IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 7, 2020, accepted January 3, 2021, date of publication January 18, 2021, date of current version January 27, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3052502

Superpixel-Based Local Features for Image Matching

YANG DONG, DAZHAO FAN, QIUHE MA, AND SONG JI
Institute of Surveying and Mapping, Information Engineering University, Zhengzhou 450000, China
Corresponding author: Dazhao Fan (fdzcehui@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 41401534 and Grant 41971427.

ABSTRACT Image matching is the research basis of many computer vision problems, such as intelligent driving, object recognition and structure from motion. However, the traditional feature-based image-matching results are usually very sparse and unevenly distributed for wide baseline or weakly textured images. Implementing an efficient and robust image-matching technology is a challenging task. To solve these problems, we propose an efficient extractor and binary descriptor based on superpixels and a modified binary robust independent elementary features (BRIEF) descriptor called FSRB. FSRB can improve the computational efficiency, number of matches, feature distribution and robustness of feature-based image matching. In theory, FSRB is rotation-, scale-, affine-, distorted-, and intensity-invariant. A comprehensive performance evaluation of FSRB is performed. The experimental results show that our method can effectively obtain many matches for different types of images. Compared with state-of-the-art algorithms, our method performed very well in terms of the number of correct matches (which increased by 2-5 times), time consumption, matching accuracy, matching success rate and feature repetition rate. In addition, our method is applied to sparse 3D reconstruction of multiview images, and satisfactory results are obtained.

INDEX TERMS Feature extraction, matching, superpixel, binary descriptor, 3D reconstruction.

I. INTRODUCTION

Feature-based image matching is a fundamental problem in computer vision and is widely used in image retrieval, 3D reconstruction, simultaneous localization and mapping (SLAM) and other fields. In this paper, we focus on image matching for general indoor and outdoor scenes. For current image-matching algorithms, such as Harris and Stephens [1], maximally stable extremal regions (MSER) [2], the scale-invariant feature transform (SIFT) [3], and KAZE [4], the matching results are usually very sparse and unevenly distributed, especially for wide baseline or weakly textured images. However, sufficient and evenly distributed image-matching points can obtain more accurate results for camera pose estimation, 3D reconstruction, and pattern recognition [5].

To obtain sufficient and evenly distributed matching results, we designed a superpixel-based feature extraction and description method called FSRB. First, feature extraction is performed using the cross edges of superpixels. Then, binary descriptors are calculated using the intensity comparison of the local image. Finally, the local descriptor is deformed to make it invariant to rotation and distortion by detecting the two directions of the local image. The intersection point represents the local maximum value of the image function in multiple directions, thus giving a stable positioning. The main advantage of this method over previous feature detectors based on gradients and regions is that there is no need to set any global thresholds.

A comprehensive performance evaluation of FSRB is performed. Many actual image datasets were used to design experiments, and 20,812 image pairs were constructed for 31 types of publicly available datasets. The performance of FSRB was compared with that of 26 mainstream feature extraction and matching algorithms. The experimental results show that our method can generate many uniformly distributed matching points, especially for wide baseline or weakly textured images. In addition, our method is applied to 3D reconstruction of multiview images. The application of FSRB in multiview reconstruction shows that the number of reconstruction points has increased over that of previous detectors by an order of magnitude, thus improving scene coverage and reducing errors.

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

Overall, the contributions of this paper are as follows:

1. An image interest point detection method based on the intersection of superpixel edges is designed. This method can extract many robust interest points for image matching.

2. A local feature description method combining primary and secondary directions is designed. The descriptor has a certain rotation distortion invariance.

3. Many datasets are used for experiments, and our method is compared with current mainstream algorithms. The advantages and disadvantages of our algorithm and current mainstream algorithms are discussed comprehensively.

The structure of the remainder of this paper is as follows: Section II provides a comprehensive review of current feature-based image feature detection and description methods. Section III introduces the ideas and overall process of our method in detail. Section IV uses many actual datasets for experimental verification, compares our method with current mainstream algorithms, and discusses the performance of our method in detail. Section V uses multiview image datasets to apply our method in 3D reconstruction and discusses the results. Finally, Section VI summarizes this paper and future work.

II. RELATED WORK

Traditional feature-based image-matching methods include interest point detection, local image feature description, and descriptor matching. For different stages, many scholars have conducted much research over the years. In this paper, we focus on the problems of interest point detection and local image feature description, which are reviewed below.

A. INTEREST POINT DETECTION

Interest points are defined image points that have a definite and an obvious appearance. In general, the selection of interest points should meet the following requirements [6]: distinctness, invariance, stability, seldomness, and interpretability. These requirements make interest points very useful in applications such as feature-based image matching and spatiotemporal analysis of image sequences. The earliest interest point detection method was proposed by Moravec [7]. Current research on interest point detection is divided mainly according to four feature types [8]: gradient-based features, template-based features, contour-based features, and learning-based features, as shown in Table 1.

TABLE 1. Interest point detection methods.

Methods	Strength	Limitations
Gradient-based features [3-4, 11-16]	Good robustness to scale, illumination and local affine distortions	Computation of the image gradients are sensitive to image noise and are computationally expensive
Template-based features [17-20]	Fast computation speed; fewer computation resources required	Not affine invariants, thus limiting their ability to handle changes in viewpoint
Contour-based features [21-27]	Image contours and special intersections are very deterministic and more robust to viewpoint changes	Rely on the detection quality of the image contour; the detection accuracy is low
Learning-based features [28-31]	No need to manually set parameters	Need many prior data for training; wide baseline or weak texture image matching is not considered

1) GRADIENT-BASED FEATURES

Most feature detection methods in the early literature were based on gradient calculations, such as the Harris corner detector [1] and Forstner corner detector [9]. However, gradient-based interest point detection methods have the disadvantage of being sensitive to image noise. Therefore, for interest point detection, many researchers have begun to use Gaussian pyramids, such as Difference of Gaussians (DoG), Laplacian of Gaussians (LoG), and Hessian-Laplacian [10].

In 2004, Lowe proposed SIFT [3], which uses DoG pyramids and Hessian matrices to locate interest points. However, due to the complex design of SIFT, the computational cost is very high, and a series of improved algorithms have been developed. SIFT-like methods can be divided into two categories. One category concerns the study of how to quickly and accurately calculate Hessian matrices. For example, speeded up robust features (SURF) [11] uses the box function to approximate the Hessian matrix, and dense articulated real-time tracking (DART) [12] uses the piecewise trigonometric function to approximate the Hessian matrix. The other method is to improve the performance of the Gaussian template, thereby readily causing double poles or double edges. For example, the rank order Laplacian of Gaussian (ROLG) [13] uses the hierarchical LoG filter to divide the Gaussian template into two parts, thereby readily causing the bipolar sidelobe to reach zero response. Rank-SIFT [14] uses the ranking support vector machine (RankSVM) supervised learning method to screen stable SIFT points. In recent years, some scholars have carried out research by using nonlinear partial differential equations (PDEs) for interest point detection. The wave-based detector (WADE) [15] uses wave propagation to detect interest points. KAZE [4] uses nonlinear diffusion filtering to detect interest points, but the computational cost is very high. In response, Reference [16] proposed an accelerated version of KAZE (AKAZE).

2) TEMPLATE-BASED FEATURES

The template-based methods detect interest points by comparing the intensity of the central pixel with the intensity of surrounding pixels. The smallest univalue segment assimilating nucleus (SUSAN) [17] compares the intensity of the central pixel with all pixels in its circular neighborhood to detect interest points. Features from the accelerated segment test (FAST) compare the intensity of the central pixel only with the intensity of the pixels on its neighboring ring, thereby

greatly accelerating the speed of interest point detection [18]. Reference [19] implemented rotation-invariant FAST, and Reference [20] implemented multiscale versions.

3) CONTOUR-BASED FEATURES

Contour-based features are generally defined as the local extreme points of curvature on a contour line or the intersections of multiple contour lines. Reference [21] detected interest points by detecting local extreme points of the curvature of an image contour in the scale space. Reference [22] used structured tensor and image contour information to achieve reliable interest point detection. Reference [23] adopted Gaussian kernel-based anisotropic directional derivative (ANDD) filters for contour detection to reduce the influence of noise. Recently, References [24], [25] introduced a feature detection algorithm based on image segmentation, and the algorithm uses image segmentation edge intersections as interest points for wide baseline image matching. However, the image segmentation results produced by this algorithm lack compactness and are vulnerable to image contrast and shadow. Moreover, the algorithm does not calculate the direction of features or focus much on image distortion existing in wide baseline images.

A contour-based interest point detection algorithm relies on the detection quality of image contours [26]. Thus, the scale-space processing commonly used in the above algorithms improves robustness and reduces accuracy. However, image contours and special intersections are very deterministic and more robust to viewpoint transformations [26], [27].

4) LEARNING-BASED FEATURES

In recent years, with the resurgence of machine learning, some learning-based methods have been proposed for structure from motion (SfM), visual recognition and other directions. Reference [28] developed FAST-enhanced repeatability (FAST-ER) to improve the repeatability and extraction speed of interest points. Reference [29] proposed a learned invariant feature transform (LIFT) detector and descriptor. Reference [30] used contextual information of adjacent bits to implement robust interest point detection. Reference [31] implemented a binary online learned feature detector and descriptor. However, none of these learning-based algorithms consider wide baseline or weak texture image matching.

B. LOCAL IMAGE FEATURE DESCRIPTION

Descriptors are usually formed by aggregating local image features around interest points. Local image feature descriptions can be classified mainly into two types [32]: handcrafted descriptions and learning-based descriptions, as shown in Table 2.

TABLE 2. Local image feature description methods.

Methods	Strength	Limitations
Handcrafted description [19-20, 33-50]	Good understandability; widely used	Need to manually determine the optimal parameter configuration; it is difficult to achieve high certainty, invariance, and stability
Learning-based description [51-57]	No need to manually set parameters	Need many prior data for training; the model has poor adaptability

1) HAND-CRAFTED DESCRIPTION

Handcrafted feature descriptors are currently the most widely used local image descriptors. The SIFT descriptor is currently the most popular local image feature descriptor. This descriptor has been used to derive a series of improved algorithms, such as affine-SIFT (ASIFT [33]); RGB-SIFT, HSV-SIFT [34]; and OpponentSIFT [35]. Reference [36] extended the SIFT descriptor and proposed the gradient location and orientation histogram (GLOH) descriptor, which improved robustness and discrimination. Reference [37] proposed DAISY descriptors for wide baseline stereo and dense feature extraction. Reference [38] proposed the rotation-invariant fast feature (RIFF) and used it for real-time tracking and recognition. Reference [39] proposed a compact real-time descriptor (CARD) consisting of a short binary code that can be calculated very quickly. Reference [40] proposed a method of image block description based on sparse quantization. Reference [41] proposed a local image descriptor based on a Zernike moment phase; this descriptor has strong adaptability to illumination and geometric transformation.

The local binary mode is another way to describe the spatial distribution of local images around interest points. The local binary mode encodes the relative intensity values between the central pixel and surrounding pixels. Reference [42] proposed a basic local binary pattern (LBP) method for rotation-invariant texture classification and derived a series of improved algorithms. To improve the discrimination performance of LBP descriptors, Reference [43] proposed a complete LBP algorithm. Reference [44] proposed a local ternary pattern (LTP) that extended the LBP to a three-valued encoding mode. Reference [45] proposed a local four-domain model for content-based image retrieval. Reference [46] proposed a rotation-invariant local frequency descriptor (LFD) for texture classification.

The independent binary intensity contrast descriptor uses multiple independent pixel-to-pixel binary intensity contrasts to form a binary string. References [47], [48] proposed the binary robust independent elementary features (BRIEF) descriptor. Reference [19] aimed to solve the problem that BRIEF lacks rotational invariance by proposing an oriented fast and rotated BRIEF (ORB) descriptor and used the greedy search learning method to select a better location sample. Reference [20] proposed the binary robust invariant scalable keypoints (BRISK) descriptor. Reference [49] proposed the fast retina keypoint (FREAK) descriptor based on the principle of human retinal image perception; the closer the sampling position is to the center, the higher the density is.

Reference [50] proposed a locally uniform comparison image descriptor (LUCID).

2) LEARNING-BASED DESCRIPTION

Handcrafted descriptors are a difficult means of determining the optimal parameter configuration, and designed descriptors have difficulty achieving high certainty, invariance and stability. In recent years, researchers have applied machine learning methods to descriptor design. These methods have become a popular topic in local descriptor research.

References [51], [52] proposed a unified local image descriptor construction framework that decomposed the construction of local descriptors into several modules. However, the joint optimization objective function used in this method is prone to fall into local extremes. In response, Reference [53] proposed a local descriptor learning method based on convex optimization. Reference [54] proposed a local descriptor learning method for low-dimensional boosted gradient maps (LBGM). Reference [55] proposed the linear discriminant analysis hash (LDAHash), which uses machine learning to project and quantize high-dimensional descriptors, such as SIFT, into binary string descriptors. Reference [56] proposed a binary descriptor learning method, the discriminative BRIEF (D-BRIEF) descriptor. Reference [57] introduced a boost-based binary descriptor and obtained good experimental results.

III. METHOD

The image color space is transformed, and the superpixel extraction algorithm is used on multiple image scales to obtain many uniformly distributed superpixel extraction results. The superpixel edge intersections are used as the detected interest points. On the corresponding scale, the primary and secondary directions of the interest point are calculated; the two directions are used to rotate and deform the local image; and then, the local binary descriptor is sampled. This strategy makes the features have illumination invariance, scale invariance, rotation invariance and deformation invariance and thus can better address most of the actual image matching.

A. SUPERPIXEL EXTRACTION

Superpixel algorithms group visually similar pixels to create visually meaningful entities while dramatically reducing the number of primitives used in subsequent processing steps [58]. The superpixel edge obtained by superpixel segmentation is a natural ''image content mutation'' boundary. Therefore, the intersections of multiple superpixels can be used as natural candidate interest points. Determining how to efficiently obtain accurate superpixel extraction results is a problem that this paper needs to solve first.

Superpixel generation has been an important research problem. Many classic algorithms have been proposed, including FH [59], mean shift [60], and watershed [61]. However, the superpixel results extracted by these classic algorithms lack compactness, especially when the image contrast is poor or there are shadows. Reference [62] proposed the simple linear iterative clustering (SLIC) method based on linear clustering. Because of its high efficiency and good performance, this method has been widely used. Recently, Reference [63] proposed the fast linear iterative clustering with active search (FLIC) algorithm, which reduces the overall number of iterations and improves the boundary sensitivity of the superpixel extraction results. In particular, the lowest time cost was reported for existing methods. Therefore, this paper uses the FLIC algorithm for superpixel extraction to efficiently obtain numerous candidate interest points with uniform distribution. The following briefly introduces the FLIC algorithm.

The FLIC algorithm is an improved version of the SLIC algorithm. SLIC converts a color image into a 5-dimensional feature vector in the LAB color space (which can better express the natural scene) and XY coordinates. SLIC constructs a distance metric for the 5-dimensional feature vector, and finally performs local k-means clustering on the image pixels. FLIC still uses a 5-dimensional feature vector for image pixel description and optimizes and improves the local clustering of image pixels. FLIC considers only the surrounding pixels to determine the label of the current pixel. To provide better estimates of the superpixels, each pixel proactively selects which superpixels it should belong to. The pixel allocation step and update step in FLIC are performed together. Therefore, the FLIC method needs only a few iterations to converge well, thus greatly speeding up the superpixel extraction.

A number of superpixels K and an input image $I = \{I_i\}_{i=1}^{N}$ are given, where N is the number of image pixels. First, the image is converted from the RGB space to the LAB color space to obtain the LAB value (l_i, a_i, b_i) of each pixel. Then, the image is combined with the pixel coordinate value (x_i, y_i), from which the 5-dimensional feature vector $I_i = (l_i, a_i, b_i, x_i, y_i)$ can be obtained. The distance metric between pixels I_i and I_j is defined as

$$D(I_i, I_j) = \sqrt{(l_i-l_j)^2+(a_i-a_j)^2+(b_i-b_j)^2+m[(x_i-x_j)^2+(y_i-y_j)^2]} \tag{1}$$

where m is the weight coefficient. Then, the distribution principle for pixel I_i is

$$L_i = \arg\min_{L_j} D(I_i, S_{L_j}), \quad I_j \in A_i \tag{2}$$

where A_i is the four-domain pixel of I_i and S_{L_j} is the superpixel center of pixel I_j. Then, a back-and-forth traversal similar to PatchMatch [64] is used to simultaneously perform the pixel allocation and update steps to quickly obtain the superpixel segmentation results.

B. MULTISCALE DETECTOR

Superpixel extraction can quickly obtain many significant pixel sets. The edge of the superpixel can clearly indicate the

transformation of the image content and is the extreme point of the local image area. The intersections of multiple edges can be accurately located, are easy to identify, and have better robustness under viewpoint changes. These attributes provide good interest point detection results for image matching, especially for wide baseline or weakly textured images. In this way, the superpixel extraction algorithm can be used to obtain the segmented edge intersections, thereby extracting many uniformly distributed candidate interest points.

1) DETECTION

The superpixel extraction algorithm is used to divide the image into several superpixel regions. Each superpixel region is assigned a different label value. The intersection of multiple superpixel segmentation lines is defined as a candidate interest point. The image pixel and its label category within the neighborhood $n \times n$ are checked. If there are multiple (3 or more) superpixel category labels in the local area of the image, the point is defined as a candidate interest point. Through this screening strategy, many candidate interest points with a relatively uniform distribution can be obtained. The superpixel region is essentially a collection of similar content in the image. The intersection point of the edge of the superpixel is the local extreme point of the image. The interest points thus obtained also have good repeatability, discrimination, and invariance of viewpoints, as shown in Figure 1.

FIGURE 1. Feature detection.

The two images on the left in Figure 1 are the original wide baseline images to be matched, and the red points represent corresponding points. The two images on the right in Figure 1 are the corresponding superpixel segmentation results. The superpixel region is visualized with random colors. The two partially enlarged images show the local segmentation results of the corresponding points. The corresponding points are the intersections of the edges of multiple superpixel regions and can be detected, thus illustrating the robustness of the method presented in this paper and the repeatability of detecting interest points. The content of the scene can also be indistinctly seen through the two superpixel extraction results on the right. The overall object contour can be correctly expressed, thus showing that the superpixel-based algorithm can detect interest points with obvious physical significance; this ability is very important for wide baseline or extreme deformation image matching.

2) FILTERING

Many robust interest points can be obtained from the intersection of superpixel edges. To further screen out the robust extreme points to improve the effectiveness of subsequent matching, the interest points need to be selected and judged. This approach is inspired by FAST feature detection algorithms, which use a fast and effective extreme point filtering strategy. For the candidate interest point, the superpixel segmentation score of the point is compared with the segmentation scores of the surrounding four neighborhood points. If the candidate interest point values are less than or greater than the four neighborhood values, the point is considered a robust extreme point, and the point is retained; otherwise, the point is discarded. The superpixel segmentation scores of pixels are defined as the color distance between the pixel point and the superpixel center of the candidate interest point; that is,

$$S(I_i) = \sqrt{(l_i - l_0)^2 + (a_i - a_0)^2 + (b_i - b_0)^2} \tag{3}$$

where $S(I_i)$ represents the superpixel segmentation score of pixel I_i and (l_0, a_0, b_0) represents the color center (color average) of the superpixel where the candidate interest point is located. Each color component of the LAB color gamut has a clear physical meaning. Neighboring extreme point filtering in the LAB color gamut can effectively eliminate nonrobust points, thereby providing a more compact set of interest points for subsequent feature description and matching and improving the processing efficiency of the overall process.

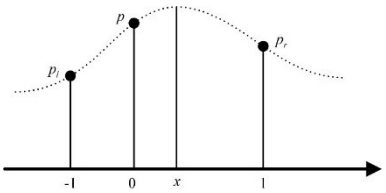


FIGURE 2. Subpixel estimation of local extrema.

3) SUBPIXEL ESTIMATION

After obtaining the interest points, a subpixel-level positioning calculation is performed to further obtain accurate local extreme points. Assume that the value at the point of interest is p, the value on the left is p_l, the value on the right is p_r, and the true extreme point to be found is x, where $-0.5 < x < 0.5$, as shown in Figure 2. Then, the local extreme point is defined as the centroid of the points as follows [65]:

$$x = \frac{\sum x_i p_i}{\sum p_i} = \frac{p_r - p_l}{p + p_r + p_l} \tag{4}$$

By using this simple but effective method to accurately locate the x- and y-directions of interest points, subpixel-level positioning results can be obtained to further improve detection accuracy and provide a basis for subsequent high-precision applications.

4) MULTISCALE DETECTION

To make the detected interest points have scale invariance, a multiscale image pyramid is constructed, and interest points are detected on different layers of images, as shown in Figure 3. When constructing the pyramid in this paper, the scale of the pyramid of layer i is defined as $sc_i = 1/(sf)^i$,

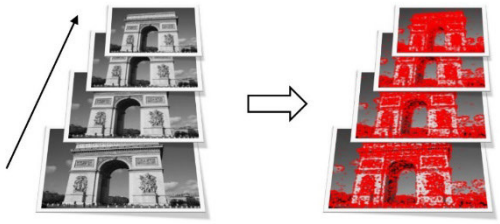


FIGURE 3. Pyramid structure.

where $i = 0$ is the original image and sf is the scale factor. If we take $sf = 1.2$ and build a pyramid with 8 layers, the theoretical detection image scale is approximately 3.6 times, thus being able to effectively cover common image scale changes and provide a basis for robust matching of interest points.

C. BINARY DESCRIPTOR

In this section, we first introduce the BRIEF feature descriptor and then introduce two methods of calculating the local image orientation to deform BRIEF to achieve rotation and local deformation invariance of the descriptor.

1) BRIEF

BRIEF is a binary-coded descriptor. It uses a local image gray value independent sampling binary test to establish descriptors. This approach has a great speed advantage over traditional local image gray histogram statistical methods to establish descriptors. The calculation steps are as follows: First, we perform Gaussian filtering to reduce noise interference. Then, taking the interest point as the center, we take the $S \times S$ neighborhood window to obtain the local image block p. We randomly select a pair of points (x, y) in p and compare the intensities of the pixel values of the points to obtain the binary value τ as follows:

$$\tau(p; x, y) = \begin{cases} 1 & p(x) < p(y) \\ 0 & p(x) \geq p(y) \end{cases} \tag{5}$$

where $p(x)$ and $p(y)$ are the intensity values of the random point pair $x = (u_1, v_1)$ and $y = (u_2, v_2)$, respectively. We repeatedly select n pairs of random points to obtain a binary vector of length n as follows:

$$f_n(p) = \sum_{1 \leq i \leq n} 2^{i-1} \tau(p; x_i, y_i) \tag{6}$$

Reference [48] also tested multiple random point pair selection methods. Among them, the Gaussian distribution mode centered on the interest point exhibits the best performance. However, the resulting binary descriptor does not have rotational invariance or local deformation invariance. Below, we introduce two local image orientation calculation methods to adapt the descriptor to a wider range of image-matching situations.

2) FIRST DIRECTION DETERMINATION

For the first direction, the gray centroid method recommended by the ORB descriptor was used for the calculation [24]. The gray centroid method uses the local image gray value to calculate the centroid and considers the connection direction between the centroid and the center of the interest point to be robust and can be used as a direction vector, defined as follows:

$$m_{pq} = \sum_{x,y} x^p y^q I(x, y) \tag{7}$$

The centroid of the local image is defined as

$$C_1 = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \tag{8}$$

The center of interest point O and its neighborhood image block centroid C_1 are used to construct a direction vector $\vec{OC_1}$. $\vec{OC_1}$ is defined as the first direction and is characterized as

$$\theta_1 = a \tan 2(m_{01}, m_{10}) \tag{9}$$

where $a \tan 2$ is the quadrant version of arctan. This method is shown in Figure 4.

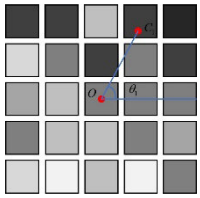


FIGURE 4. First direction.

3) SECOND DIRECTION DETERMINATION

Based on the superpixel extraction results, we give the method of defining the second direction of the interest point. Assume that there are k superpixel regions in the neighborhood of the interest point and calculate the gray centroid of each superpixel region separately as follows:

$$\begin{cases} SC = \{SC_1, SC_2, \cdots SC_n\} \\ SC_i = (x_i, y_i) \end{cases} \tag{10}$$

Take the average to obtain the superpixel average centroid as follows:

$$C_2 = \left(\frac{\sum_{i=1}^{n} x_i}{n}, \frac{\sum_{i=1}^{n} y_i}{n} \right) \tag{11}$$

The center of interest point O and its neighborhood image superpixel average centroid C_2 are used to construct a direction vector $\vec{OC_2}$. $\vec{OC_2}$ is defined as the second direction and is characterized as

$$\theta_2 = a \tan 2 \left(\sum_{i=1}^{n} x_i, \sum_{i=1}^{n} y_i \right) \tag{12}$$

where $a \tan 2$ is the quadrant version of arctan.

4) ROTATION AND DEFORMATION

After obtaining the two directions, let $\theta' = \theta_2 - \theta_1$ and $\theta = \theta' - \theta'\%\Delta\theta$, where $\Delta\theta$ is the angular interval value. First, the first direction is rotated from θ_1 to the 0° direction. Then, the second direction is rotated to the direction of the adjacent integer θ angle. After the rotation of the first direction, the descriptor can become rotation invariant. After the rotation of the second direction, the local image is actually deformed so that the resulting local descriptor is deformation invariant, as shown in Figure 5.

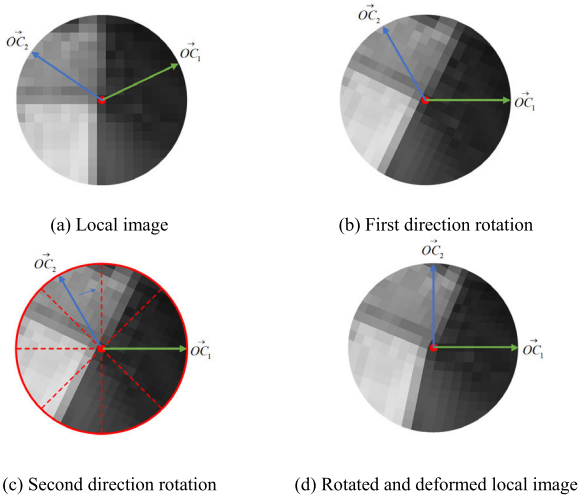


FIGURE 5. Rotation and deformation.

In actual processing, the calculation of the overall rotation and deformation in the primary and secondary directions of the local image takes a relatively long time. To further reduce the calculation time while keeping the local image unchanged, only the binary contrast sampling coordinates are transformed. Then, the transformed coordinates are used to obtain the gray value of the original image for comparison. This strategy is equivalent to transforming the image before sampling and can greatly reduce the running time of descriptor generation and improve the overall operation efficiency.

D. SUPERPIXEL-BASED LOCAL FEATURES

The overall process of the FSRB algorithm proposed in this paper is shown in Figure 6. The extraction and description strategy can be expressed as follows: (1) Generate an image pyramid from the input image, perform superpixel extraction on the multilayer pyramid image, and define the intersection of the superpixel edges as a candidate interest point. (2) Perform extreme value filtering and subpixel precise positioning of candidate interest points. (3) Calculate the primary and secondary directions of candidate interest points, and transform the sampling coordinates. (4) Finally, the pixel intensity of the local image of the candidate interest point is used to compare and generate binary descriptors. Our method has a high degree of parallelization and can easily achieve parallel optimization on a CPU or GPU to further improve processing efficiency.

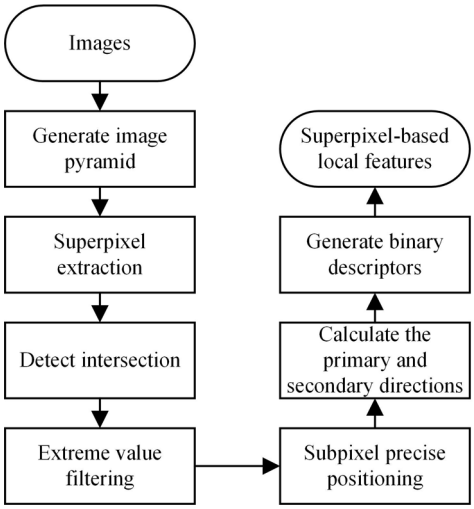


FIGURE 6. Overall processing flowchart.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

A. DATASETS

Experiments were performed using 8 test sequences from the Affine Covariant Features dataset [66], 20 test sequences from the RGB-D dataset [67], [68], and 3 test sequences from the ISPRS dataset [69]. In the 8 test sequences of the Affine Covariant Features dataset, there are 6 types of changes in imaging conditions: viewpoint changes, scale changes, image blur, rotation changes, JPEG compression, and illumination changes. Each test sequence contains 6 images with progressive geometric or photometric transformations; all of these images are of medium resolution. The RGB-D dataset is an actual video dataset. Each test sequence has from several hundred to thousands of images, including those showing static objects, dynamic objects, rich textures, weak textures, structured objects, unstructured objects, etc., which can fully reflect the performance of feature extraction and matching algorithms in practical applications. The ISPRS dataset includes wide baseline image sets, including general indoor and outdoor scenes. The image data have obvious changes in viewpoint and relatively high image resolution, which can effectively reflect the matching performance of the algorithm for wide baseline images. A detailed description of the dataset is shown in References [66]–[69].

B. SETUP

We combine current mainstream feature extraction and description methods and design 27 methods for experimental comparative analysis; these methods include sift [3], surf [11], orb [19], akaze [16], kaze [4], brisk [20], dlco [53], latch [70], freak [49], daisy [37], binboost [54], lucid [50], brief [47], [48], msd [71], star [72], fast [28], agast [73], asift [33], mods [74], frif [75], harraff [76], hessaff [76], mseraff [2], liop [77], [78], oiop [77], [78], miop [77], [78], and fsrb (ours). The descriptor matching uses brute-force matching based on the Hamming distance for binary

descriptors and the Euclidean distance for floating-point descriptors. After the initial matching, first, grid-based motion statistics (GMS) [79] was used to remove the mismatches, and second, the fundamental matrix constraint random sample consensus (RANSAC) [80] was used to remove the mismatches. A comparison of the algorithm experimental designs is shown in Table 3.

TABLE 3. Feature extraction and description methods.

Order	Method name	Extractor	Descriptor
1	sift [3]	sift	sift
2	surf [11]	surf	surf
3	orb [19]	orb	orb
4	akaze [16]	akaze	akaze
5	kaze [4]	kaze	kaze
6	brisk [20]	brisk	brisk
7	dlco [53]	sift	dlco
8	latch [70]	sift	latch
9	freak [49]	sift	freak
10	daisy [37]	sift	daisy
11	binboost [54]	sift	binboost
12	lucid [50]	sift	lucid
13	brief [47-48]	sift	brief
14	msd [71]	msd	orb
15	star [72]	star	sift
16	fast [28]	fast	sift
17	agast [73]	agast	sift
18	asift [33]	asift	asift
19	mods [74]	mods	mods
20	frif [75]	frif	frif
21	harraff [76]	HarrisAffine	sift
22	hessaff [76]	HessianAffine	sift
23	mseraff [2]	MSERAffine	sift
24	liop [77-78]	HessianAffine	liop
25	oiop [77-78]	HessianAffine	oiop
26	miop [77-78]	HessianAffine	miop
27	fsrb (ours)	fsrb (ours)	fsrb (ours)

In terms of matching performance evaluation, multiple indicators are used for evaluation calculations; these indicators include the number of matches, run time, matching accuracy, matching success rate, feature repetition rate, and scene coverage. The number of matches is defined as the number of matches ultimately obtained after GMS and RANSAC purification. The run time is defined as the total time needed to obtain the final matching point through feature extraction, description and matching after loading the image into memory. The matching success rate is defined as the ratio of the number of correct matches to the total number of matches. The feature repetition rate is defined as the ratio of the number of repeated interest points to the total number of interest points extracted. Here, a repeated interest point is an interest point extracted from image A and mapped onto image B through the homography matrix. Within the error threshold, if the interest point is also detected in image B, the interest point is considered to be a repeated point. Scene coverage refers to the degree of coverage and uniform distribution of the final matches in the overall image and is determined mainly by visual perception.

We use a variety of methods to identify correct matching point pairs. For the Affine Covariant Features dataset, the homography matrix between two images is known and is used for the threshold decision. For the RGB-D dataset, the camera matrix, rotation vector and translation vector are known, and the back-projection error is used for the threshold decision. For the ISPRS dataset, the accurate external parameters of the camera are unknown. Here, the performance is evaluated by the total number of matches.

In the experiments, the repeated point judgment threshold, the homography matrix error judgment threshold, and the triangulation back-projection error judgment threshold are all set to $\varepsilon = 3$. RANSAC uses a fundamental matrix model with a threshold of 3. GMS uses the default parameters. Feature extraction and matching algorithms are implemented with their default thresholds by using the OpenCV function or source code published by the author. The numerical parameters in the ORB descriptor limit the maximum number of features detected. For a fair comparison, we set this value to 100,000 to remove the limitation. The feature descriptor matching uses OpenCV to achieve CUDA-accelerated brute-force matching. The number of image pyramid layers in our method is set to 8, the scale factor is set to $sf = 1.2$, and the angular interval is set to $\Delta\theta = 15°$. The experimental platform uses a dual Intel Xeon Gold 6140 CPU, an Nvidia T4 GPU, and 160 GB of memory.

C. RESULTS

1) AFFINE COVARIANT FEATURES BENCHMARK DATASET EXPERIMENTS

First, the Affine Covariant Features benchmark dataset is used for the experimental evaluation. Image 1 in each test sequence is sequentially formed with the remaining images 2-6 to form image pairs for matching experiments. As the number increases, the greater the difference is between the corresponding image and image 1, and the more difficult it is to match. This process can ably evaluate the characteristics of viewpoint invariance, scale invariance, rotation invariance and illumination invariance of the algorithm.

The specific matching results of the proposed algorithm are shown in Figure 7. The experimental results indicate that our algorithm can better extract and match many matching points, the matching point distribution is relatively uniform, and the algorithm can achieve better scene coverage. The final result is far better than the extraction matching results of the other 26 matching methods.

a: NUMBER OF MATCHES

The number of matches directly reflects the performance of the matching algorithm. The greater the number of matches that are ultimately obtained, the better the matching algorithm. The horizontal axis represents the image pairs for each test sequence, and the vertical axis represents the number of matches that are ultimately obtained. From this, the trend of the number of matches can be seen, as shown in Figure 8. The horizontal axis numbers are 2 to 6, which indicate the matching results between image 1 and images 2-6. From the

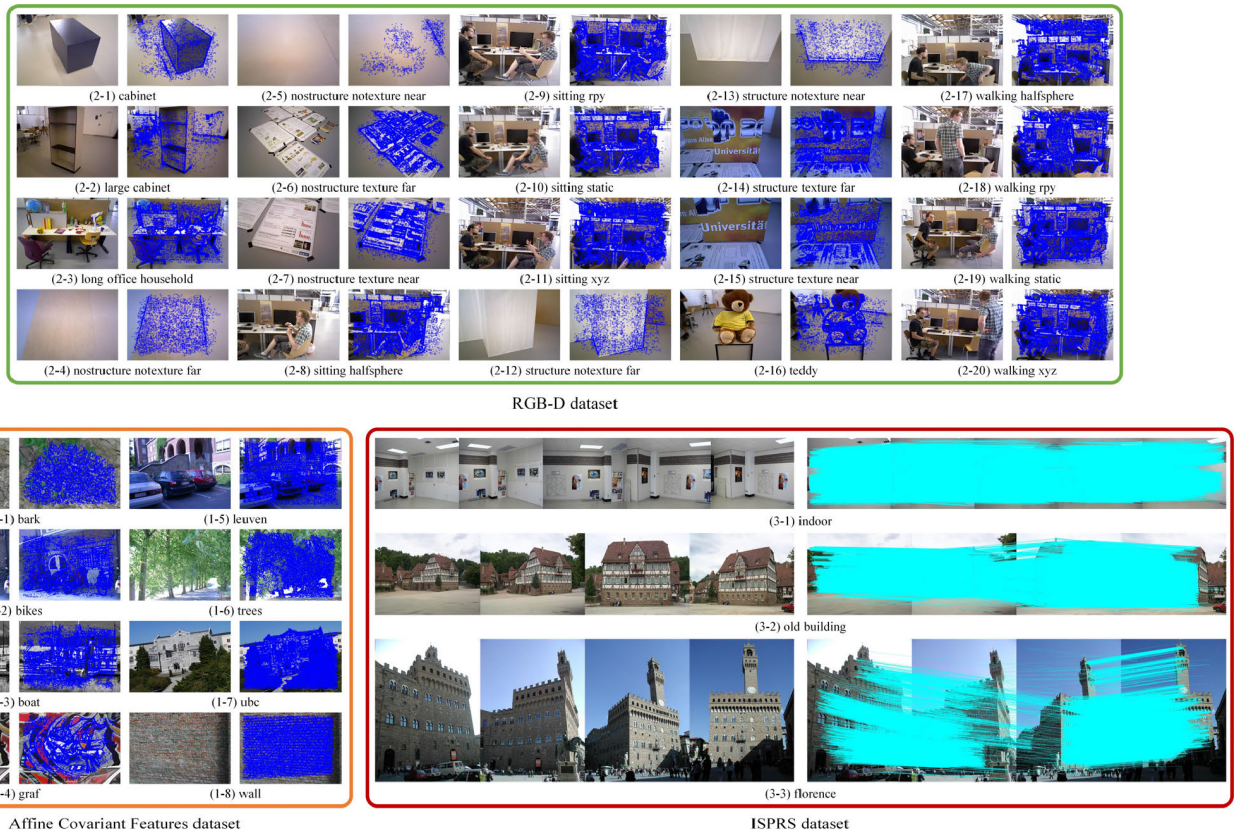


FIGURE 7. **Matching results.**

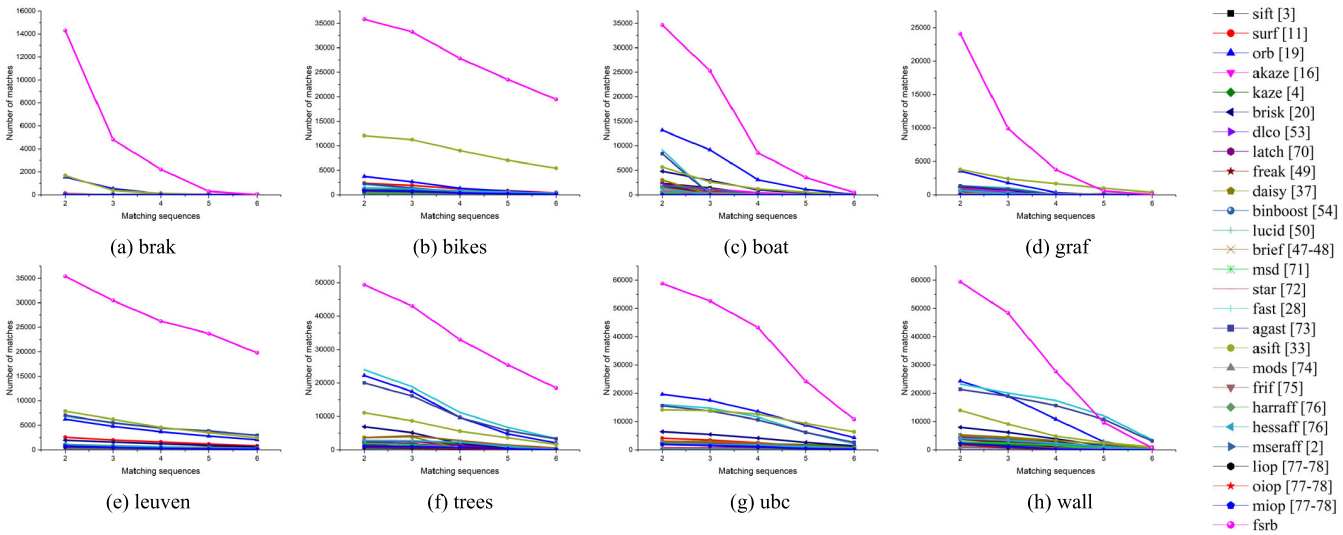


FIGURE 8. **Number of matches.**

description of the dataset, as the number of image pairs increases, it becomes more difficult to match image pairs and extract rich feature matching points.

The experimental results in Figure 8 illustrate that the overall matching curve shows a downward trend, which accords with expectations. Overall, fsrb, orb, and asift show good matching performance, and our method shows excellent performance in all test sequences. Our method can extract many matches in a variety of practical application scenarios, such as viewpoint changes, scale changes, image blurs, rotation changes, JPEG compression, and illumination changes, and is far superior to the other feature descriptors. Compared with the state-of-the-art methods (orb and asift), our method can provide 3-7 times as many matching points.

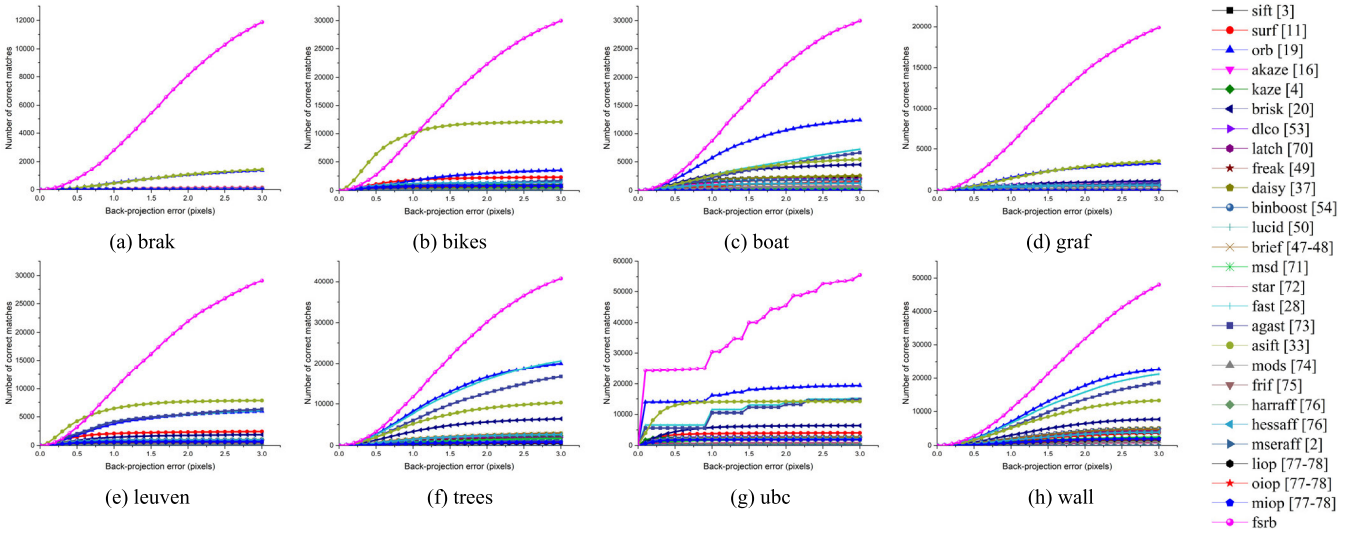


FIGURE 9. Error threshold and number of matches.

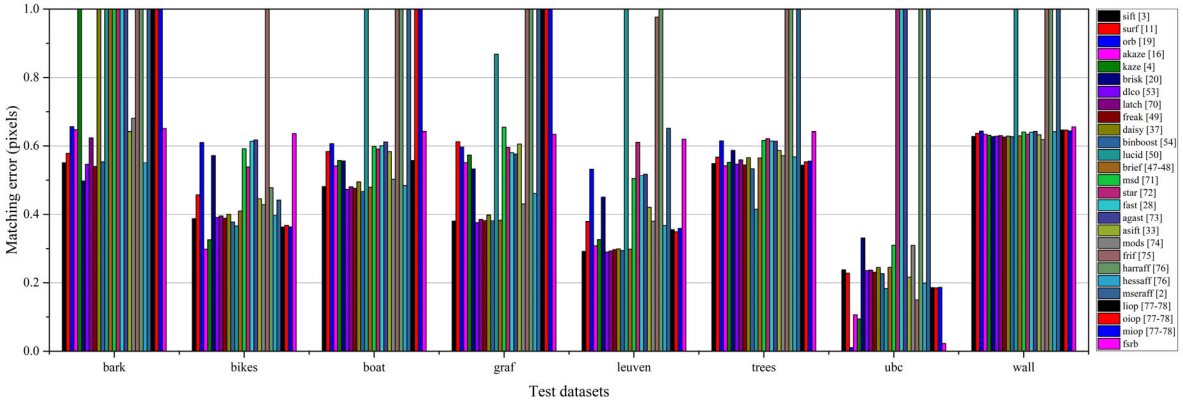


FIGURE 10. Statistics of matching accuracy.

b: MATCHING ACCURACY

Matching accuracy is an important indicator in the performance evaluation of matching algorithms. Matching accuracy reflects the independence, discrimination, and discriminative power of the extracted feature points. The error threshold of the projected image points is shown on the horizontal axis, and the number of correct matches under the corresponding threshold is shown on the vertical axis, as shown in Figure 9. In the experiments, the projection error is incremented by 0.1 pixels, from 0-pixel error to 3-pixel error. The number of matching points between images 1 and 2 under the corresponding error threshold in each test sequence is counted. Figure 9 shows that under different error threshold constraints, the number of matches that our method can extract is far better than the number of matches extracted by the current mainstream algorithms. In Figure 9(g), the matching error curve rises stepwise; this rise is determined by the characteristics of the test image itself. The test sequence image corresponding to Figure 9(g) is a JPEG compressed image. Images 1 to 6 show an increasing mosaic effect, which results in a steplike error-matching curve.

The statistical average matching error under the constraint of a 1-pixel error threshold is shown in Figure 10. As shown in Figure 10, the average matching error of our algorithm is approximately 0.6 pixels, which is comparable to the overall accuracy of current matching algorithms and is within the tolerable range of error. Of course, for applications that require extremely high matching accuracy, our matching result can be used as an initial value, and least-square matching [81] or phase correlation matching [82] can be performed to achieve high-precision matching results.

c: MATCHING SUCCESS RATE

The matching success rate is a quantitative evaluation of the performance of the matching algorithm under a specific matching accuracy. The matching results of images 1 and 2 in each test sequence are taken for statistics, with the number of matches being shown on the horizontal axis and the matching success rate being shown on the vertical axis, as shown in Figure 11. The closer to the upper-right corner in Figure 11, the better the matching algorithm performance is. Figure 11 shows that the success rate of our algorithm

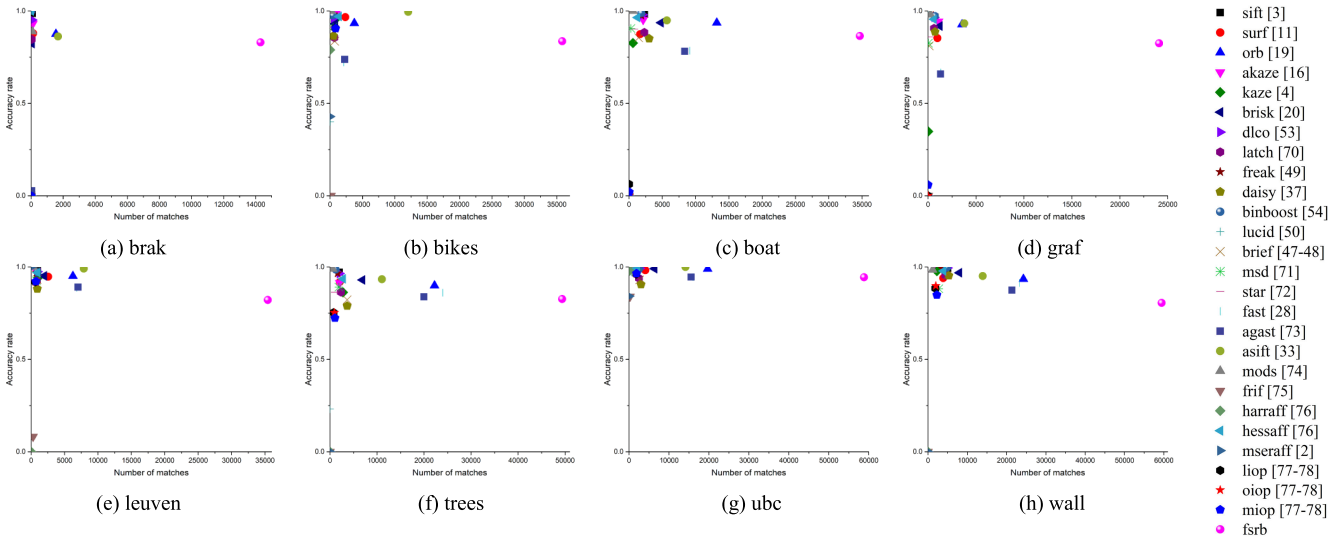


FIGURE 11. Number of matches and success rate.

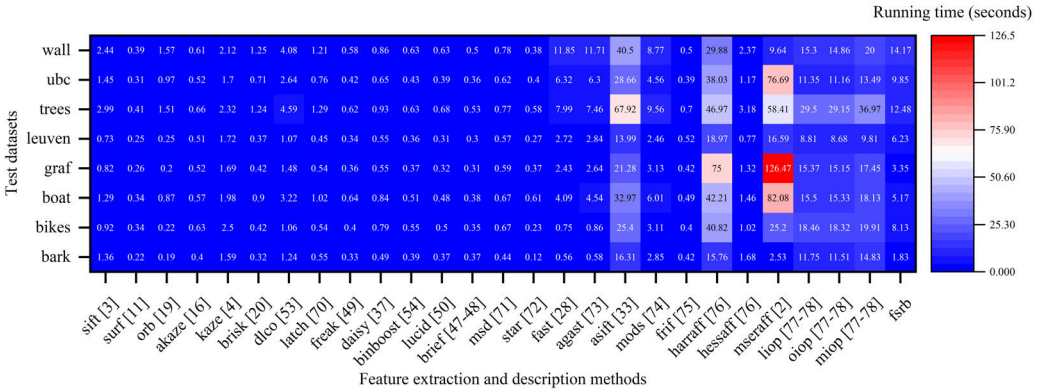


FIGURE 12. Running time.

is comparable to that of current mainstream matching algorithms. However, the number of matches is far greater than that of current mainstream matching algorithms.

d: RUNNING TIME

The running time directly represents the computational complexity of the matching algorithm. The matching results of images 1 and 2 in each test sequence are taken for statistics. The time consumption of each matching algorithm under the corresponding test image sequence is calculated, as shown in Figure 12. The overall time consumption of the algorithm presented in this paper is on the order of seconds and is at a medium level; this result is better than the time consumption of wide baseline matching algorithms, such as asift, mods, harraff, mseraff, liop, oiop, and miop.

The statistics of the number of matches indicate that the total number of matches obtained by our algorithm is much higher than that of the other matching algorithms, thus resulting in a large time consumption. The number of matches is divided by the matching time to obtain the calculation speed of the matching algorithm. The overall efficiencies of the matching algorithms can be compared fairly. The horizontal axis represents the test sequence category and matching algorithm, and the vertical axis represents the running efficiency. The statistical results are shown in Figure 13. The orb algorithm has the highest running efficiency, followed by the efficiencies of our algorithm, surf and the akaze algorithm. The running efficiency of our algorithm is better than the running efficiency of most of the mainstream algorithms and can meet the needs of general scenarios.

e: FEATURE REPETITION RATE

The feature repetition rate characterizes the repeatability of features and can directly explain the quality of feature extraction algorithms. The matching results of images 1 and 2 in each test sequence are taken for statistics. The horizontal axis represents the test sequence category and feature extraction algorithm, and the vertical axis represents the feature repetition rate. The statistical results are shown in Figure 14. The feature repetition rate obtained by our algorithm is optimal in all test sequences; this rate is followed by the feature repetition rates of the orb, brisk, fast, and agast

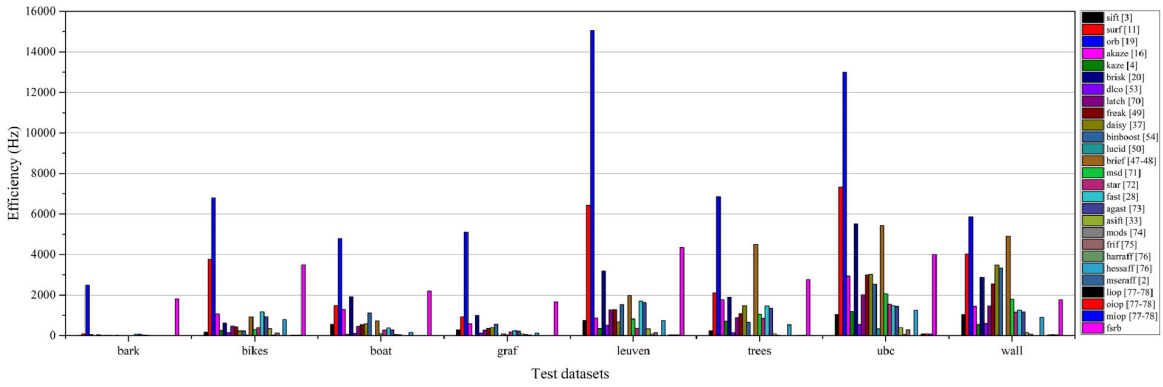


FIGURE 13. Matching efficiency.

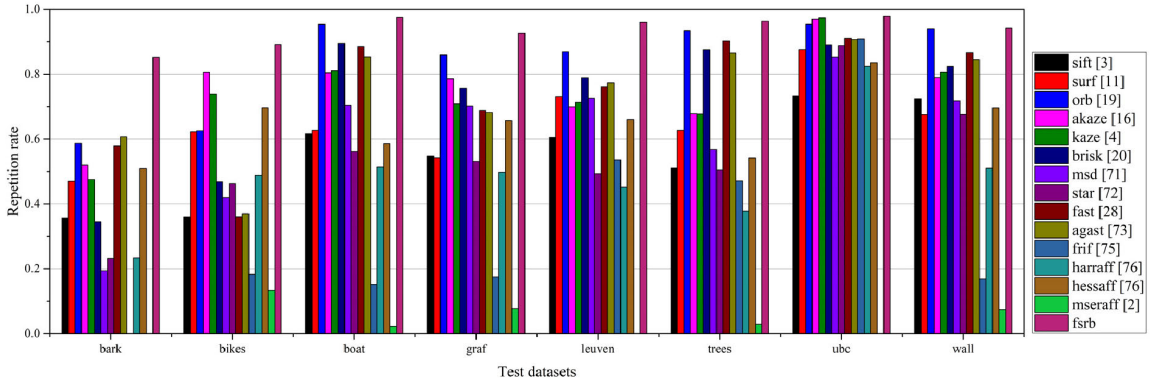


FIGURE 14. Feature repetition rate.

extraction algorithms. This result also shows that features based on the intersection of superpixel edges can implement feature repetition well.

The results obtained by the overall experiments on the Affine Covariant Features dataset show that our algorithm can extract many interest points that have excellent scene coverage and an excellent repetition rate. The matching results are far superior to those of the current algorithms, and the matching accuracy and matching efficiency of our algorithm are comparable to those of current mainstream algorithms. Our algorithm can also perform well with two wide baseline test sequences of ''graf'' and ''wall''; has excellent performance in viewpoint changes, scale changes, rotation changes, and illumination changes; and can better address image matching in various practical situations.

2) RGB-D BENCHMARK DATASET EXPERIMENTS

The 15 images of each test sequence from the RGB-D benchmark dataset are compiled into a group, and the first image in the group is matched with each of the remaining 14 images in turn. As the image number increases, the greater the differences are between the corresponding image content and the first image content, and the harder it is to achieve matching. The RGB-D benchmark test data contain 20 test sequences, and 20750 pairs of images were constructed for experiments. Many experiments on actual image data with different contents can better verify the robustness of the matching algorithm.

The specific matching results of the proposed algorithm are shown in Figure 7. The experimental results show that our algorithm can obtain many matching points. The points are relatively evenly distributed, and compared to the other 26 matching algorithms in actual experiments, our algorithm can achieve better scene coverage in various scenarios. The experimental results show that for weakly textured or notexture images, ''cabinet'', ''large cabinet'', ''nostructure notexture far'', ''nostructure notexture near withloop'', ''structure notexture far'', and ''structure notexture near'', the algorithm presented in this paper can effectively extract features through superpixel segmentation in the domain of the LAB color space and obtain good matching results for weakly textured or notexture images.

The number of final correct matches for each test sequence is counted for comparative analysis. The 14 pairs of images constructed in each group are shown on the horizontal axis, and the number of correct matches is shown on the vertical axis. The statistical average is shown in Figure 15. In Figure 15, the subfigures from (a) to (t) represent the corresponding 20 test sequences from 1 to 20. As the number of image pairs increases, the baseline between image pairs gradually increases, the differences in image content gradually increase, and the number of matching points decreases.

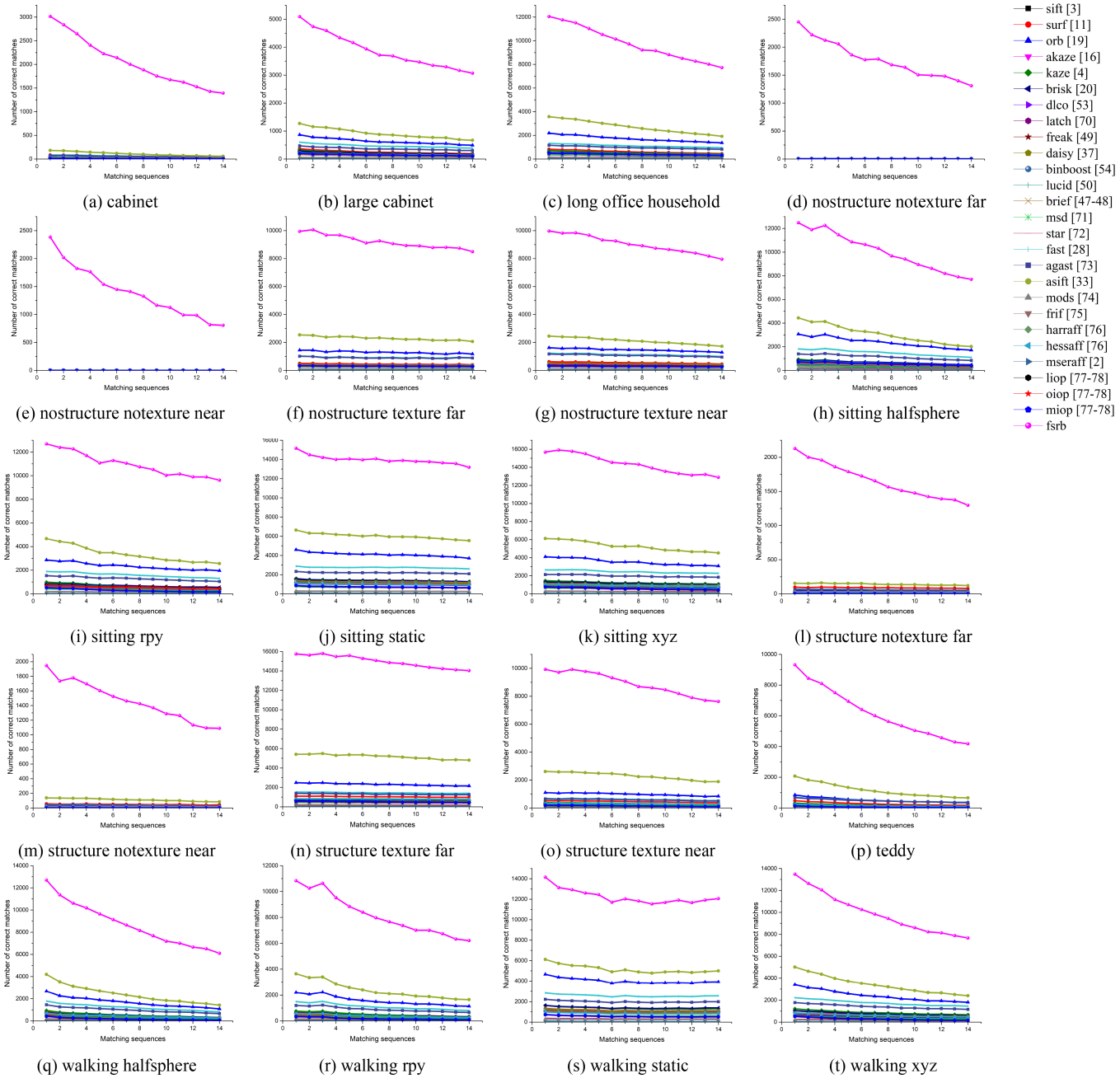


FIGURE 15. **Comparison of the number of correct matches.**

Figure 15 shows that the number of correct matching points extracted by our algorithm is far superior to that of the other algorithms; the second- and third-highest numbers of correct matches were obtained by the asift and orb algorithms, respectively. For the RGB-D dataset, the number of correct matches obtained by our algorithm is generally approximately 2-5 times higher than the number of correct matches obtained by the state-of-the-art algorithms. Especially for weakly textured regions, a sufficient number of matching points can be extracted.

The average number of correct matches obtained by each matching algorithm under each test sequence is averaged for statistics, as shown in Figure 16. The corresponding matching success rates are obtained, as shown in Figure 17. In Figure 16, the horizontal axis represents the corresponding 20 test sequences from 1 to 20, and the vertical axis represents the corresponding matching method. In the figure, the colors from blue to red indicate that the number of correct matches varies from few to most, respectively, and black indicates that the number of correct matches is zero. Figure 16 also

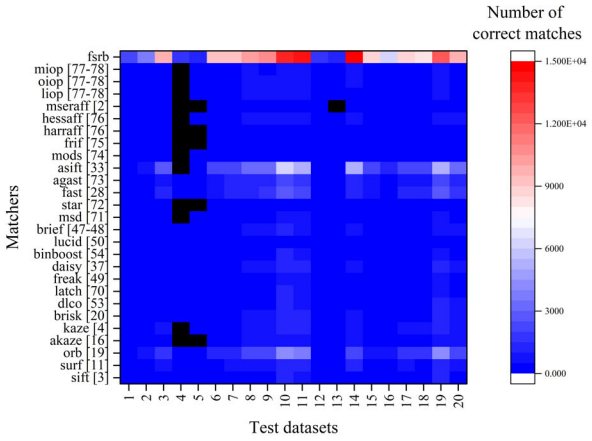


FIGURE 16. Number of correct matches.

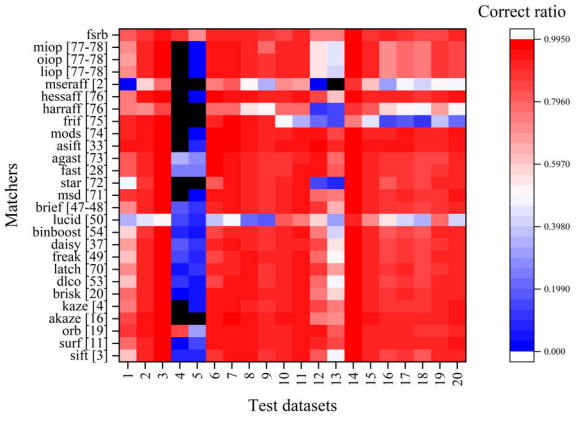


FIGURE 17. Matching success rate.

shows that compared to other algorithms, our algorithm can extract many matching points. Similarly, the horizontal axis in Figure 17 represents the corresponding 20 test sequences, and the vertical axis represents the corresponding 27 matching algorithms. In the figure, the colors from blue to red indicate matching success rates from low to high, respectively, and black indicates that the matching success rate is zero. The success rate of our algorithm is comparable to the success rate of mainstream algorithms and is far superior in weakly textured regions (test sequences 4, 5, 12, and 13). Among these algorithms, mseraff, harraff, frif, and lucid perform significantly worse than the other matching algorithms.

As in the case of the Affine Covariant Features dataset, the matching efficiencies of different feature extraction and matching methods under each test sequence in the RGB-D dataset are calculated. Using the horizontal axis as the matching test sequence and the vertical axis as the matching efficiency, the matching efficiency curve is drawn, as shown in Figure 18. Followed by our algorithm, the orb algorithm has the best matching efficiency. In the weak texture test sequences 4, 5, 12, and 13, the efficiency of our algorithm is much better than that of the mainstream algorithms.

The overall experiments on the RGB-D dataset show that in the matching of many actual constructed image pairs

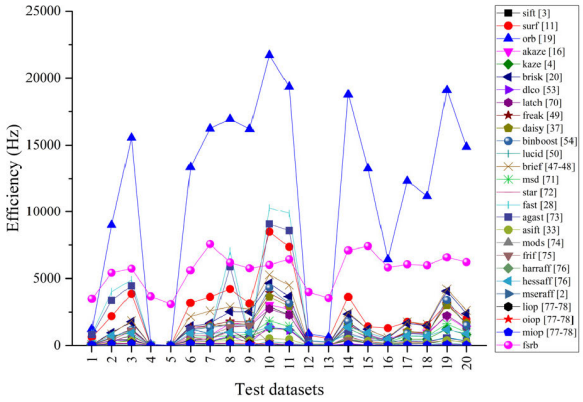


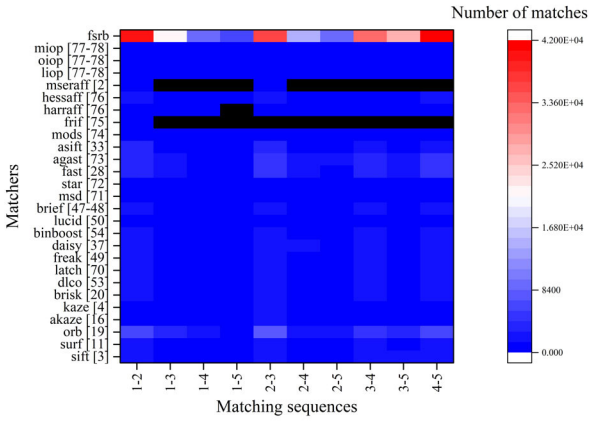
FIGURE 18. Matching efficiency.

showing, for example, static objects, dynamic objects, rich textures, weak textures, structured objects, and unstructured objects, our algorithm can correctly match many matching points. In the experiments, for the weakly textured regions that the current mainstream algorithms cannot handle, our algorithm can also match many correct points because the superpixel algorithm presented in this paper operates in the LAB color space and can capture very small detail changes. Thus, our algorithm can detect many interest points in weakly textured areas.

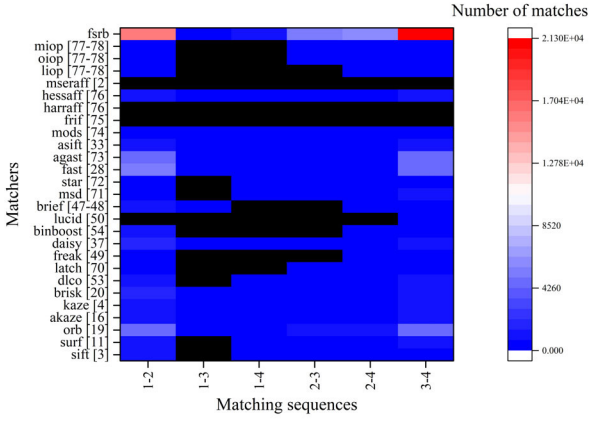
3) ISPRS WIDE BASELINE BENCHMARK DATASET EXPERIMENTS

Experiments were performed using a wide baseline dataset published by ISPRS, and the matching experimental results are shown in Figure 7. Figure 7(3-1), 7(3-2), and 7(3-3) present the matching results of our algorithm; these results are obtained by connecting the matching points of two adjacent images. Figure 7 shows that for all test sequences, the algorithm presented in this paper can obtain many correct matching points, and the matching results have good scene coverage. The second and third images in Figure 7(3-2) and the second and third images in Figure 7(3-3) are relatively extremely wide baseline image pairs. The corresponding matching results show that our algorithm can still obtain many matching points for these extremely wide baseline situations.

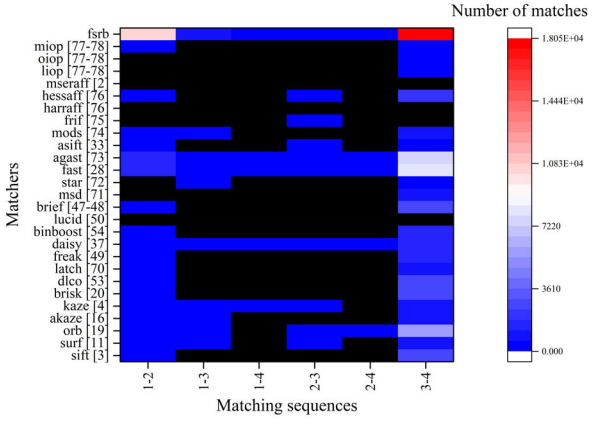
The numbers of final matching points of different matching algorithms under each test sequence are calculated. The statistical results are shown in Figure 19. In Figure 19, (a) presents the statistical results of the ''indoor'' test sequence, (b) presents the statistical results of the ''old building'' test sequence, and (c) presents the statistical results of the ''florence'' test sequence. In each subfigure, the horizontal axis represents a matched image pair, where ''i-j'' means that the ''i''-th image matches the ''j''-th image, and the vertical axis represents the different matching methods. The colors from blue to red indicate the number of final matches from least to most, respectively, and black indicates that the number of final matches is less than 20.



(a) indoor



(b) old building



(c) florence

FIGURE 19. **Number of matches.**

Figure 19 shows that our algorithm can maintain good performance for different test sequences and many matching points are obtained.

For the ''old building'' test sequence, the ''1-3'', ''1-4'', and ''2-3'' image pairs are all clearly wide baseline image pairs. Compared with the other algorithms, our algorithm can obtain many matches. For the ''florence'' test sequence, there are very clear illumination and viewpoint changes between the image pairs. Compared with the other mainstream algorithms, our algorithm can still obtain relatively many matching points. The ''1-4'' image pairs constructed in the ''florence'' test sequence are shots of the two sides of the building at different angles, and there is actually no content overlap. However, our algorithm, fast, kaze and other algorithms still obtain matches because the local building structure on the two sides of the building is basically the same. These ''mismatches'' instead show the effectiveness of the proposed feature extraction and description algorithm, which can effectively match the same building structure. Some scholars have conducted in-depth research on the problem of ''mismatches'' between images of the same building structure [83], which can be avoided by implementing carefully designed postprocessing strategies. This problem is not the focus of this paper and will not be discussed further.

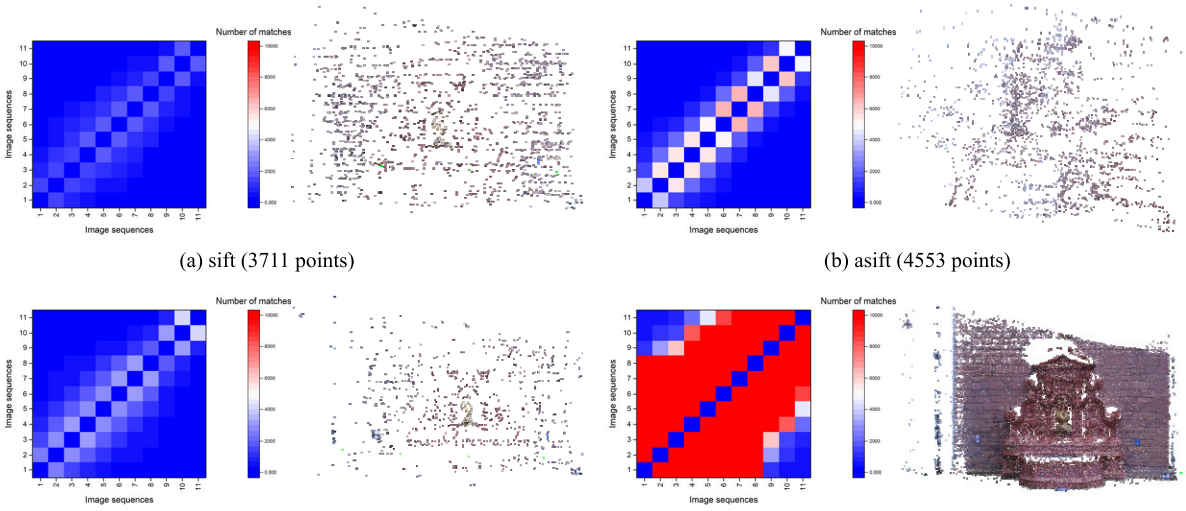
The ISPRS experiments show that the algorithm proposed in this paper can effectively address wide baseline matching problems in general indoor and outdoor scenes. For general wide baseline image pair matching, our algorithm is superior to current mainstream algorithms and can obtain a sufficient number of matching points.

V. APPLICATIONS

A sparse 3D reconstruction experiment was performed using a multiview image dataset to verify the performance of the proposed algorithm in practical applications. Experiments were performed using two public multiview image datasets: ''fountain'' and ''herzjesu'' [84]. The fountain dataset contains 11 multiview images, and the herzjesu dataset contains 8 multiview images. The images in each dataset used the same camera to shoot around the same target. The image content is rich in texture, and the viewpoint changes between adjacent images are appropriate. All images in each dataset are matched pair by pair, and the matching points are input to Theia [85] for sparse 3D reconstruction. The global SfM method is used for sparse 3D reconstruction, and the 3D reconstruction parameters are set using the default parameters recommended by Theia. In the image matching, step, sift, asift, orb, and our algorithm are used for image matching. The matching parameter settings are the same as before. The final sparse 3D reconstruction results for the two datasets are shown in Figures 20 and 21.

In Figures 20 and 21, (a), (b), (c), and (d) present the matching and 3D reconstruction results of sift, asift, orb, and our algorithm, respectively. The left of each subfigure shows the matrix of the number of matches between the images, and the horizontal and vertical axes represent the image serial numbers. The corresponding values are the numbers of matches between image pairs, and the colors from blue to red indicate the numbers from least to most, respectively. The right side of each subfigure shows the results of the sparse 3D reconstruction.

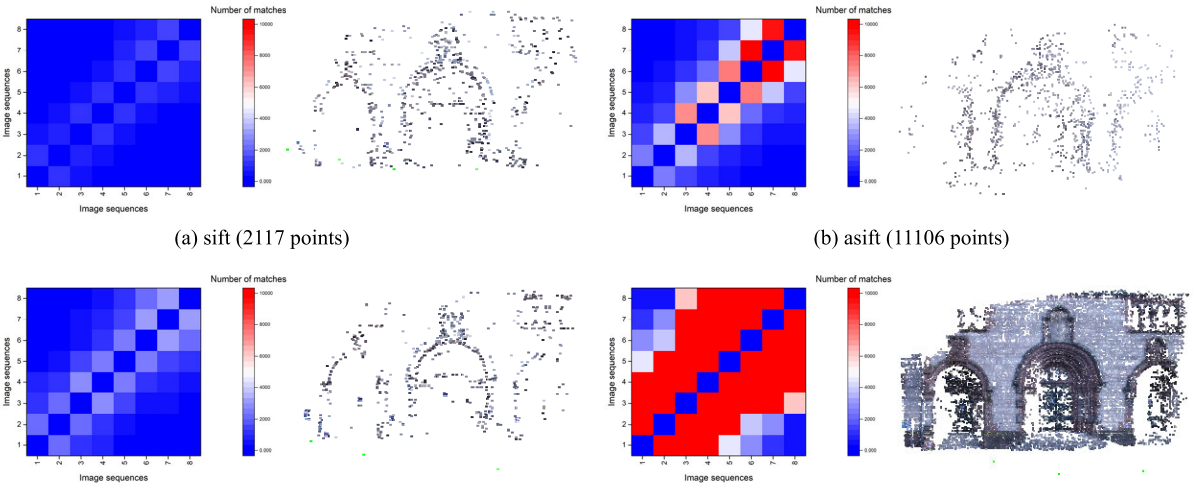
As shown in Figures 20 and 21, our algorithm can obtain many image-matching points. The algorithms that obtained the three next highest number of points are asift, orb, and sift.



(a) sift (3711 points) (b) asift (4553 points)

(c) orb (4591 points) (d) fsrb (325650 points)

FIGURE 20. Sparse 3D reconstruction results for the fountain dataset.



(a) sift (2117 points) (b) asift (11106 points)

(c) orb (4088 points) (d) fsrb (208710 points)

FIGURE 21. Sparse 3D reconstruction results for the herzjesu dataset.

The final 3D reconstruction results are significantly better than the results obtained by the other algorithms. In actual experiments, although the asift algorithm obtained a relatively sufficient number of matching points, the final 3D reconstruction results show clear structural errors and many noise points. Although the matching point pairs obtained by the orb algorithm are redundant to those of the sift algorithm, the completeness of the final scene reconstruction is inferior to that of the sift algorithm. The 3D reconstruction results obtained by our algorithm are superior to those obtained by mainstream algorithms in terms of scene integrity and reconstruction density, and these findings indicate the effectiveness of the algorithm in practical applications.

VI. CONCLUSION

Starting from the image-matching problem, this paper proposes a superpixel segmentation edge intersection strategy to detect interest points and an improved binary descriptor for feature description. Using many actual datasets for experiments, this study shows that the algorithm presented in this paper is far superior to current mainstream matching algorithms in terms of the number of final matches, and the number of final correct points increases by 2-5 times. This algorithm is equivalent to current mainstream algorithms in terms of matching efficiency, matching accuracy and success rate. This algorithm can adapt well to a variety of actual matching situations and can also match many correct points for wide baseline and weakly textured images. In future work, we will further design a targeted matching strategy for many interest points detected by the algorithm presented in this paper and optimize the overall calculation process to achieve real-time or near real-time processing.

REFERENCES

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

[2] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *Proc. Brit. Mach. Vis. Conf.*, 2002, pp. 36.1-36.10.
[3] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.
[4] P. F. Alcantarilla, A. Bartoli, and A. J. Davison, "KAZE Features," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 214–227.
[5] J. Lim and S. Lee, "Patchmatch-based robust stereo matching under radiometric changes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 5, pp. 1203–1212, May 2019.
[6] Z. Geng, B. Zhang, and D. Fan, *Digital Photogrammetry*. Beijing, China: Surveying and Mapping Press, 2010, pp. 80–88.
[7] H. Moravec, "Rover visual obstacle avoidance," in *Proc. Int. Joint Conf. Artif. Intell.*, 1981, pp. 785–790.
[8] Y. Li, S. Wang, Q. Tian, and X. Ding, "A survey of recent advances in visual feature detection," *Neurocomputing*, vol. 149, pp. 736–751, Feb. 2015.
[9] W. Förstner and E. Gülch, "A fast operator for detection and precise location of distinct points, corners and centers of circular features," in *Proc. Conf. Fast Process. Photogramm. Data. (ISPRS)*, 1987, pp. 281–305.
[10] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool, "A comparison of affine region detectors," *Int. J. Comput. Vis.*, vol. 65, nos. 1–2, pp. 43–72, Nov. 2005.
[11] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, 2008.
[12] D. Marimon, A. Bonnin, T. Adamek, and R. Gimeno, "DARTs: Efficient scale-space extraction of DAISY keypoints," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2416–2423.
[13] Z. Miao and X. Jiang, "Interest point detection using rank order LoG filter," *Pattern Recognit.*, vol. 46, no. 11, pp. 2890–2901, Nov. 2013.
[14] B. Li, R. Xiao, Z. Li, R. Cai, B.-L. Lu, and L. Zhang, "Rank-SIFT: Learning to rank repeatable local interest points," in *Proc. CVPR*, Jun. 2011, pp. 1737–1744.
[15] S. Salti, A. Lanza, and L. Di Stefano, "Keypoints from symmetries by wave propagation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 511–518.
[16] P. Alcantarilla, J. Nuevo, and A. Bartoli, "Fast explicit diffusion for accelerated features in nonlinear scale spaces," in *Proc. Brit. Mach. Vis. Conf.*, 2013, pp. 1281–1298.
[17] S. M. Smith and J. M. Brady, "SUSAN-a new approach to low level image processing," *Int. J. Comput. Vis.*, vol. 23, no. 1, pp. 45–78, 1997.
[18] E. Rosten and T. Drummond, "Fusing points and lines for high performance tracking," in *Proc. 10th IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 1, 2005, pp. 1508–1511.
[19] 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.
[20] 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.
[21] F. Mokhtarian and R. Suomela, "Robust image corner detection through curvature scale space," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 12, pp. 1376–1381, 1998.
[22] C. Vicas and S. Nedevschi, "Detecting curvilinear features using structure tensors," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 3874–3887, Nov. 2015.
[23] P.-L. Shui and W.-C. Zhang, "Corner detection and classification using anisotropic directional derivative representations," *IEEE Trans. Image Process.*, vol. 22, no. 8, pp. 3204–3218, Aug. 2013.
[24] A. Mustafa, H. Kim, E. Imre, and A. Hilton, "Segmentation based features for wide-baseline multi-view reconstruction," in *Proc. Int. Conf. 3D Vis.*, Oct. 2015, pp. 282–290.
[25] A. Mustafa, H. Kim, and A. Hilton, "MSFD: Multi-scale segmentation-based feature detection for wide-baseline scene reconstruction," *IEEE Trans. Image Process.*, vol. 28, no. 3, pp. 1118–1132, Mar. 2019.
[26] M. Awrangjeb, G. Lu, and C. S. Fraser, "Performance comparisons of contour-based corner detectors," *IEEE Trans. Image Process.*, vol. 21, no. 9, pp. 4167–4179, Sep. 2012.
[27] S. K. Ravindran and A. Mittal, "CoMaL: Good features to match on object boundaries," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 336–345.
[28] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 105–119, Jan. 2010.
[29] K. Yi, E. Trulls, V. Lepetit, and P. Fua, "Lift: Learned invariant feature transform," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 467–483.
[30] Y. Duan, J. Lu, J. Feng, and J. Zhou, "Learning rotation-invariant local binary descriptor," *IEEE Trans. Image Process.*, vol. 26, no. 8, pp. 3636–3651, Aug. 2017.
[31] V. Balntas, L. Tang, and K. Mikolajczyk, "Binary online learned descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 555–567, Mar. 2018.
[32] Y. Xu and F. Chen, "Recent advances in local image descriptor," *J. Image Graph.*, vol. 20, no. 9, pp. 1133–1150, 2015.
[33] G. Yu and J.-M. Morel, "ASIFT: An algorithm for fully affine invariant comparison," *Image Process. Line*, vol. 1, pp. 11–38, Feb. 2011.
[34] A. Bosch, A. Zisserman, and X. Munoz, "Scene classification using a hybrid generative/discriminative approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 4, pp. 712–727, Apr. 2008.
[35] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek, "Evaluating color descriptors for object and scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1582–1596, Sep. 2010.
[36] M. Brown and D. G. Lowe, "Automatic panoramic image stitching using invariant features," *Int. J. Comput. Vis.*, vol. 74, no. 1, pp. 59–73, Apr. 2007.
[37] E. Tola, V. Lepetit, and P. Fua, "DAISY: An efficient dense descriptor applied to wide-baseline stereo," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 815–830, May 2010.
[38] G. Takacs, V. Chandrasekhar, S. Tsai, D. Chen, R. Grzeszczuk, and B. Girod, "Unified real-time tracking and recognition with rotation-invariant fast features," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 934–941.
[39] M. Ambai and Y. Yoshida, "CARD: Compact and real-time descriptors," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 97–104.
[40] X. Boix, M. Gygli, G. Roig, and L. Van Gool, "Sparse quantization for patch description," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2842–2849.
[41] Z. Chen and S.-K. Sun, "A zernike moment phase-based descriptor for local image representation and matching," *IEEE Trans. Image Process.*, vol. 19, no. 1, pp. 205–219, Jan. 2010.
[42] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.
[43] Z. Guo, L. Zhang, and D. Zhang, "A completed modeling of local binary pattern operator for texture classification," *IEEE Trans. Image Process.*, vol. 19, no. 6, pp. 1657–1663, Jun. 2010.
[44] X. Tan and B. Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," *IEEE Trans. Image Process.*, vol. 19, no. 6, pp. 1635–1650, Jun. 2010.
[45] S. Murala, R. P. Maheshwari, and R. Balasubramanian, "Local tetra patterns: A new feature descriptor for content-based image retrieval," *IEEE Trans. Image Process.*, vol. 21, no. 5, pp. 2874–2886, May 2012.
[46] R. Maani, S. Kalra, and Y.-H. Yang, "Rotation invariant local frequency descriptors for texture classification," *IEEE Trans. Image Process.*, vol. 22, no. 6, pp. 2409–2419, Jun. 2013.
[47] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary robust independent elementary features," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 778–792.
[48] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua, "BRIEF: Computing a local binary descriptor very fast," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1281–1298, Jul. 2012.
[49] A. Alahi, R. Ortiz, and P. Vandergheynst, "FREAK: Fast retina keypoint," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 510–517.
[50] A. Ziegler, E. Christiansen, D. Kriegman, and S. Belongie, "Locally uniform comparison image descriptor," in *Proc. Neural Inf. Process. Syst.*, 2012, pp. 1–9.
[51] S. Winder, G. Hua, and M. Brown, "Picking the best DAISY," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 178–185.
[52] M. Brown, G. Hua, and S. Winder, "Discriminative learning of local image descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 43–57, Jan. 2011.
[53] K. Simonyan, A. Vedaldi, and A. Zisserman, "Learning local feature descriptors using convex optimisation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1573–1585, Aug. 2014.
[54] T. Trzcinski, M. Christoudias, and V. Lepetit, "Learning image descriptors with boosting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 597–610, Mar. 2015.

[55] C. Strecha, A. M. Bronstein, M. M. Bronstein, and P. Fua, "LDAHash: Improved matching with smaller descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 1, pp. 66–78, Jan. 2012.
[56] T. Trzcinski and V. Lepetit, "Efficient discriminative projections for compact binary descriptors," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 7–13.
[57] T. Trzcinski, M. Christoudias, P. Fua, and V. Lepetit, "Boosting binary keypoint descriptors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2874–2881.
[58] X. Ren and J. Malik, "Learning a classification model for segmentation," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, 2003, pp. 10–17.
[59] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graphbased image segmentation," *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 167–181, 2004.
[60] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.
[61] L. Vincent and P. Soille, "Watersheds in digital spaces: An efficient algorithm based on immersion simulations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 6, pp. 583–598, Jun. 1991.
[62] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.
[63] J. Zhao, R. Bo, Q. Hou, M.-M. Cheng, and P. Rosin, "FLIC: Fast linear iterative clustering with active search," *Comput. Vis. Media*, vol. 4, no. 4, pp. 333–348, Dec. 2018.
[64] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman, "Patchmatch: A randomized correspondence algorithm for structural image editing," in *Proc. Int. Conf. Comput. Graph. Interact. Techn.*, vol. 28, no. 3, 2009, p. 24.
[65] D. Bailey, "Sub-pixel estimation of local extrema," in *Proc. Image Vis. Comput.*, 2003, pp. 414–419.
[66] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool, "A comparison of affine region detectors," *Int. J. Comput. Vis.*, vol. 65, nos. 1–2, pp. 43–72, Nov. 2005.
[67] J. Sturm, W. Burgard, and D. Cremers, "Evaluating egomotion and structure-from-motion approaches using the TUM RGB-D benchmark," in *Proc. IEEE/RJS Int. Conf. Intell. Robot Syst.*, Oct. 2012, pp. 1–7.
[68] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 573–580.
[69] ISPRS. (2019). *The ISPRS Data Set Collection*. [Online]. Available: https://www.isprs.org/data/default.aspx
[70] G. Levi and T. Hassner, "LATCH: Learned arrangements of three patch codes," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2016, pp. 1–9.
[71] F. Rombari and L. D. Stefano, "Interest points via maximal self-dissimilarities," in *Proc. Asian Conf. Comput. Vis.*, 2014, pp. 586–600.
[72] M. Agrawal, K. Konolige, and M. R. Blas, "Censure: Center surround extremas for realtime feature detection and matching," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 102–125.
[73] M. Elmar, D. H. Gregory, B. Darius, S. Michael, and H. Gerhard, "Adaptive and generic corner detection based on the accelerated segment test," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 183–196.
[74] D. Mishkin, J. Matas, and M. Perdoch, "MODS: Fast and robust method for two-view matching," *Comput. Vis. Image Understand.*, vol. 141, pp. 81–93, Dec. 2015.
[75] Z. Wang, B. Fan, and F. Wu, "FRIF: Fast robust invariant feature," in *Proc. Brit. Mach. Vis. Conf.*, 2013, pp. 1–12.
[76] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *Int. J. Comput. Vis.*, vol. 60, no. 1, pp. 63–86, 2004.
[77] Z. Wang, B. Fan, and F. Wu, "Local intensity order pattern for feature description," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 603–610.
[78] Z. Wang, B. Fan, G. Wang, and F. Wu, "Exploring local and overall ordinal information for robust feature description," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 11, pp. 2198–2211, Nov. 2016.
[79] J. Bian, W.-Y. Lin, Y. Matsushita, S.-K. Yeung, T.-D. Nguyen, and M.-M. Cheng, "GMS: Grid-based motion statistics for fast, ultra-robust feature correspondence," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2828–2837.
[80] O. Chum, J. Matas, and J. Kittler, "Locally optimized RANSAC," in *Proc. Joint Pattern Recognit. Symp.*, in Lecture Notes in Computer Science, 2003, pp. 236–243.
[81] H. Yang, S. Zhang, and Q. Zhang, "Least squares matching methods for wide base-line stereo images based on SIFT features," *Acta Geodaetica et Cartographica Sinica*, vol. 39, no. 2, pp. 187–194, 2010.
[82] E. Shen, D. Fan, and X. Shun, "Small baseline stereo matching method based on SGM and phase correlation," *J. China Univ. Mining Technol.*, vol. 44, no. 1, pp. 183–188, 2015.
[83] J. Heinly, E. Dunn, and J. M. Frahm, "Correcting for duplicate scene structure in sparse 3D reconstruction," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 780–795.
[84] C. Strecha, W. von Hansen, L. Van Gool, P. Fua, and U. Thoennessen, "On benchmarking camera calibration and multi-view stereo for high resolution imagery," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
[85] C. Sweeney. (2019). *Theia Multiview Geometry Library: Tutorial*. [Online]. Available: http://theia-sfm.org

YANG DONG received the B.S. and M.S. degrees in photogrammetry and remote sensing from Information Engineering University, in 2014 and 2017, respectively, where he is currently pursuing the Ph.D. degree in surveying and mapping under Prof. Q. Ma and Prof. D. Fan. His current research interests include image matching and 3D reconstruction.

DAZHAO FAN received the Ph.D. degree in photogrammetry and remote sensing from Information Engineering University, in 2007. He is currently a Professor with Information Engineering University. His current research interests include digital photogrammetry and its applications.

QIUHE MA received the B.S. and M.S. degrees in photogrammetry and remote sensing from the Institute of Surveying and Mapping, in 1983 and 1993, respectively. She is currently a Professor with Information Engineering University. Her current research interests include image mapping and its applications.

SONG JI received the Ph.D. degree in photogrammetry and remote sensing from Information Engineering University, in 2012. He is currently an Associate Professor with Information Engineering University. His current research interests include image processing and 3D reconstruction.