5649	0.4938
	Reference [36]	0.2401,0.2423	0.8914	23.0535	29.3847
	The proposed	0.1654,0.1653	**0.9001**	**23.9899**	**0.4558**

As shown in figure 16, the stitched images using the above algorithm have no significant visual differences. To accurately and objectively evaluate the stitching quality of each algorithm, the corresponding matching rate, SSIM, PSNR and RMSE are shown in table 9. As shown in table 9, due to the fact that the FSC algorithm can more accurately remove the mismatched feature points of the NNRD algorithm, the proposed algorithm has a lower matching rate. In terms of SSIM, PSNR and RMSE, the proposed algorithm has the best performance, indicating that it can effectively improve the quality of image stitching.

G. ALGORITHM COMPREHENSIVE PERFORMANCE TESTING

To more accurately evaluate the performance of the algorithm, the above seven algorithms were independently

TABLE 10. Comparison of algorithm performance in dataset 1.

Algorithm	Number of failures	Time(s)		Matching rate		SSIM		PSNR		RMSE	
		Mean	Std.Dev	Mean	Std.Dev	Mean	Std.Dev	Mean	Std.Dev	Mean	Std.Dev
SIFT	0	10.7199	9.0015	0.3480	0.2225	0.8653	0.2708	31.5902	14.2831	5.7471	8.4727
Reference [27]	0	4.3266	1.7126	**0.5335**	0.3090	0.8699	0.2662	31.9487	16.0585	5.5502	8.4813
Reference [28]	0	6.1365	2.9688	0.3802	0.1881	0.6360	0.4798	24.9055	20.7676	5.8610	8.2941
Reference [33]	0	4.7579	2.8423	0.2670	**0.1814**	0.8723	0.2683	31.5610	**12.3174**	0.3383	0.1899
Reference [35]	0	4.4583	4.6151	0.2485	0.1819	0.8934	0.2171	32.0169	12.5290	**0.2853**	0.1877
Reference [36]	0	4.1765	1.4769	0.4332	0.2592	0.8534	0.2811	30.1671	13.3726	5.7727	9.2140
The proposed	0	**2.8695**	**0.9399**	0.3931	0.2328	**0.9002**	**0.2170**	**32.4022**	12.4727	0.2946	**0.1875**

TABLE 11. Comparison of algorithm performance in dataset 2.

Algorithm	Number of failures	Time(s)		Matching rate		SSIM		PSNR		RMSE	
		Mean	Std.Dev	Mean	Std.Dev	Mean	Std.Dev	Mean	Std.Dev	Mean	Std.Dev
SIFT	1053	9.3004	5.3256	0.0366	0.0396	0.4722	0.1622	14.1776	3.1820	3.0872	3.0270
Reference [27]	1199	6.7067	3.3732	0.2880	0.2820	**0.4780**	0.1518	**14.4948**	2.4343	2.5856	2.0399
Reference [28]	1050	4.6063	1.9363	0.0496	0.0518	0.4761	0.1485	14.4946	2.5374	2.6380	2.0124
Reference [33]	173	5.1821	2.1858	0.0157	0.0192	0.4402	0.1493	13.9648	2.3400	0.4888	**0.1057**
Reference [35]	107	6.1121	4.0950	0.0128	0.0169	0.4467	0.1510	13.9781	2.5144	0.4893	0.2075
Reference [36]	860	4.9926	**1.2242**	**0.3030**	0.0309	0.4331	0.1654	13.6044	2.7613	2.9160	2.4981
The proposed	**70**	**4.8755**	2.0776	0.0106	**0.0163**	0.4452	**0.1474**	13.9857	**2.2803**	**0.4828**	0.1545

used for image stitching on the test set. Meanwhile, the number of failures, algorithm time, matching rate, SSIM, PSNR, and RMSE were calculated, and the results are shown in tables 10-11 (images that failed to be stitched were not subjected to statistical calculations).

As shown in tables 10-11, in terms of the mean of time, the proposed algorithm has the best performance compared to other algorithms. The improvement of the proposed algorithm is relatively small due to the large overlap regions of the images in dataset 2. Compared with the SIFT algorithm, the proposed algorithm has reduced the time by 73.24% and 47.58%, respectively. Meanwhile, the proposed algorithm has a small standard deviation in time, indicating its good stability. In terms of matching rate, the algorithm of reference [27] has the best performance due to the fact that the algorithms of references [27] and [36] use the RANSAC algorithm. Although the matching rate is relatively low when the proposed algorithm uses the FSC algorithm, it has better performance compared to the other three algorithms. In terms of the number of failures, the proposed algorithm has the best performance. The proposed algorithm reduces number of failures by 93.35% compared to the SIFT algorithm. In terms of RMSE, the proposed algorithm performs second only to the algorithm of reference [35] in dataset 1, and has the best performance on dataset 2. Compared to the SIFT algorithm, the RMSE of the proposed algorithm decreased by 94.87% and 84.36%, respectively. In terms of SSIM and PSNR, the proposed algorithm has the best performance in dataset 1, while the algorithm proposed in dataset 2 performs relatively well. However, considering the number of failures, it can be concluded that the proposed algorithm has the best image stitching quality. This indicates that the proposed algorithm improves the quality of image stitching. In summary, the proposed algorithm improves the efficiency and quality of image stitching.

H. DISCUSSION

It can be seen that the proposed algorithm effectively reduces the computation of non-overlapping regions and weak texture regions in the preprocessing stage from the above experimental results. The time required is also reduced for feature point extraction. At the same time, the proposed algorithm avoids the impact caused by inaccurate calculation of phase correlation algorithms. In the descriptor stage, the proposed descriptor has better matching performance and smaller dimensions compared to other descriptors. And the proposed descriptor effectively reduces the time cost of feature point matching stage. In the feature matching stage, the proposed twice matching method based on extreme value classification effectively reduces the time overhead of the feature point matching stage compared with other methods. And this method improves the stitching quality by combining with the FSC algorithm. The effectiveness of the proposed algorithm was validated in all three stages mentioned above.

According to the analysis of time, stitching quality, and comprehensive performance, the proposed algorithm has the lowest time overhead and higher stitching quality compared to existing algorithms. Compared with the SIFT algorithm, the time has reduced by 73.24% and 47.58%, the RMSE has reduced by 94.87% and 84.36%, and the number of images with failed stitching has decreased by 93.35%. The proposed algorithm significantly reduces time costs and improves image stitching quality.

However, it is known from the principle of the algorithm that the proposed algorithm will not achieve ideal results in preprocessing when dealing with images with large overlapping regions and complex textures. It would result in less overall improvement of the algorithm. At the same time, the resolution of the image has a direct impact on the efficiency of the algorithm, so the time of the proposed algorithm will also be relatively long when the image resolution is huge. In addition, the proposed algorithm's image stitching quality is not satisfactory when dealing with complex affine transformations. Therefore, we will consider studying the speed improvement of high-resolution complex texture images and handling complex affine transformations in further work.

VI. CONCLUSION

In conclusion, A fast image stitching algorithm based on texture classification and improved SIFT is presented, which is improved in three stages. The effective regions of images are calculated using phase correlation algorithm and gradient based texture classification method. And a feature descriptor with only 40 dimensions was designed. Meanwhile, a twice matching method based on extreme value classification was proposed to reduce the time cost of feature point matching. Importantly, the proposed algorithm has the lowest time overhead and higher stitching quality compared to existing algorithms. The proposed algorithm significantly reduces time costs and improves image stitching quality. A better stitching quality and efficiency are achieved applying the new algorithm. It also proves that the proposed algorithm has potential application value in real-time image stitching. However, when the preprocessing effect is poor, the algorithm improvement is relatively small. In addition, the proposed algorithm is difficult to handle complex affine transformations. Therefore, we will consider studying the speed improvement of high-resolution complex texture images and handling complex affine transformations in further work.

REFERENCES

[1] D. Ghosh and N. Kaabouch, "A survey on image mosaicing techniques," *J. Vis. Commun. Image Represent.*, vol. 34, pp. 1–11, Jan. 2016.

[2] W. Ma, Z. Wen, Y. Wu, L. Jiao, M. Gong, Y. Zheng, and L. Liu, "Remote sensing image registration with modified SIFT and enhanced feature matching," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 1, pp. 3–7, Jan. 2017.

[3] H.-H. Chang, G.-L. Wu, and M.-H. Chiang, "Remote sensing image registration based on modified SIFT and feature slope grouping," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 9, pp. 1363–1367, Sep. 2019.

[4] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. Alvey Vis. Conf.*, Aug. 1988, pp. 147–151.

[5] D. G. Viswanathan, "Features from accelerated segment test (fast)," in *Proc. 10th Workshop Image Anal. Multimedia Interact. Services*, May 2009, pp. 6–8.

[6] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2564–2571.

[7] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2548–2555.

[8] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[9] H. Cai, X. Wu, L. Zhuo, Z. Huang, and X. Wang, "Fast SIFT image stitching algorithm combining edge detection," *Infr. Laser Eng.*, vol. 47, no. 11, 2018, Art. no. 1126003.

[10] Y. Li, G. Li, S. Gu, and K. Long, "Image mosaic algorithmic based on area blocking and SIFT," *Opt. Precision Eng.*, vol. 24, pp. 1197–1205, May 2016.

[11] J. Liu, P. You, J. Zhan, and J. Liu, "Improved SIFT for fast image stitching and ghosting optimization algorithm," *Opt. Precis. Eng.*, vol. 28, pp. 2076–2084, Sep. 2020.

[12] X. Li, Y. Chen, W. Li, Y. Li, L. Zheng, and S. Wu, "Projection aided digital shearography scanning detection technology," *Infr. Laser Eng.*, vol. 50, pp. 325–332, Aug. 2021.

[13] H. Shi, L. Guo, S. Tan, G. Li, and J. Sun, "Improved parallax image stitching algorithm based on feature block," *Symmetry*, vol. 11, no. 3, p. 348, Mar. 2019.

[14] Y.-B. Liu, J.-H. Qin, M.-Y. Zhu, and T.-T. Huang, "Fast stitching method for multi-view images of cupping spots," *Signal, Image Video Process.*, vol. 17, no. 5, pp. 1905–1913, Jul. 2023.

[15] Y. Yang, C. Zhao, and X. Zhang, "Image stitching method based on SURF and improved RANSAC," *Laser J.*, vol. 42, pp. 105–108, Apr. 2021.

[16] J. Ma, J. Jiang, H. Zhou, J. Zhao, and X. Guo, "Guided locality preserving feature matching for remote sensing image registration," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 8, pp. 4435–4447, Aug. 2018.

[17] C. Wang, T. Lei, B. Zhang, R. Xu, and D. Chen, "Quick stitching of unmanned aerial vehicle remote sensing images based on improved SIFT algorithm," *J. Central China Normal Univ., Natural Sci.*, vol. 57, pp. 302–309, Apr. 2023.

[18] B. Kupfer, N. S. Netanyahu, and I. Shimshoni, "An efficient SIFT-based mode-seeking algorithm for sub-pixel registration of remotely sensed images," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 2, pp. 379–383, Feb. 2015.

[19] H. Zhou, W. Yi, L. Du, and Y. Qiao, "Convolutional neural network-based dimensionality reduction method for image feature descriptors extracted using scale-invariant feature transform," *Laser Optoelectron. Prog.*, vol. 56, no. 14, 2019, Art. no. 141008.

[20] Y. Li, W. Qiao, H. Jin, J. Jing, and C. Fan, "Reliable and fast mapping of keypoints on large-size remote sensing images by use of multiresolution and global information," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 9, pp. 1983–1987, Sep. 2015.

[21] Z. Wang, J. Li, X. Wang, and X. Niu, "Underwater terrain image stitching based on spatial gradient feature block," *Comput., Mater. Continua*, vol. 72, no. 2, pp. 4157–4171, 2022.

[22] H. Gao, Z. Huang, H. Yang, X. Zhang, and C. Cen, "Research on improved multi-channel image stitching technology based on fast algorithms," *Electronics*, vol. 12, no. 7, p. 1700, Apr. 2023.

[23] D. Deng, "Smooth stitching method for the texture seams of remote sensing images based on gradient structure information," *Processes*, vol. 9, no. 10, p. 1689, Sep. 2021.

[24] Z. Wang, Z. Tang, J. Huang, and J. Li, "Fast calibration stitching algorithm for underwater camera," *Multimedia Tools Appl.*, vol. 82, no. 18, pp. 27707–27726, Feb. 2023.

[25] T. Zhang and Y. Cao, "Indoor scene splicing based on genetic algorithm and ORB," *Intell. Autom. Soft Comput.*, vol. 33, no. 3, pp. 1677–1685, Feb. 2022.

[26] Q. An, X. Chen, and S. Wu, "A novel fast image stitching method based on the combination of SURF and cell," *Complexity*, vol. 2021, no. 1, Jun. 2021, Art. no. 9995030.

[27] Y. Wang, Z. Tang, M. Zhong, Y. Wang, R. Zeng, D. Zhu, and C. Yang, "SIFT image stitching algorithm based on phase correlation and texture classification," *Chin. J. Quantum Electron.*, vol. 37, pp. 650–658, Nov. 2020.

[28] Y. Chen, M. Xu, H.-L. Liu, W.-N. Huang, and J. Xing, "An improved image mosaic based on Canny edge and an 18-dimensional descriptor," *Optik*, vol. 125, no. 17, pp. 4745–4750, Sep. 2014.

[29] Y. Liu, M. He, Y. Wang, Y. Sun, and X. Gao, "Fast stitching for farmland aerial panoramic images based on optimized SIFT algorithm," *Trans. Chin. Soc. Agric. Eng.*, vol. 39, pp. 117–125, Feb. 2023.

[30] J. Zhao, X. Zhang, C. Gao, X. Qiu, Y. Tian, Y. Zhu, and W. Cao, "Rapid mosaicking of unmanned aerial vehicle (UAV) images for crop growth monitoring using the SIFT algorithm," *Remote Sens.*, vol. 11, no. 10, p. 1226, May 2019.

[31] H. Liang, C. Liu, X. Li, and L. Wang, "A binary fast image registration method based on fusion information," *Electronics*, vol. 12, no. 21, p. 4475, Oct. 2023.

[32] Y. Liu, M. He, Y. Wang, Y. Sun, and X. Gao, "Farmland aerial images fast-stitching method and application based on improved SIFT algorithm," *IEEE Access*, vol. 10, pp. 95411–95424, 2022.

[33] T. Wu, I.-K. Hung, H. Xu, L. Yang, Y. Wang, L. Fang, and X. Lou, "An optimized SIFT-OCT algorithm for stitching aerial images of a loblolly pine plantation," *Forests*, vol. 13, no. 9, p. 1475, Sep. 2022.

[34] Y. Wu, W. Ma, M. Gong, L. Su, and L. Jiao, "A novel point-matching algorithm based on fast sample consensus for image registration," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 1, pp. 43–47, Jan. 2015.

[35] Y. Wang, Z. Tang, M. Zhong, Y. Wang, G. Zhao, C. Ding, and C. Yang, "Image matching algorithm for fast scale-invariant feature transformation based on mask search," *Laser Optoelectronics Prog.*, vol. 58, no. 4, 2021, Art. no. 0410010.

[36] Z. Tang, Z. Zhang, W. Chen, and W. Yang, "An SIFT-based fast image alignment algorithm for high-resolution image," *IEEE Access*, vol. 11, pp. 42012–42041, 2023.

[37] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[38] M. Zhong, Z. Tang, Y. Wang, and C. Yang, "Multi threshold SIFT image stitching algorithm based on texture classification," *Comput. Simul.*, vol. 39, pp. 364–368, Oct. 2020.

[39] Z. Tang, Z. Ding, R. Zeng, Y. Wang, J. Wen, L. Bian, and C. Yang, "Multi-threshold corner detection and region matching algorithm based on texture classification," *IEEE Access*, vol. 7, pp. 128372–128383, 2019.

[40] L. Zeng and D.-L. Gu, "A SIFT feature descriptor based on sector area partitioning," *Acta Autom. Sinica*, vol. 38, no. 9, p. 1513, 2012.

[41] S. Phos, N. Chanveasna, S. Acmatac, and V. Kimsoer. *Image for Stitching in Computer Vision (CV2) Python*. Kaggle Website. [Online]. Available: https://www.kaggle.com/datasets/phsophea101/image-for-stitching

[42] A. Wong and D. A. Clausi, "ARRSI: Automatic registration of remote-sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 5, pp. 1483–1493, May 2007.