IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 19, 2020, accepted April 16, 2020, date of publication April 21, 2020, date of current version May 21, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2989157

Fast SIFT Feature Matching Algorithm Based on Geometric Transformation

YONGCHAO WANG[1], YIJUN YUAN[2], AND ZHAO LEI[2]
[1]Information Technology Center, Zhejiang University, Hangzhou 310027, China
[2]School of Computer Science, Zhejiang University, Hangzhou 310027, China

Corresponding author: Zhao Lei (cszhl@zju.edu.cn)

This work was supported in part by the key research and development plan of Zhejiang province (Grant No.2018C03051), the technological project of cultural relics conservation of Zhejiang province (Grant No.2018010), the natural science foundation of Zhejiang Province (Grant No.LY19F020049), the science and technology project of Zhejiang Province (Grant No.2019C03137) and the key scientific research base for digital conservation of cave temples of the national cultural heritage administration.

ABSTRACT Structure from Motion (SfM) is a series of methods for reconstructing scene structure (i.e.three-dimensional space points) and camera motion (i.e.camera pose) from an image set. In this paper, aimed at the low accuracy of the global reconstruction, the robustness to external points and the time-consuming incremental reconstruction, a fast, closed-loop and high precision reconstruction method is proposed. The method is based on the SIFT matching algorithm GeoMatch, which is constrained by geometric structure of the scene and by numerical and statistical characteristics of feature invariant scale transformation. Experiments show that GeoMatch outperforms both traditional tree-based and hash-based matching methods in terms of time and accuracy.

INDEX TERMS Global reconstruction, incremental reconstruction, three dimensional reconstruction.

I. INTRODUCTION

Motion recovery structure(Structure from Motion, SfM)is a series of methods to reconstruct scene structure (i.e. three-dimensional space points) and camera motion (i.e. camera pose) from image centralization. The typical methods are incremental reconstruction and global reconstruction. Firstly, for these two reconstruction methods, feature point matching is the most time-consuming stage. Especially when there is no prior information of matching order between images in the image set (i.e. the disordered image set), all image pairs need to be matched. Secondly, compared with global reconstruction methods, incremental reconstruction methods have the advantages of high reconstruction accuracy and robustness to external points, while its disadvantages are large time complexity caused by the selection of initial image pairs and unable to stop the loop. Compared with incremental reconstruction methods, global reconstruction methods have the advantages of fast reconstruction speed, accurate closed-loop, and its disadvantages are low reconstruction accuracy and robustness to external points. Fast, closed-loop and high precision reconstruction method is particularly important. Based on the hybrid formula proposed in 2017, this paper proposes a method aimed at improving the performance of feature point matching stage: a SIFT matching algorithm GeoMatch(Geometric structure and SIFT-based Matching algorithm)based on scene geometric structure constraints and numerical statistical features of feature invariant scale transformation(Scale Invariant Feature Transform, SIFT) is proposed. Experiments show that GeoMatch outperforms both traditional tree-based and hash-based matching methods in terms of time and accuracy.

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

II. RELEVANT WORK

The known Structure from Motion (SfM) technology can be roughly divided into two parts: one is the description and matching of feature points, the other is the calculation of motion and scene structure of two cameras.

A. DESCRIPTION AND MATCHING OF FEATURE POINTES

Feature point matching is essentially the Nearest Neighbor retrieval problem (NN). The nearest neighbor search has the following definition [1]: There is a non-empty point set T_D in D-dimensional space, if $|T_D| > 1$, then for $\forall q \in T_D$, there is $NN(q) = \underset{p_i \in T_D - q,\ 1 \le i \le |T_D|}{argmin} d(q, p_i)$, where d(·, ·) is a vector distance metric function [2], and Nearest Neighbor (NN) is the nearest neighbor function notation.

Generally, feature point matching methods can be divided into two categories: standard matching technologies and approximate matching technologies. The standard matching technologies of feature points include linear search and tree-based search. The basic idea of the linear search is to select the nearest vector by comparing the distances of all vectors in the set T_D for any query vector Q. The advantage of this method is that it is very simple to implement, but the disadvantage is that it is very time-consuming [3]. The query time complexity for a single feature point is $O(|T_D|)$. Because the time complexity of the linear search is too high, document [1] proposed Kd-tree for nearest neighbor search, and the time complexity of single record query can reach $O(\log|T_D|)$. The literature [4] proposed a tree K-means method for Nearest Neighbor retrieval. This method first classifies data sets into K classes by K-means method and then divides the K classes recursively. Although the original Kd-tree search has been greatly improved compared with the linear search and works well when the data dimension is low, the time performance is not ideal with the increase of dimension in actual use [5]. In many cases, the time complexity is even worse than the linear search [6]. Document [7] proposed a tree-like K-means similar to document [4], which only uses data points instead of clustering mean in clustering center.

In [8], a new data structure spill-tree is proposed, which allows data to appear repeatedly in the node's sub-nodes. However, experiments show that the temporal and spatial performance of multiple random Kd-trees is better than spill-tree.

In [9], the author proposed a fast method of searching the nearest neighbor in large-scale data.

The data structure used in this method is similar to that in document [4] but only a single leaf node can be accessed.

When the scene is very large and complex, which leads to a large number of descriptors, how to store a large number of descriptors for fast retrieval is a very significant problem. One is quantization [10], [11] and dimensionality reduction [3], [12], the other is hash-based method [13]. The hash-based method not only expresses the original data more compactly, but also calculates the distance on Hemingway metric space more efficiently than Lp norm.

B. MOTION RECOVERY STRUCTURE

At present, the main method of monocular sparse reconstruction is the motion recovery structure (SfM). According to the different methods of camera initial motion estimation, SfM can be divided into Incremental Reconstruction [14]–[16], Global Reconstruction [17]–[19] and Hybrid Reconstruction [20].

The main method of Incremental Reconstruction is to estimate camera motion and structure by adding images to a reconstructed system step by step. This estimation will iterate over and over again. With the increase of images, the more parameters need to be estimated and optimized, and the iteration will be slower and slower. Besides, the cumulative errors will lead to Scene Drift [21]. According to different process of adding images, Incremental Reconstruction can be divided into two kinds: one is to select several images as the initial image group for reconstruction, and then add new images for iteration; the other is to cluster the images into several groups, reconstruct these groups separately, and then incrementally merge these groups. The main processes of Incremental Reconstruction are internal parameter extraction, feature point extraction, feature point matching, feature point mismatching filtering, Fundamental Matrix, Essential Matrix, Triangulation, and Bundle Adjustment.

There are many motion estimation methods for Global Reconstruction. Documents [18], [19] uses a matrix decomposition method based on rank theory to obtain camera motion estimation and point depth estimation, but this kind of method has the disadvantage of tracking corresponding points on all frames. Literature [17] uses a linear fitting method to estimate camera rotation before camera translation, but the accuracy of this method is not high.

Compared with Incremental Reconstruction, Global Reconstruction reduces the number of iteration optimization, and the error will be globally or evenly dispersed in each camera without accumulating. However, because of the sensitivity to external points, the reconstruction accuracy is not as high as Incremental Reconstruction.

III. FAST SIFT FEATURE MATCHING ALGORITHM BASED ON GEOMETRIC TRANSFORMATION

In this paper, a fast SIFT feature matching algorithm (Geometry transformation-based Feature Matching, GeoMatch) based on geometric transformation is proposed. The algorithm is divided into two stages, the first stage is initialization, and the second stage is fast matching based on geometric transformation. The initialization stage is divided into two steps: the first step is to match QSearch with global descriptors from rough to fine, and the second step is to reduce the dimension of SIFT descriptors. A more detailed algorithm flow of GeoMatch is shown in Table 1.

TABLE 1. Imaging procedure of GeoMath algorithm.

Input	Set of feature points **A** and **B**
Imaging process	**Step 1**: initialization (1) Constructing Quadtree and calculating global descriptors for all regions. (2) For each feature **A** (3) Matching each feature **A** # in **A** with feature point descriptors. (4) Repeat step a-c until a certain number of matching pairs are found and **M** is added. (5) The basic matrix **F** and corresponding matrix **H** are computed on **M**, and the final transformation matrix **T** with less re-projection error is selected. **Step 2:** matching based on geometric transformation: (1) For the unmatched features in **A**, **T** is used to quickly find the matched feature point descriptor and **M:F** matrix is added to search the epipolar line, **H** matrix is used to search the region.
Output	Matched feature index pair set **M**

Next, we will describe in detail the initialization phase and fast matching based on geometric transformation.

A. LINEAR SEARCH BASED ON CROSS QUADTREE

Two images with the same content should satisfy the polar geometric relationship. The polar geometric relationship between two-dimensional image points is described by the basic matrix F:

$$x_2^T F x_1 = 0 \tag{1}$$

In (1), x_1 and x_2 denote a matching pair of image pixels, $F = K_2^T E K_1$ is the basic matrix, where K_1 and K_2 are the internal parameter matrices corresponding to two images respectively, E is the essential matrix, describing the corresponding relationship between three-dimensional points, and T represents the transposition. This formula shows that if the image points x_1 and the basic matrix F are already in existence, a polar line $x^T F x_1 = 0$ can be determined in image 2. Solving the nearest neighbor is converted to solving the equation $x^T F x_1 = 0$. Since the solution obtained is not necessarily the desired feature point, it is necessary to search along the epipolar line to find the most matching feature point of the descriptor. In this way, the two-dimensional search problem is transformed into a one-dimensional search problem. However, if the scene described by an image is on a plane or an approximate plane, the projective mapping matrix (i.e. homography Matrix) can be used to describe the geometric relationship between points more accurately. Because the homography matrix is needed in feature matching, the derivation of the homography matrix is briefly introduced below.

Suppose the plane equation is $n^T P + d = 0$, n^T is a plane normal vector, D is a plane intercept and P is a 3D point. We can get the following equation.

$$-\frac{n^T P}{d} = 1 \tag{2}$$

Suppose there are two cameras: Camera 1 and Camera 2, whose projection transformation parameters are $K_1, R_1, t_1, K_2, R_2, t_2$, Combining (2), the following deductions are made:

$$\begin{aligned} p_2 &= K_2 (R_2 P + t_2) \\ &= K_2 \left(R_2 P + t_2 \left(-\frac{n^T P}{d} \right) \right) \\ &= K_2 \left(R_2 + t_2 \left(-\frac{n^T}{d} \right) \right) P \\ &= K_2 \left(R_2 + t_2 \left(-\frac{n^T}{d} \right) \right) R_1^T (K_1^{-1} p_1 - t_1) \end{aligned} \tag{3}$$

When calculating the homography matrix, we need to take an image as a reference image, and then take image 1 as a reference. There are $K_1 = E, t_1 = 0$, substitution (3), there are:

$$p_2 = K_2 \left(R_2 + t_2 \left(-\frac{n^T}{d} \right) \right) K_1^{-1} p_1 = H p_1 \tag{4}$$

In formula (4), $H = K_2 \left(R_2 + t_2 \left(-\frac{n^T}{d} \right) \right) K_1^{-1}$ is the corresponding matrix. In feature matching, because there is no prior knowledge of whether the feature points are coplanar or not, the matrix with small re-projection error between H and F is generally used as the transformation matrix in actual matching operations.

When using H and F to match, H and F need to be calculated beforehand. The calculation of H requires 4 pairs of matching points, and the calculation of F generally requires 8 pairs of matching points. Therefore, a certain number of matching points (20 to 30 pairs) needs to be calculated first. To get the initial matching point pairs as simple as possible, this paper uses a linear search. But the efficiency of the linear search is too low, so a linear search QSearch algorithm based on cross-Quadtree is proposed to speed up the search process.

The so-called cross-Quadtree means that the different areas divided by Quadtree are not completely independent, which have certain cross-areas as shown in Figure 1. Assuming a plane as shown by dotted lines, the plane is evenly divided by dotted lines and the upper left area is marked by black dotted lines. However, to increase matching accuracy (explained below), the actual coverage of the upper left area is marked by the blackened solid line. It can be seen that the four larger areas intersect with each other.

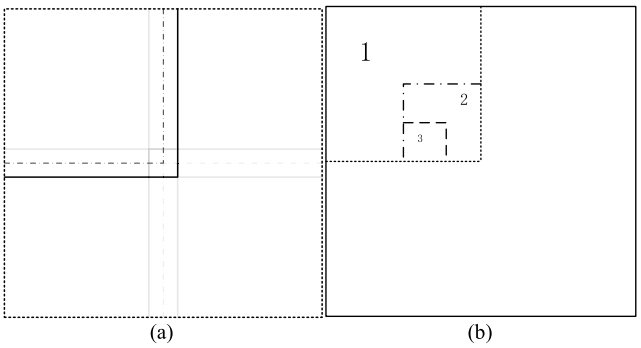


FIGURE 1. Cross-Quadtree diagram. (b) Search strategy diagram.

In order to match feature point descriptors, QSearch partitions the image plane several times and then calculates a global descriptor in each partitioned sub-region. All images are partitioned and the descriptor is computed in this way. For the two images that need to be matched, a coarse-to-fine search strategy is adopted. Firstly, the global descriptor is matched to get the approximate matching region, and then the feature point descriptor is matched in the matching region. As shown in Figure 1, firstly, the matching of global descriptors is performed from four largest regions to obtain the most matched region 1 (the region marked by dots and lines); secondly, the most matched sub-region 2 (marked by dots and lines) is obtained from the four sub-regions of region 1; finally, the most matched sub-region 3 is obtained from the four sub-regions of region 2. After region 3 is obtained, feature point descriptors are matched in the region.

If there is no coverage between regions, the features appearing at the boundary of one image region are likely to be transferred to adjacent regions in the next image. In order to solve this problem, when dividing space into sub-regions, sub-regions will overlap with each other partially.

B. REDUCED DIMENSION MATCHING OF SIFT FEATURE DESCRIPTORS

Natural and non-natural images. Natural images are two-dimensional snapshots of the real world. They are digital images captured by imaging equipment under certain illumination conditions. They have the characteristics of the gradual gray transition of pixels. Non-natural images are not reactions to the objective world, including images generated in a computer (such as cartoons, hand drawings, etc.) and realistic images generated by graphic technology. These images are not objective reactions of the real world, so there are two main problems: first, the gray level of pixels may change dramatically in a certain region; second, the geometric relationship of image response is not significant. Therefore, natural images should be selected as far as possible in three-dimensional image reconstruction. Natural images are used in all experiments in following chapters.

Planar and non-planar structures. The foreground object of the natural image is sometimes planar structure (such as the side of buildings, murals, etc.). Because all information of the planar structure is displayed on the two-dimensional plane without the information of the third dimension, using panoramic mosaics and other means will have a better observation effect in this case, and restoring the depth information of planar structure will not have practical application value. Therefore, the reconstruction of a single plane should be avoided as far as possible, and the image set should have a clear hierarchical structure. The images used in this paper are all-natural images with an obvious hierarchical structure.

In the process of studying SIFT characteristic values, it is found that all SIFT values (assuming that they are not normalized, i.e. the range of values is 0-255) tend to be smaller on the whole. To use this rule in nearest neighbor retrieval, the value of SIFT features is counted in this section.

In this section, the Oxford data set [22] is used for statistical feature experiments. The Oxford data set contains eight image sets and five image changes. These changes include blurring change, viewpoint change, distances and rotation change, illumination change and JPEG compression losses to varying degrees. Each image set contains six images and five transformation files, which are img1.ppm-img6.ppm and H1to2p-H1to6p respectively. In H1to2p, the single strain transformation matrix from img1.ppm to img2.ppm is saved. For a pair of matching SIFT key points calculated in img1.ppm and img2.ppm, we can calculate the expected position in img2.ppm of the key point of img1.pmm by H1to2p. If the distance error between the expected position and the calculated position is within the set threshold, it is considered to be a correct match.

In this section, statistics of SIFT numerical frequency distribution and cumulative distribution are made for four scenarios in the Oxford data set. The statistical results are shown in Figure 2 and Figure 3.

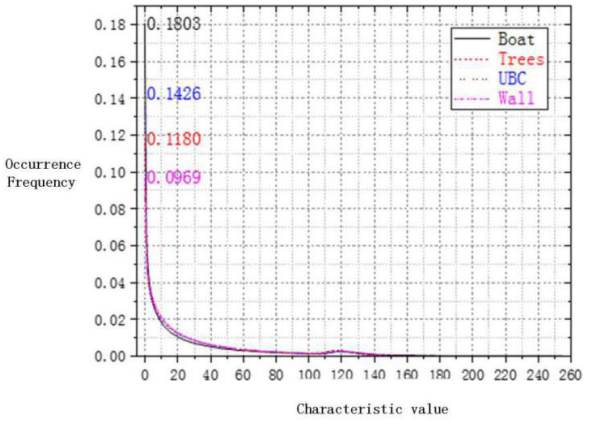


FIGURE 2. SIFT characteristic numerical frequency distribution.

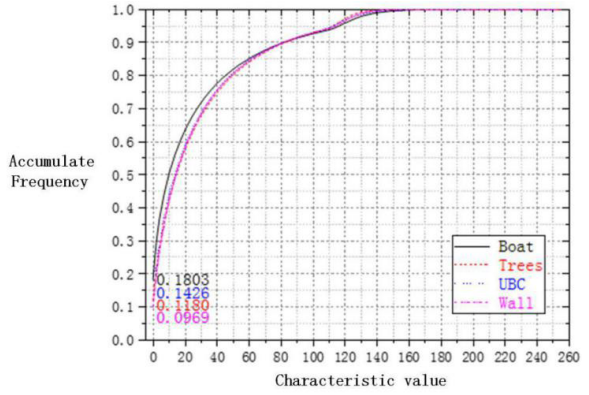


FIGURE 3. SIFT characteristic numerical frequency cumulative distribution.

As can be seen from Figures 2 and 3, the numerical frequency characteristics of SIFT features of different images are highly consistent. Firstly, from the frequency distribution, we can see that the frequency of numerical value 0-10 is above 2%, and that of 10-255 is below 2%. Secondly, it can be seen from the cumulative distribution that the probability of the value appearing in the interval [0,100] is about 90%. This shows that if a numerical value x is sampled from several SIFT features of a natural image, the probability of X appearing in the interval [0,100] is 90%. Finally, it can be seen from the frequency distribution chart that the larger value [101,255] will still appear with a lower probability (about 10%).

In summary, two statistical features of SIFT can be obtained: one is that the probability of arbitrarily sampled values from several SIFTs falling into the interval [0,100] is 90%, and the other is that the probability of arbitrarily sampled values from several SIFTs falling into the interval [101, 255] is 10%.

Based on the above two features, it can be further deduced that when matching features, 128 dimensions of one feature can be matched from large to small, and only a few of the small dimensions need to be matched. It's not necessary to match all the dimensions, to achieve the goal of dimensionality reduction.

IV. COMPLEXITY ANALYSIS

A. COMPLEXITY OF INITIALIZATION PHASE

The complexity of initialization depends on the implementation strategy. Specifically, it depends on the selection and calculation of global descriptors and the height of Quadtree. Assuming that the image size is w*h, the height of Quadtree is H, and the global descriptor is a gray histogram which takes $O\left(\frac{w^*h}{4^H}\right)$ to compute and requires $O(1)$ space. If the bottom-up global descriptor method is adopted, the time complexity of computing all global descriptors is as follows

$$O\left(\frac{w^*h^*}{4^H}4^H+4^{H-1}+4^{H-2}+\cdots+4\right)=O\left(w^*h+4^H\right),$$

Spatial complexity is $O\left(4+4^2+\cdots+4^H\right)=O(4^{H+1})$.

B. MATCHING COMPLEXITY BASED ON GEOMETRIC TRANSFORM

The complexity of this stage depends on the transformations used.

Assuming that the average number of vectors in the queried area is M, the average number of vectors in the queried area is N (finding the matching vectors of N vectors among M vectors), the dimension of the query is d, and the dimension of the feature is D.

Firstly, for each feature, it is necessary to select the dimension before selecting the eigenvalue from the 128-dimensional eigenvector, which depends on the specific algorithm. If relying on sorting algorithm, the heap sorting with the best average time complexity can be used to solve the problem with big D. The time complexity is O (128 * logD), because the constant 128 is larger than D, so the constant is not ignored here. The experiment uses a faster nth_element method with the average time complexity of O(128).

Secondly, each feature needs to search the nearest neighbors linearly on the selected d dimensions. Since the d dimensions used for each query may be different, Kd-tree acceleration is not appropriate here. The time complexity of the linear search is O(M*d).

Therefore, the optimal average time complexity of a query is O (128 + M * d).

Also, to query efficiently, the algorithm uses O (M*D) auxiliary space to save the transpose of the feature matrix in the database to index the value of one dimension of all features.

V. EXPERIMENTAL DESIGN

In this experiment design, SIFT features are extracted from two data sets: Oxford data set [22] and Tsinghua data set [23]. In order to evaluate the speed and accuracy of the proposed algorithm GeoMatch, two typical matching strategies are selected: linear search BruteForce and random Kd-tree matching. Besides, CasHash (Cascade Hash) proposed by literature [3] is added as a comparison according to the results of literature retrieval in the last three years.

The experimental environment is desktop PC, Windows 10 64 bit operating system, i5-4590 CPU, 8GB memory, C++ language. To be fair, the experiment shuts down all multithreaded acceleration, using only one thread. Because CasHash has no available CPU version on the public website, this experiment has compiled a single-threaded CPU version of CasHash.

A. EXPERIMENTS ON OXFORDS DATA SET

This experiment uses the standard Oxford data set [22] to evaluate the acceleration ratio [3] and accuracy [24] of four algorithms in SIFT feature matching.

The acceleration ratio of Kd-tree, CasHash, and GeoMatch to linear search BruteForce was measured. Accelerate Rate is defined as AR = t (BruteForce) / T (current algorithm). GeoMatch algorithm has several main parameters: the global descriptor used, the height of Quadtree and the dimension used for matching. In this experiment, the global descriptor uses the gray histogram, the height of Quadtree is 4 and the dimension setting standard is that GeoMatch algorithm has the same recall rate as CasHash. The experimental results are shown in Table 2.

TABLE 2. Acceleration ratio at the same recall rate.

	Boat	Trees	UBC	Wall
Kd-tree	10.64	13.93	3.59	8.97
CasHash	9.84	13.70	7.49	11.68
GeoMatch	12.61	14.91	13.30	14.77

As can be seen from Table 2, the proposed algorithm has the best time performance under the same recall rate when matching SIFT features.

The evaluation criteria used in the algorithm are from the literature [3]. The accuracy and recall rates of the algorithm are tested under four scenarios (far-near rotation, ambiguity, JPEG compression, viewpoint change). According to the convention, the accuracy rate needs to be converted to the error rate, i.e. 1-Precision. The experimental results are shown in Fig. 4.

The results show that the proposed algorithm GeoMatch is equal to Kd-tree on the whole but significantly better than CasHash in matching accuracy and recall rate experiments for SIFT features.

B. EXPERIMENTS ON TSINGHUA DATA SET

The purpose of this experiment is to test the impact of different matching algorithms on the final dense reconstruction, mainly focus on the accuracy and completeness of the model [24]. Assuming that the reconstructed model is R, the real model is G, the set of points corresponding from

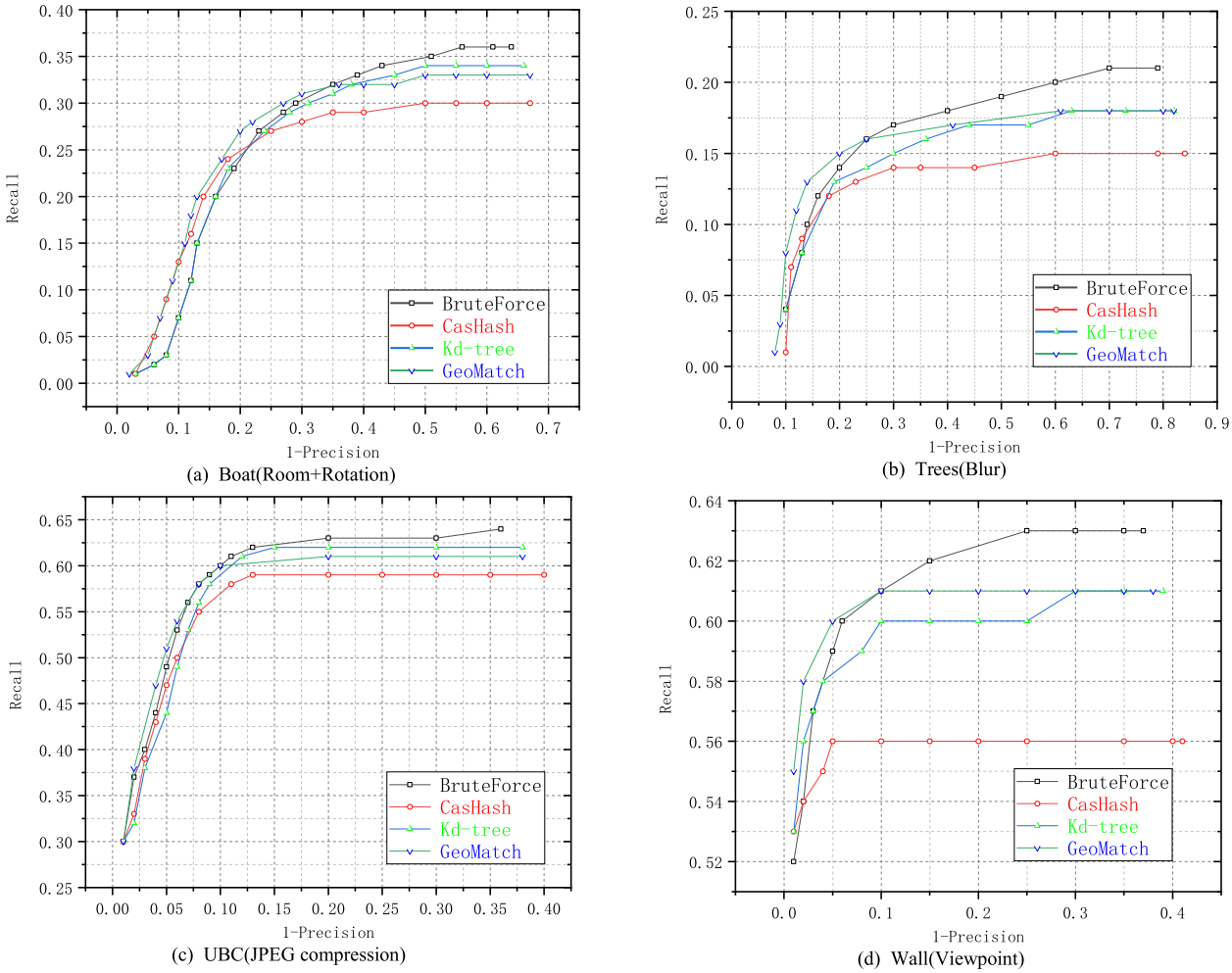


FIGURE 4. Recall rate curve on Oxford data set.

R to G is RG, and the set of points corresponding from G to R is GR (usually, RG and RG are not the same), RG should remove the case that points in multiple R correspond to points in one G. Accuracy is a dimensionless distance, which depends on how the model scales up and down. If R and G are aligned according to the actual size, then R and G are dimensionless. If R and G are relatively aligned, then there is dimensionless distance. It is defined as follows: If the set RG is not empty, the elements in RG are sorted from small to large according to the distance between corresponding points. With proportion X∈[0, 1], the distance between the corresponding points represented by the $\lfloor |RG|^* X \rfloor$ elements in RG. Among the formula, $|\cdot|$ denotes the number of elements in a set, which is also called the base of a set. And $\lfloor \cdot \rfloor$ is a downward integer operation. The concept of completeness is similar to accuracy, just by exchanging the contents of R and G.

The data set used in the experiment is the Tsinghua School and Tsinghua Life Sciences Building in document [22], the Campus data set with 1040 images [29], and Trafalgar, with 4591 images [30]. There are 193 and 102 images respectively in Tsinghua School and Tsinghua Life Sciences Building data set. Meanwhile, the data set gives the real value data of buildings obtained by Riegl-LMS-Z420i laser scanner. The main body of the program used in the experiment is Bunlder [26] andi PMVS2 [25], which only need to modify the key points of matching module. Because the coordinate scale and spatial orientation used between R and

TABLE 3. Time performance of three-dimensional reconstruction on Tsinghua data set.

Method	Data Set - Life Science Building		
	Time	Speedup Ratio	Point Number
Brute	48609s	1.00	**441803**
Kd-tree	5250s	9.26	419305
CasHash	1021s	47.61	410042
GeoMatch	**707s**	**68.75**	409989

Continued Table 3

Method	Data Set - Tsinghua School		
	Time	Speedup Ratio	Point Number
Brute	78669s	1.00	1078924
Kd-tree	12302s	6.39	**1108710**
CasHash	2801s	28.09	1035545
GeoMatch	**917s**	**85.80**	1022221

TABLE 4. Accuracy of three-dimensional reconstruction on Tsinghua data set.

Method	Data Set - Life Science Building									
	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Brute	2.24e-4	2.92e-4	4.78e-4	8.82e-4	9.17e-4	2.22e-3	2.37e-3	2.88e-3	4.45e-3	**16.66**
Kd-tree	1.92e-4	3.42e-4	5.05e-4	7.01e-4	9.53e-4	1.30e-3	1.85e-3	2.96e-3	6.06e-3	465
CasHash	3.01e-4	4.80e-4	5.94e-4	7.77e-4	0.82e-3	1.69e-3	2.85e-3	4.42e-3	8.89e-3	64.87
GeoMatch	**0.87e-4**	**2.76e-4**	**3.59e-4**	**6.62e-4**	**7.34e-4**	**9.97e-4**	**0.67e-3**	**1.99e-3**	**0.06e-2**	102.3
method	Data Set - Tsinghua School									
	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Brute	1.13e-4	2.64e-4	4.08e-4	8.87e-4	9.64e-4	2.88e-3	4.42e-3	7.14e-3	3.61e-2	18.30
Kd-tree	**1.59e-4**	3.15e-4	5.51e-4	9.45e-4	1.64e-3	2.88e-3	5.47e-3	2.14e-2	2.80e-2	6.77
CasHash	1.84e-4	**2.99e-4**	5.56e-4	0.89e-3	1.84e-3	3.80e-3	6.92e-3	4.19e-2	4.19e-1	22.23
GeoMatch	1.61e-4	3.01e-4	**5.01e-4**	**9.69e-4**	**1.02e-3**	**2.12e-3**	**3.87e-3**	**1.07e-2**	**2.01e-2**	**4.35**

TABLE 5. Integrity of three-dimensional reconstruction on Tsinghua data set.

Method	Data Set - Life Science Building									
	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Brute	**1.63e-4**	**2.88e-4**	3.32e-4	5.60e-4	8.01e-4	9.23e-4	0.17e-3	**1.19e-3**	7.22e-3	1.11
Kd-tree	1.74e-4	3.07e-4	4.46e-4	6.08e-4	8.09e-4	1.08e-3	1.49e-3	2.23e-3	4.32e-3	1.48
CasHash	1.93e-4	3.82e-4	4.72e-4	7.47e-4	9.98e-4	1.86e-3	2.09e-3	2.77e-3	6.43e-3	3.82
GeoMatch	1.83e-4	2.99e-4	**3.04e-4**	**5.19e-4**	**7.89e-4**	**8.99e-4**	**1.68e-3**	2.44e-3	**4.01e-3**	**1.01**
Method	Data Set - Tsinghua School									
	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Brute	1.33e-4	**2.18e-4**	3.42e-4	3.95e-4	7.61e-4	1.12e-3	2.40e-3	4.24e-3	**1.09e-2**	1.02
Kd-tree	**1.23e-4**	2.26e-4	3.56e-4	5.47e-4	8.51e-4	1.39e-3	2.41e-3	4.69e-3	1.35e-2	**2.68e-1**
CasHash	1.87e-4	2.34e-4	3.94e-4	5.87e-4	9.11e-4	2.21e-3	2.89e-3	1.16e-2	2.32e-2	2.97
GeoMatch	1.29e-4	2.51e-4	**3.27e-4**	**3.89e-4**	**7.37e-4**	**9.78e-4**	**1.83e-3**	**4.03e-3**	1.46e-2	6.54e-1

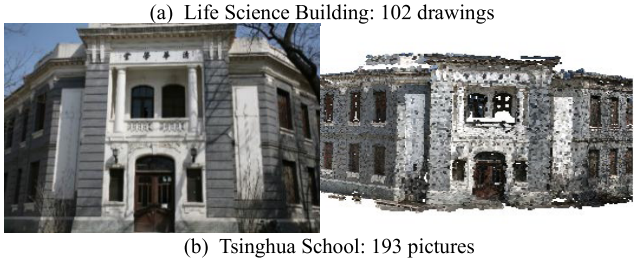
TABLE 6. Time performance of three-dimensional reconstruction on Campus data set.

Method	Time	Speedup Ratio	Point Number
Brute	128606s	1.00	**3416042**
Kd-tree	82500s	19.24	3193654
CasHash	70210s	37.67	3104422
GeoMatch	**6703s**	**58.75**	3110229

TABLE 7. Time performance of three-dimensional reconstruction on Trafalgar data set.

Method	Time	Speedup Ratio	Point Number
Brute	508669s	1.00	20634242
Kd-tree	423820s	16.32	**21108910**
CasHash	146789s	38.06	21085945
GeoMatch	**45170s**	**75.80**	21076401

(a) Life Science Building: 102 drawings



(b) Tsinghua School: 193 pictures

FIGURE 5. Result of 3D reconstruction on "Life Science Building" and "Tsinghua School".

G are inconsistent, the model needs to be scaled and aligned before calculating accuracy and completeness. In this experiment, point fixing in MeshLab [27] and ICP (Iterative Closest Points) methods are used for rotation and translation and scaling is used to align the scale of R with that of G.

The main variable of GeoMatch is the number of dimensions used. The main variable of CasHash is the number of bits encoded by hash bucket. The best of the number of bits is 8 according to literature [3].

The experimental results are shown in Tables 3, 4, 5, 6, 7. As can be seen from Tables 3, 6, 7, the algorithm proposed in this chapter has the best performance in terms of time complexity and acceleration ratio at the appropriate sacrifice of the number of point clouds. As can be seen from Tables 4 and 5, the algorithm presented in this chapter is the best in most sensitivity tests.

In Fig. 5, the left one is a real building and the right one is a reconstruction building. Intuitively speaking, the matching algorithm proposed in this paper can perform feature points matching well to complete the reconstruction task.

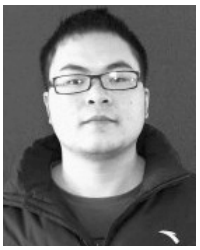
VI. TOTAL CONCLUSION

A fast matching algorithm GeoMatch for SIFT features is proposed based on the contrapole geometry and SIFT numerical distribution statistics of the scene. Experiments show that GeoMatch can greatly improve the matching speed of SIFT descriptors at the expense of a small number of three-dimensional points. At the same time, it can get a good guarantee of accuracy and integrity.

REFERENCES

[1] J. H. Friedman, J. L. Bentley, and R. A. Finkel, "An algorithm for finding best matches in logarithmic expected time," *ACM Trans. Math. Softw. (TOMS)*, vol. 3, no. 3, pp. 209–226, Sep. 1977.
[2] M. S. Charikar, "Similarity estimation techniques from rounding algorithms," in *Proc. 34th Annu. ACM Symp. Theory Comput. (STOC)*, 2002, pp. 380–388.
[3] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, Oct. 2005.
[4] K. Fukunaga and P. M. Narendra, "A branch and bound algorithm for computing k-nearest neighbors," *IEEE Trans. Comput.*, vol. C-24, no. 7, pp. 750–753, Jul. 1975.
[5] M. Muja, "Fast approximate nearest neighbors with automatic algorithm configuration," in *Proc. Int. Conf. Comput. Vis. Theory Appl. (Vissapp)*. 2009, pp. 331–340.
[6] C. Silpa-Anan and R. Hartley, "Localization using an imagemap," in *Proc. Australas. Conf. Robot. Automat.*, 2004, pp. 1–8.
[7] S. Brin, "Near neighbor search in large metric spaces," in *Proc. Int. Conf. Very Large Data Bases*. San Mateo, CA, USA: Morgan Kaufmann, 1995, pp. 574–584.
[8] T. Liu, A. Moore, A. Gray, and K. Yang, "An investigation of practical approximate nearest neighbor algorithms," in *Proc. Neural Inf. Process. Syst.*, 2004, pp. 825–832.
[9] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2006, pp. 2161–2168.
[10] T. Tuytelaars and C. Schmid, "Vector quantizing feature space with a regular lattice," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.
[11] S. Winder, G. Hua, and M. Brown, "Picking the best DAISY," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 178–185, doi: 10.1109/CVPR.2009.5206839.
[12] G. Hua, M. Brown, and S. Winder Discriminant, "Discriminant embedding for local image descriptors," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.
[13] P. Indyk and R. Motwani, "Approximate nearest neighbor: Towards removing the curse of dimensionality," in *Proc. 13th Annu. ACM Symp. Theory Comput.*, 1998, pp. 604–613.
[14] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *Proc. Int. Conf. Very Large Data Bases*. San Mateo, CA, USA: Morgan Kaufmann, 1999, pp. 518–529.
[15] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: Exploring photo collections in 3D," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 835–846, 2006.
[16] R. Gherardi, M. Farenzena, and A. Fusiello, "Improving the efficiency of hierarchical structure-and-motion," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 1594–1600.
[17] M. Farenzena, A. Fusiello, and R. Gherardi, "Structure-and-motion pipeline on a hierarchical cluster tree," in *Proc. IEEE 12th Int. Conf. Comput. Vis. Workshops, ICCV Workshops*, Sep. 2009, pp. 1489–1496.
[18] V. M. Govindu, "Combining two-view constraints for motion estimation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Dec. 2001, p. 2.
[19] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: A factorization method," *Int. J. Comput. Vis.*, vol. 9, no. 2, pp. 137–154, Nov. 1992.
[20] P. Sturm and B. Triggs, "A factorization based algorithm for multi-image projective structure and motion," in *Computer Vision—ECCV*, vol. 1065. Berlin, Germany: Springer, 1996, pp. 709–720.
[21] H. Cui, X. Gao, S. Shen, and Z. Hu, "HSfM: Hybrid structure-from-motion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2393–2402.
[22] K. Cornelis, F. Verbiest, and L. Van Gool, "Drift detection and removal for sequential structure from motion algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 10, pp. 1249–1259, Oct. 2004.
[23] *The Visual Geometry Group, Katholieke Universiteit Leuven, Inria Rhone-Alpes and the Center for Machine Perception*. Accessed: Jul. 15, 2007. [Online]. Available: http://www.robots.ox.ac.uk/~vgg/research/affine/
[24] *National Laboratory of Pattern Recognition Institute of Automation*, Chin. Acad. Sci., Beijing, China, 2010.
[25] *We Use Benchmark Database in 3D Reconstruction Dataset to Evaluate Our Algorithm. Benchmark Database Includes the Ground Truth Data From Laser Scanner, the Image Data and the Camera Projection Matrix of Each Buildings*. [Online]. Available: http://vision.ia.ac.cn/data/index.html
[26] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2006 , pp. 519–528.
[27] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 8, pp. 1362–1376, Aug. 2010.
[28] P. Cignoni and G. Ranzuglia. (2016). *Meshlab Visual Computing Lab-ISTI-CNR*. [Online]. Available: http://meshlab.net/
[29] Z. Cui and P. Tan, "Global Structure-from-Motion by similarity averaging," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 864–872.
[30] K. Wilson and N. Snavely, "Robust global translations with 1dsfm," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Berlin, Germany: Springer, 2014, pp. 61–75.

YONGCHAO WANG received the B.S. degree in photoelectric communication from the Zhejiang University of Technology, Hangzhou, China, in 1997, and the M.S. degree in circuits and systems from Zhejiang University, Hangzhou, in 2004. He has been a Senior Engineer with the Information Technology Center, Zhejiang University, since 1997. His research interests include 3D imaging acquisition, computer vision, machine learning, and so on.



YIJUN YUAN received the B.S. degree in computer science and technology from Southwest Petroleum University, Chengdu, China, in 2014, and the M.S. degree in computer science and technology from Zhejiang University, Hangzhou, China, in 2019. His research interests include 3D imaging acquisition and display.

ZHAO LEI received the B.S. degree in environmental engineering from Zhejiang University, Hangzhou, China, in 1998, and the Ph.D. degree in computer science and technology from Zhejiang University, Hangzhou, in 2009. Since 2010, he has been a Research Assistant with the Zhejiang University Network and Media Laboratory. His research interests include image processing and deep learning.