IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON BIOLOGICALLY INSPIRED IMAGE PROCESSING CHALLENGES AND FUTURE DIRECTIONS

Received March 26, 2019, accepted April 8, 2019, date of current version May 6, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2912647

Parallel K Nearest Neighbor Matching for 3D Reconstruction

MING-WEI CAO[1,2], LIN LI[1,2], WEN-JUN XIE[1], WEI JIA[1,2], (Member IEEE), ZHI-HAN LV[3], LI-PING ZHENG[1,2], (Member IEEE), AND XIAO-PING LIU[1,2]

[1]School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230009, China
[2]Anhui Province Key Laboratory of Industry Safety and Emergency Technology, Hefei, China
[3]School of Data Science and Software Engineering, Qingdao University, Qingdao, China

Corresponding authors: Wen-Jun Xie (wjxie@hfut.edu.cn) and Zhi-Han Lv (lvzhihan@gmail.com)

This work was supported in part by the National Key Research and Development Plan under Grant 2016YFC0800106, in part by the National Natural Science Foundation of China under Grant 61802103, Grant 61877016, and Grant 61673157, in part by the China Postdoctoral Science Foundation under Grant 2018M632522, in part by the Fundamental Research Funds for the Central Universities under Grant JZ2018HGBH0280 and Grant PA2018GDQT0014, and in part by the Natural Science Foundation of Shandong Province under Grant ZR2017QF015.

ABSTRACT In recent years, a 3D reconstruction based on structure from motion (SFM) has attracted much attention from the communities of computer vision and graphics. It is well known that the speed and quality of SFM systems largely depend on the technique of feature tracking. If a big volume of image data is inputted for SFM, the speed of this SFM system would become very slow. And, this problem becomes severer for large-scale scenes, which typically needs to capture several thousands of images to recover the point-cloud model of the scene. However, none of the existing methods fully addresses the problem of fast feature tracking. Brute force matching is capable of producing correspondences for small-scale scenes but often getting stuck in repeated features. Hashing matching can only deal with middle-scale scenes and is not capable of large-scale scenes. In this paper, we propose a new feature tacking method working in a parallel manner rather than in a single thread scheme. Our method consists of steps of keypoint detection, descriptor computing, descriptor matching by parallel k-nearest neighbor (Parallel-KNN) search, and outlier rejecting. This method is able to rapidly match a big volume of keypoints and avoids to consume high computation time, then yielding a set of correct correspondences. We demonstrate and evaluate the proposed method on several challenging benchmark datasets, including those with highly repeated features, and compare to the state-of-the-art methods. The experimental results indicate that our method outperforms the compared methods in both efficiency and effectiveness.

INDEX TERMS 3D reconstruction, K nearest neighbor, feature matching, structure from motion, parallel computing.

I. INTRODUCTION

In recent years, feature tracking include feature detection, descriptor matching, and outliers remove, has received much attention from the communities of computer vision and machine learning due to its many potential applications, such as image-based localization, image stitching, stereo matching [1], image retrieval [2], [3] and structure from motion (SFM) [4]. For example, the SFM use the feature correspondences that produced by feature tracking to recover sparse point clouds with respect to the scene and camera parameters. SFM is a set of technologies, which contains feature tracking [5], camera calibration [6], pose estimation, motion averaging [7], perspective-n-point (PnP) [8], registration [9], triangulation [10], and bundle adjustment [4]. Generally, SFM can be used to estimate camera parameters containing intrinsic and extrinsic parameters, and can also be used to recover point-cloud model of the scene from the given image collections. It has been proved to be one of the most effective 3D reconstruction approaches, and has been widely and successfully exploited in many 3D model-based applications including virtual reality [11], [12], augmented reality [13], city-scale modeling [14], navigation [15], smart city [16], [17], geographic information system [18], [19] and autonomous driving [20].

One of the most distinguished SFMs is Bundler [21], which is a standard implementation of incremental SFM and has good extendibility. Later, many excellent SFMs have also

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

2169-3536 © 2019 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

been proposed, such as ETH-3D, Visual SFM [22], Hyper SFM [23], ACTS [24], LS-ACTS [25], and COLMAP [26]. State-of-the-art SFM systems can produce accurate 3D point clouds for large-scale scenes [26], [27], but they are very time consuming. Thus, various strategies have been proposed to reduce the computation burden of SFM. For example, Wu *et al.* [28] proposed a multicore bundle adjustment method to optimize the point-cloud model and camera parameters for saving computation time. Crandall *et al.* [29] proposed a discrete continuous optimization method with Markov random filed (MRF) for large-scale structure from motion in parallel architecture, the proposed method does not have feature tracking routines, then significantly relieve computational burden. Bhowmick *et al.* [30] proposed graph partitioning-based approach to reconstruct point-cloud models, in which a divide and conquer method is used to partition the image data set into smaller sets or components that can be reconstructed independently. Sweeney *et al.* [31] proposed a distributed camera model for large-scale SFM, in which the incrementally adding one camera at a time to grow the reconstruction is replaced by a distributed camera. Although some effective methods have been proposed to accelerate the SFM system, the computational cost of SFM still needs to relieve especially in large-scale scenes.

According to the recent survey works made by Saputra *et al.* [32] and Ozyesil *et al.* [33], one of the most expensive steps in SFM is feature tracking. They also consider that as data grows, feature-tracking methods become more time-consuming. Therefore, a fast and effective feature-tracking method is urgently needed to handle large-scale 3D reconstruction with thousands of images having high resolution. To accelerate feature tracking, Guofeng *et al.* [25] proposed an effective non-consecutive feature tracking (ENFT) method for large-scale SFM. But, the ENFT heavily relies on the segmented-based coarse-to-fine scheme to improve the quality of SFM, it could not handle crowdsourcing image data. With the development of graphics process units (GPUs), many expensive operations could be accelerated using parallelization technique. For example, Sinha *et al.* [34] implemented KLT-GPU method with CUDA to improve the efficiency of original KLT, resulting in a significant acceleration. Moreover, depth image-based 3D reconstruction methods have already exploited GPUs to accelerate time-consumed operations, such as fusion of depth map and RGB image. For example, BundleFusion [35], dense fusion and mapping [36] and Parallel Kinect Fusion [37] all run on GPU devices for saving computation times. Thus, GPU have been proved to be an efficient approach to accelerate the computational expensive operations in the field of 3D reconstruction [38].

Inspired by GPU-acceleration, we propose a real-time feature tracking method based on parallel k nearest neighbor search (Parallel-KNN) for large-scale SFM. In the proposed feature tracking method, we first use ORB feature [39] to locate keypoints, and compute feature descriptors. Second, we design a Parallel-KNN search algorithm, and implement it in CUDA SDK for matching feature descriptors. Third, a distance-based test approach is proposed to remove outliers from initial matching collections that constructed by the naïve brute-force-match (BFM). Owing to the combination of above efficient and effective strategies, the proposed RTFT not only have fast speed but also has high matching precision. Our work is of broad interest to the 3D reconstruction, computer vision and computer graphics communities since many of the key steps in the proposed method are shared by other methods, which can also be accelerated on the GPU. In summary, the contributions of this work are summarized as follows:

(1) A GPU-accelerated k nearest neighbor matching called Parallel-KNN is designed and implemented in Nvidia CUDA SDK on GPU device, which can significantly improve feature matching speed and can also accelerate SFM-based large-scale 3D reconstruction with thousands of images having high resolution.
(2) A distance-based testing (DBT) approach is proposed to reject incorrect feature correspondences that created by traditional feature matching methods, such as BFM and KLT.
(3) Based on the proposed Parallel-KNN and DBT, we design a parallel pipeline for feature tracking, then resulting in a significantly acceleration on speed and highly matching precision.
real-time feature tracking method for large-scale 3D reconstruction based on SFM.

The rest of this paper is organized as follows. Related work is presented in Section 2. The parallel k nearest neighbor and the proposed real-time feature tracking (RTFT) method is described in Section 3. In Section 4, Comparative experiments conducted on several challenging datasets are presented. The conclusion and final remarks are given in Section 5.

II. RELATED WORK

In this section we will briefly revisit some existing works contain feature tracking and 3D reconstruction methods based on SFM technique to better understand the proposed method.

A. FEATURE TRACKING

Recently, feature tracking based on feature detection and matching framework (DMF) [40], [41] have received much attention from the communities of compute vision [42]–[46] and computer graphics [47]. For example, Zhang *et al.* [42] proposed a non-consecutive feature tracking method for SFM-based 3D reconstruction and simultaneous localization and mapping. This method uses brute-force-matching (BFM) scheme to match descriptors in feature database. To improve the robustness and speed of feature tracking method in large-scale scenes, Garrigues and Manzanera [48] proposed a mobile feature tracking method, which can be run on mobile device and also can recover dense trajectories. Lee and Hollerer [49] proposed an optical flow-based feature tracking method named hybrid feature tracking for

virtual reality, augmented reality, and navigation. However, this method is time consuming because computing optical flow is slow, and may produce few numbers of feature correspondences in the occlusion scenes. To improve the robustness of feature tracking in occlusion surroundings, Buchanan and Fitzgibbon [50] proposed an dynamic programming-based feature tracking method which uses interactive manner to deal with feature tracking drift problem, then significantly boosting the matching precision. In addition to desire matching score, the effectiveness of the feature tracking method is also improved due to the KD-Tree used.

Recently, someone dedicated to handle the ambiguous problem which raised by repeated feature or structure in scenes. For example, Zhang *et al.* [51] proposed an epipolar-constraint approach to handle incorrect feature matches, then resulting in a compact point-cloud model of the scene. However, the epipolar-geometry estimation may increase the computation time, and may decrease the speed of the SFM system when it was used. To accommodate image rotation and scale change, Wu *et al.* [52] proposed a viewpoint-invariant patches (VIP)–based feature tracking method for SFM-based 3D reconstruction in outdoors. The proposed method has an excellent performance according to the report executed on the several challenging benchmark datasets. Zach *et al.* [53] propose to use SURF having fast speed [54] to replace SIFT to obtain significant acceleration on speed. Svärm *et al.* [55] proposed a graph matching-based feature tracking method to construct feature correspondences, in which the Gomory-Hu algorithm [56] is used to remove outliers from the initial matching collections. Roth *et al.* [57] proposed a wide-baseline image matching approach by projective view synthesis and geometric verification.

Although, the existing feature tracking methods, such as ROML [58], MODS [59] and RepMatch [60], have already obtained excellent performance on the small or middle scale surroundings, their performance including both matching speed and matching precision still needs to improve in large scale scenes with repeated structures. For example, Zhou *et al.* [61] hold that the ability of the existing feature tracking approaches is still essentially limited by the abrupt scale changes in images. Thus, they designed a scale-invariant image matching method to handle the very large-scale variation of view-points. The proposed method can work well in scale space by encoding the image's scale space into a compact multi-scale representation. Cui *et al.* [5] proposed am approximate yet efficient approach to construct the matching graph for SFM-based large-scale 3D reconstruction. The proposed GraphMatch does not require any expensive offline pre-processing phase to construct matching graph, then resulting in a significant acceleration. Shen *et al.* [62] proposed a graph-based consistent feature tracking method to handle the problems of completeness, efficiency and consistency in a unified framework. The proposed method uses a visual-similarity-based minimum spanning tree (MST) to chain all the input images. Then the MST can be incrementally expanded to construct locally consistent strong triplets. Finally, the global consistency is improved by reinforcing the large connected components. The proposed method performs an excellent performance on the duplicated scenes.

B. 3D RECONSTRUCTION

In recent years, the problem of recovering 3D geometry from image collections has drawn much attention from various research areas including computer vision, computer graphics, and topography. As a result, many 3D reconstruction approaches have been proposed for the various applications. Among those methods, one of the most efficient approach is SFM-based 3D reconstruction, which can not only recover sparse geometry but also can estimate camera parameters, so it is widely used in many practical applications. For instance, Snavely *et al.* [21] developed an excellent SFM system, named Bundler, based on the standard pipeline of incremental SFM, which consist of camera calibration, feature detection and matching also called feature tracking, pose estimation, triangulation, and bundle adjustment. The computing efficiency of Bundler is slow, although it has desire result for recovering geometry model. After analysis the designing ideology, we found that the main reason to restrict the computing efficiency of Bundler is SIFT [39] which is used to detect keypoints and compute descriptors. To improve the efficiency of Bundler, Zach [63] proposed SURF to detect keypoints and compute descriptors in feature tracking process. Based on this ideology, they developed a novel SFM system named ETH-3D, which has faster speed than that of Bundler according to the newest report in [32].

Recently, Dong *et al.* [24] developed an automatic camera tracking system (ACTS) based on keyframe extraction and recognition to estimate camera parameters and recover sparse geometry model. The ACTS uses a GPU-accelerate SIFT [64] to detect keypoints and compute descriptors, then resulting in a significantly acceleration on the speed of feature tracking. Based on the ACTS, Zhang *et al.* [51] developed a large-scale SFM system named LS-ACTS. for real-time applications, such as augmented realty and robotic navigation. Like ACTS, based on the thought of GPU-acceleration, Wu [22] developed an excellent SFM system named Visual SFM (VSFM), which has both desire robustness for recovering geometry model an fast speed. Moulon *et al.* [65] developed a global SFM system based on the global fusion of relative camera motions between images. Based on the optimized viewgraph, Sweeney *et al.* [66] developed an incremental SFM system named Theia-SFM for the purpose of producing compact and accurate point clouds in both indoor and outdoor scenes. The Theia-SFM is consist of several independent modules such as feature tracking and surface reconstruction for recovering mesh.

With the development of depth-camera such as Kinect V2, ASUS Xtion Pro and Intel RealSense, RGB-D datasets are easily to construct, and are widely used in 3D reconstruction [67]. For example, Xiao *et al.* [68] developed RGBD-based SFM system to construct dense point clouds from RGB-D image collections. Yu and Zhang [69] proposed an

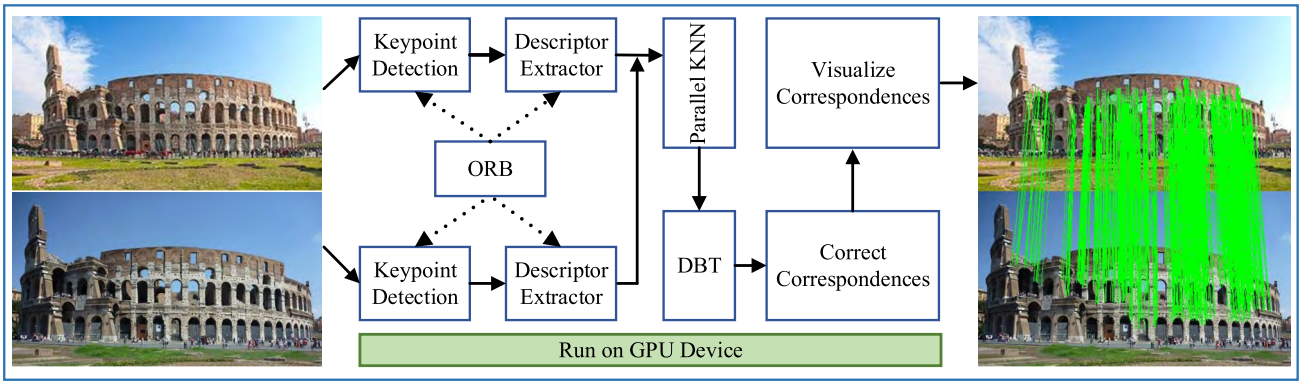


FIGURE 1. The flowchart of the proposed feature tracking method. (a) Input. (b) Pipeline of feature tracking. (c) Matching results.

approach for 3D reconstruction of indoor scenes based on RGB-D feature tracking and graph optimization, in which they detect loop closures based on the keyframe selection. Dai *et al.* [35] developed a real-time globally consistent 3D reconstruction approach, named BundleFusion, based on on-the-fly surface re-integration and depth-map fusion. The BundleFusion obtained a state-of-the-art result. Zeng *et al.* [70] proposed an octree-based depth-map fusion approach for real-time 3D reconstruction for RGB-D cameras. Bao and Savarese [71] proposed to use semantic information to help SFM system to recover geometry model, so they developed a semantic SFM (SSFM) system which can not only produce point clouds but also can recognize objects in scenes. Wang *et al.* [72] developed a dynamic SFM system, which can detect scene changes from image pairs. Kelly *et al.* [14] proposed a deep learning-based approach to reconstruct urban surface from UAVs with high resolution.

Although, some excellent 3D techniques have been proposed for various applications, their performance still needs to improve especially in large-scale outdoor scenes. Schöps *et al.* [73] hold that the fisheye is capable of reconstructing large-scale scenes in only a few minutes by simply walking through the scene. Thus, they proposed a method for reconstructing large-scale outdoor scenes through monocular fisheye camera, in which they exploit GPU device to compute depth maps for accelerating 3D reconstruction pipelines. Brilakis *et al.* [74] proposed a videogrammetric framework for acquiring 3D model of the outdoor scene, which uses a calibrated set of low-cost high resolution video cameras that is progressively traversed around the scene and aims to reconstruct a dense point-cloud model. Cui *et al.* [75] hold that insufficient feature correspondences may break the completeness of the reconstructed scene, they proposed a progressive SFM approach to handle the completeness, robustness and efficiency issues in a unified framework. By progressively performing the feature tracking and pose estimation, the method can produce a large number of redundant correspondences that can improve quality of the reconstructed point clouds. The most recently, Zhu *et al.* [76] hold that global structure from motion (GSFM) techniques have superior performance in both efficiency and accuracy than that of incremental SFM. They proposed a very large-scale global SFM at the scale of millions of high-resolution images. The global SFM is solved by the distributed framework that significantly improve the effectiveness and robustness of the large-scale rotation averaging.

III. PARALLEL KNN BASED FEATURE TRACKING

To improve the efficiency and accuracy of feature tracking for SFM-based 3D reconstruction, we propose a real-time feature tacking method (RTFT), which is based on the Parallel-KNN. The RTFT consists of three phrases: keypoint detection, descriptor computing, descriptor matching, and incorrect matches removing. The pipeline of the RTFT is depicted in Fig.1 where the green lines represents correct feature correspondences.

According to the pipeline of the RTFT depicted in Fig.1, for given two images, the ORB feature [77] is firstly utilized to locate keypoints and compute descriptors with respect to the located keyoints due to its fast speed and highly discrimination of the descriptor. Second, based on the similarity of feature descriptors, we use the implemented Parallel-KNN to match descriptors for producing visual correspondences that may include many outliers. Third, the distance-based testing (DBT) approach is proposed, and is utilized to remove incorrect correspondences from the initial matching collections, then resulting in a set of correct correspondences. To this end, the RTFT method has fast speed and desirable matching precision. The procedure of the RTFT is summarized in Algorithm 3, where loop operation is not required when calculating the distances between the query descriptor vectors and the reference descriptor vectors, so the computation time can be reduced significantly.

A. PROBLEM FORMULATION

For given a set of images I_s with n frames, $I_s = \{I_t | t = 1, \cdots, n\}$, the goal of feature tracking is to extract and match potential local features in all frames to construct a set

FIGURE 2. Visual correspondences of two views with repeated features, rotation and illumination variation, where some incorrect matches are produced.

of visual correspondences [78]. A visual correspondence K is defined as a tuple that include two similar keypoints in different images: $K = \left\langle k_{1,1}, k'_{2,1} \right\rangle$, where k_i represents the i^{th} image. For feature tracking, a feature from I_t to I_{t+1}, an invariant keypoint in I_t is represented as $k_{t,i}$ with descriptor $p\left(k_{t,i}\right)$. To determine if there is a corresponding keypoint $k_{t+1,i}$ with descriptor $p\left(k_{t+1,i}\right)$ in I_{t+1}, the 2nd closet ratio approach [59] is adopted.

For a query keypoint $k_{t,i}$ in I_t, we want to find the two nearest neighboring keypoints in I_{t+1} with respect to the Euclidean distance of the descriptor vectors and denote them as $I^1_{t+1}\left(k_{t,i}\right)$ and $I^2_{t+1}\left(k_{t,i}\right)$. Their corresponding descriptor vectors are represented as $p\left(I^1_{t+1}\left(k_{t,i}\right)\right)$ and $p\left(I^2_{t+1}\left(k_{t,i}\right)\right)$ respectively. As a result, the matching score for measuring the similarity of $k_{t,i}$ and $I^1_{t+1}\left(k_{t,i}\right)$ can be defined as

$$s = \frac{\left\| p\left(k_{t,i}\right) - p\left(I^1_{t+1}\left(k_{t,i}\right)\right)\right\|}{\left\| p\left(k_{t,i}\right) - p\left(I^2_{t+1}\left(k_{t,i}\right)\right)\right\|} \tag{1}$$

where s is used to measure the global distinctiveness of keypoint $k_{t,i}$ with respect to the ratio of the smallest descriptor distance and the second smallest one. If $s < \tau$, we consider that $I^1_{t+1}\left(k_{t,i}\right)$ is a potential candidate for $k_{t,i}$, and assign $k_{t+1,i} = I^1_{t+1}\left(k_{t,i}\right)$. According the report in [39], the feature matching approach could obtain a desirable result when $s \in [0.6, 0.8]$. In this paper, we chose 0.7 to s for achieving the best balance between matching precision and the number of matches. Base on the matching score, the standard feature matching approach for two views can be summarized in Algorithm 1 where none of post-process steps is used to remove outliers except to RANSAC.

However, the measure approach in practical can be easily affected by repeated features, image scale, image noise and lighting, which make it difficult to find reliable candidates for some keypoints even in the adjacent images. This problem usually makes the SFM produce ambiguous point-cloud model and even break the completeness and compactness of the reconstructed 3D models [79]. Fig. 2 illustrates the resulting visual correspondences constructed by purely descriptor matching, in which there are many incorrect correspondences. Thus, the post-process step, i.e., RANSAC, is usually used to reject outliers from the collection of initial visual correspondences. In addition to resulting outliers, the naïve feature matching by BFM result in highly computational cost especially in large-scale scenes. As a result, parallel procedure for feature matching is to be urgently needed for real-time feature tracking in the field of 3D reconstruction.

Algorithm 1 The Standard Two View Matching Approach

Input: I_1, I_2-two images.

Output: $\left\{\left\langle f_{1,1}, f'_{2,1}\right\rangle, \cdots, \left\langle f_{1,n}, f'_{2,n}\right\rangle\right\}$-a collection of visual correspondences.

Step 1:
for I_1 and I_2 independently do
Detect keypoints and compute descriptors using
local features such as ORB and SIFT.
end for

Step 2: Construct tentative correspondences for I_1 and I_2 using brute-force-match (BFM) approach and ratio test.

Step 3: Remove incorrect matches from the tentative correspondences using geometric verification, i.e., RANSAC, then resulting in a set of feature correspondences.

B. PARALLEL KNN SEARCH

In this subsection, we will discuss how to parallelize the traditional KNN algorithm in Nvidia CUDA SDK for computing the distance between the query descriptor vector and the reference descriptor vectors on GPU device. Unlike Hash-based approach [80] used for feature matching, the traditional KNN search alone still performs an "exhaustive search",

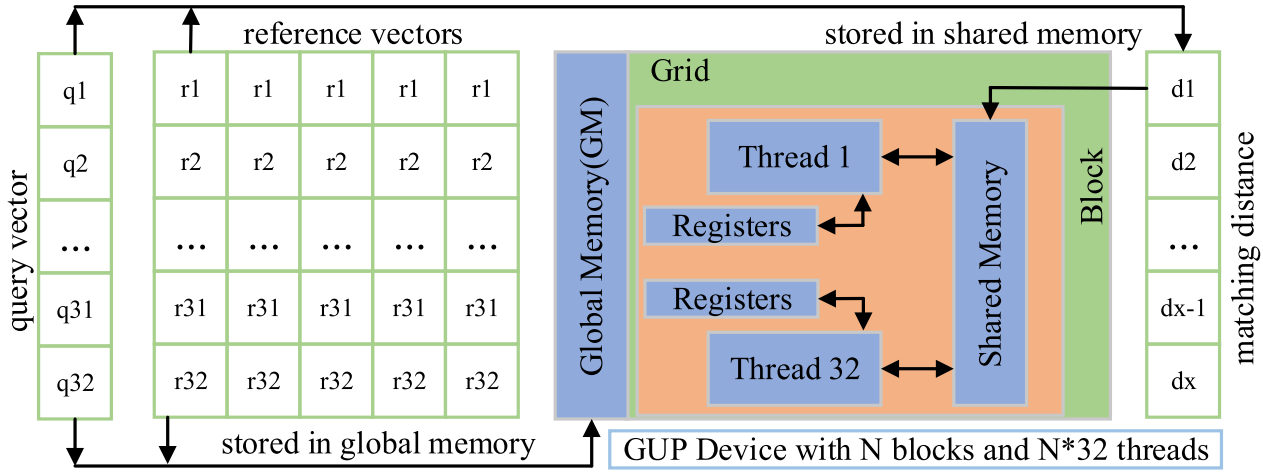


FIGURE 3. The diagram of distance computing on GPU.

meaning that the distance computing still requires comparing the query vector to every reference vector in the database.

However, the exhaustive computation is very time consuming. Thus, the key idea in this subsection is to design a parallel approach for the distance computing. Fig. 3 illustrates the diagram of distance computing between the query vector and the reference vectors on the GPU device, in which the query vectors and the reference vectors are stored in the global memory and the matching distance is temporarily stored in the shared memory for improving access time. Every matching distance computing is conducted on the single GPU thread independently and N blocks is used for the different keypoints, so the calculating for distance computing can be done in few times.

Let h be the dimension of the descriptor vector, so the CUDA kernel function of the distance computing defined as,

$$DC_{\mathrm{ker}nel} <<< N, h >>> (pq, pr) \tag{2}$$

where pq and pr are the pointers of the query vector and the reference descriptors in the GPU global memory, respectively. Since the ORB keypoint is 32 dimensional, namely $h = 32$. N is the number of keypoints in feature database. The kernel function totally calls N blocks and each block calls h threads. The distance computing needs to do $N * h$ accumulation. Owning to the GPU global memory access is relatively slow and GPU requires to access global memory repeatedly in the accumulation step, so the GPU shared memory is utilized to reduce access delay [80]. The shared memory is a kind of GPU cache having fast access speed. Thus, to save access time, the results of the distance computing are stored in the GPU shared memory temporarily when doing accumulation, so the computation speed is improved. Moreover, in order to maximize the efficiency of the distance computing in the CUDA framework and make every thread to do more works, the reference descriptors can be divided into a number of small vectors. Each block is responsible for the calculating of the small vectors, making fully use of the GPU's computing resources.

The parallel k nearest neighbor search scheme is summarized in Algorithm 2, where we respectively set the values of the GPU block and threads according to the number of keypoints and the dimensional of feature descriptor. Moreover, we make use of the shared memory to temporarily store the matching distance for further saving computational burden. As a result, the Parallel-KNN can achieve 10 times faster than that of the traditional KNN matching algorithm.

Algorithm 2 The Parallel k Nearest Neighbor Search Scheme

Input: v_q and M_r represent the query descriptor vector and the reference descriptor matrix.
Output: $\{m_1, \cdots, m_{32}\}$ -a set of matching score (matching distance).

Step 1: allocate the spaces of the global memory for the query vector and the reference matrix.
Step 2: Set the value for the input keypoints according to the number of input keypoints.
Step 3: Set the value for a single feature descriptor according to the length of the feature descriptor.
Step 4: Access the global memory to get the query descriptor and the reference descriptor vectors.
Step 5: Calculate matching distance on GPU threads and save the matching value in shared memory for improving computation time.
Step 6: Access the shared memory to get the matching matrix.

C. REMOVE OUTLIERS

A common problem in feature tracking is to remove outliers from the tentative matching collections. If many incorrect matches in the final collection of visual correspondences, then the SFM is easily to result in ambiguous point-cloud model, even cannot produce point-cloud model.

Algorithm 3 The Real-Time Feature Tracking Scheme

Input: $\{I_1, \cdots, I_N\}$-a set of images.
Output: $\{m_1, \cdots, m_x\}$-a collection of feature correspondences.
Step 1:
for I_i in $\{I_1, \cdots, I_N\}$ independently do
Detect keypoints and compute descriptors using ORB feature, then resulting keypoints $\{k_1, \cdots, k_y\}$ and corresponding feature descriptors $\{d_1, \cdots, d_y\}$.
end for
Step 2: Load the feature descriptors $\{d_1, \cdots, d_y\}$ into the global memory in GPU device.
Step 3: Allocate the space of the shared memory in GPU device to temporarily store the matching distance between the query descriptor and the reference descriptor vector.
Step 4: Construct tentative correspondences for $\{I_1, \cdots, I_N\}$ using the proposed Parallel-KNN method, so the matching collection is obtained as $\{m'_1, \cdots, m'_z\}$.
Step 5: Remove incorrect feature matches from the tentative correspondences, $\{m'_1, \cdots, m'_z\}$, using the proposed DBT method, so a new matching collection is obtained, $\{m_1, \cdots, m_x\}$, where the value of x may be less than that of y.
Step 6: Transfer the final matching collection
from the shared memory in GPU device to the CPU memory for the access from the host device.

Thus, to improve the quality of point-cloud model constructed by SFM system, the outliers must be removed in practical.

To address the problem of outlies removing, many approaches have been proposed in recent years, such as ratio test and cross check proposed by Lowe [39], geometric scheme based on RANSAC [81], in which the homography matrix or fundamental matrix should be estimated and is used to verify visual correspondences [82]. However, the above methods suffer easily from affine transformation. In the other word, for given two images, if the query image is affine transformed, then these methods cloud not remove incorrect matches from the tentative matching collection. Moreover, the RANSAC-based method has precondition that is the number of inliers must be greater than fifty percent, this condition is very rigorous. Recently, Bian *et al.* [83] proposed a statistic-based method called GMS—Grid-based Motion Statistics, for outlies removing. However, the GMS is heavily dependent on the number of keypoints, if there is few keypoints the GMS cloud not work well at al. Zhao *et al.* [84] proposed a vector filed consensus-based method (VFC) to remove outliers in feature tracking, for given a set of observed input-output pairs $s = \{(x_n, y_n) \in X \times Y\}_{n=1}^{N}$, the VFC is utilized to lean a mapping $f : X \rightarrow Y$ to fit the inliers, then result in a desirable matching precision. Unfortunately, the VFC is time consuming when processing high-resolution images.

According to deeply review recent feature matching works, we found that the distance between the first candidate and the second candidate keypoints for the query keypoint is very short in the correct matches estimated by 2nd nearest neighbor search. Conversely, if the distance between the two candidate keypoints is exceed a threshold, the match may incorrect. To deeply address the ambiguous problem in feature tracking, based on our observation, we propose a distance-based test (DBT) method to improve the feature matching method to produce high-confidence matching collection.

For given a set of query keypoints, $Q = \{q_1, \cdots, q_m\}$, and a set of reference keyoints, $R = \{r_1, \cdots, r_m\}$, we firstly use 2nd nearest neighbor search to estimate the tentative matching collection, the confidence of two candidate keypoints for the query keypoint can be calculated as

$$c = \frac{\|p(q_i) - p(r_i)\|}{\|p(q_i) - p(r_j)\|} \tag{3}$$

where $p(q_i)$ and $p(r_i)$ denote the descriptor for keypoint q_i and keypoint r_i respectively. If $c < 0.7$, the $\langle q_i, r_j \rangle$ is considered as a correct match. Looping the matching step, a set of initial matches can be obtained.

$$\hat{M} = \{\langle q_k, r_k \rangle | k \in [1, \min(m, n)]\} \tag{4}$$

Based on the initial matching collection $\hat{M}$, the homography matrix is easily estimated in four-point algorithm [85],

$$H_{q,r} = \begin{bmatrix} h_{1,1} & h_{1,2} & h_{1,3} \\ h_{2,1} & h_{2,2} & h_{2,3} \\ h_{3,1} & h_{3,2} & h_{3,3} \end{bmatrix} \tag{5}$$

thus, the geometric approach is utilized to verify the matching collection $\hat{M}$, namely

$$q'_k = H_{q,r} q_k \tag{6}$$

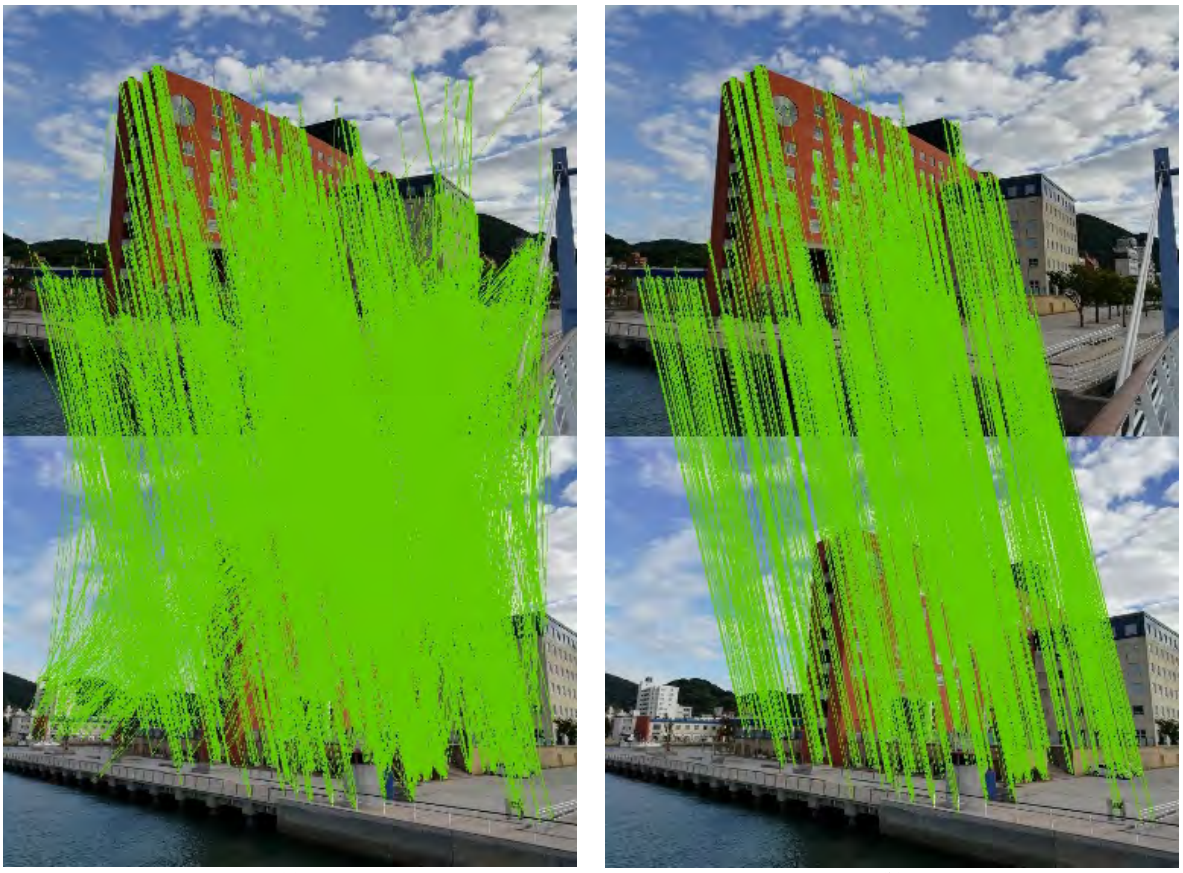
and

$$d = \|q''_k - r_k\| \tag{7}$$

where q''_k demotes the homogeneous coordinate of q'_k. If $d < \varepsilon$, the $\langle q_k, r_k \rangle$ is a correct match, thus an improved matching collection $\breve{M} = \{\langle q_l, r_l \rangle | l \in [1, \min(m, n)]\}$. Assuming r'_l is the second candidate keypoint for the keypoint q_l, the Euclidean distance between r_l and r'_l can be calculated by

$$d' = \|r_l - r'_l\| \tag{8}$$

If $d' < \varphi$, the $\langle q_l, r_l \rangle$ will consider as a correct match. Repeating the verification step, the final correct matching collection



(a) (b)

FIGURE 4. Remove outliers using the DBT method. (a) Initial matches. (b) Final matches.

can be obtained as

$$\tilde{M} = \left\{ \left\langle q_j, r_h \right\rangle | j, h \in [1, \min(m, n)] \right\} \tag{9}$$

Fig. 4 illustrates the initial matching and final matching, in which the former is generated by 2nd nearest neighbor search (2NN), the latter is generated by using the 2NN+ DBT. We can see clearly that our method can efficiently remove outliers.

IV. EXPERIMENTAL RESULTS

The proposed method is developed in C++, Nvidia CUDA SDK 10.0 and OpenCV SDK 3.4 and run on a PC with an Intel i5 processor having 3.4GHz and 32.0 GB of memory. To assess the performance of the RTFT method, we have evaluated the proposed method on the Repeating dataset (Ours) and building dataset [86] respectively, and also make a comprehensive comparison with KNN [21], ENFT [25] and MODS [59]. We have collected a set of images for evaluating feature tracking method. These form the Repeating Dataset with many repeated structures and repeated features on the surfaces of each image. The samples of Repeating dataset are depicted in Fig. 5. The dataset with keypoints and the source code of the RTFT method will be released when the paper is accepted.

A. EVALUATION ON REPEATING DATASET

We have evaluated the Parallel-KNN method on the Repeating dataset which is a challenging benchmarking dataset having 60 images, and contains some repeated features on the surface of architecture. In the experiment, we use ORB feature to detect keypoints, and also use it to compute descriptions for each image. And, we test the Parallel-KNN, and make a comparison with the state-of-the-art feature tracking methods. The assessment results are listed in Table 1, in which the traditional KNN method is selected as a baseline, and has the lowest matching precision and the highest computational cost. Comparison to KNN, the Kd-Tree has a little acceleration on matching speed, and also has a little improving on matching precision. The SiftGPU has a 11.49 times acceleration, and also has 0.689 matching score. Although the MODS has a significant improving on matching precision, it is too time consuming because of views synthesis. Both RepMatch and CODE have big improvement on aspect of matching precision, but their matching speed have only a little acceleration. The CasHash is a Local Sensitive Hashing-based feature tracking method for SFM-based 3D reconstruction, thus, it has a desirable matching precision and also has 3.10 times acceleration. The ENFT is also a GPU-acceleration feature tracking method for SFM,

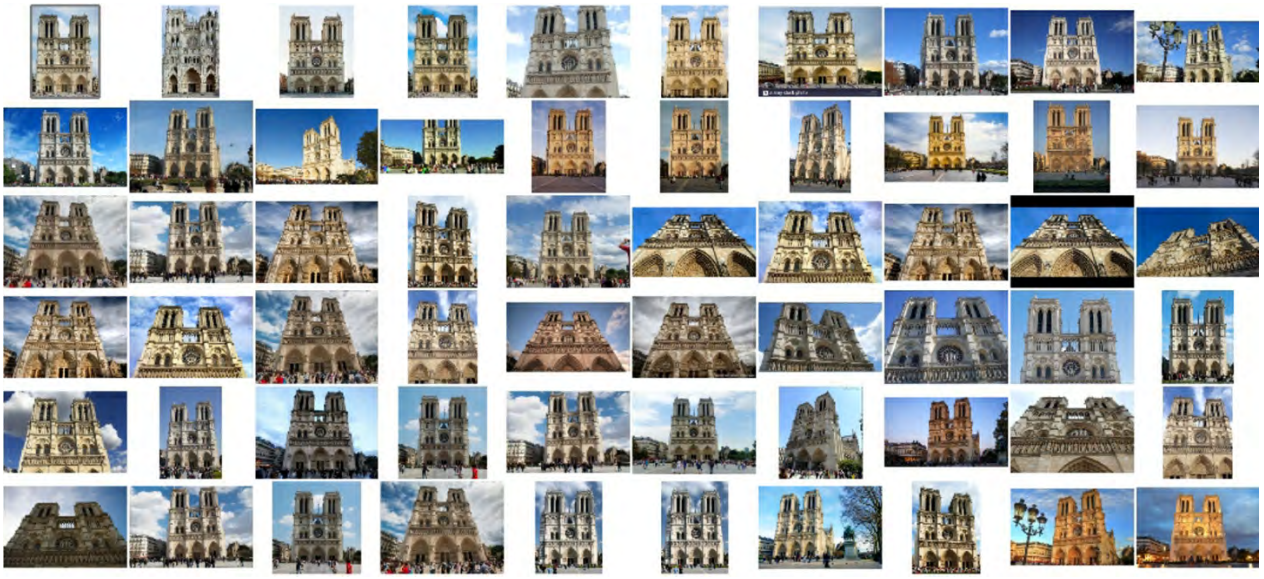


FIGURE 5. Samples from the Repeating dataset. Note that many repeated features are appeared on the surface of each image.

TABLE 1. The assessment results for repeating dataset with 60 pairs.

Method	Matching Precision	Time(s)	Speed(pairs/s)	Speedup	Grade
KNN [21]	0.463	1386.45	23.1	1.00x	✓✓
Kd-Tree [88]	0.476	896.38	14.94	1.55x	✓✓
SiftGPU [65]	0.689	120.87	2.01	11.49x	✓✓✓
ENFT [52]	0.892	110.65	1.84	12.55x	✓✓✓✓
MODS [60]	0.635	1107.65	18.46	1.25x	✓✓
CasHash [89]	0.873	446.73	7.45	3.10x	✓✓✓
RepMatch [61]	0.843	459.64	7.66	3.01x	✓✓✓
CODE [90]	0.866	596.82	9.95	2.32x	✓✓✓
RTFT (Ours)	0.935	78.36	1.31	17.63x	✓✓✓✓

FIGURE 6. Samples from the building dataset. Note that many repeating features are appeared on the surface of each image.

and use various strategies to decrease computational burden, thus it obtains a 12.55 times matching acceleration. Owing to the usage of Ratio-Test in the ENFT, then leading a desirable matching precision, namely 0.892. Among these feature tracking methods, the RTFT has significant improving on both aspects of matching precision and computational

FIGURE 7. Visual correspondences of each feature tracking method for the building dataset. (a) KNN. (b) MODS. (c) ENFT. (d) RTFT.

cost, which only consumes 1.31 second for matching image pairs. According to our statistics in experiment, the RTFT has 17.63 times faster than that of the traditional KNN approach. As a result, the RTFT could be consider as an excellent feature tracking method on aspects of efficiency and accuracy.

B. EVALUATION ON BUILDING DATASET

To assess the performance of the proposed RTFT method, we evaluate it on the building dataset, and compare it with the state-of-the-art methods. The building dataset is the latest benchmarking dataset that is constructed for evaluation of SFM-based 3D reconstruction. Fig. 6 illustrates the selected samples of the building dataset, which contains repeated structures and repeated features.

Fig. 7 illustrates the visual correspondences for each feature tracking method, in which the traditional KNN has the minimum number of matches, and conversely the proposed RTFT has the maximum number of matches. The number of visual correspondences for the MODS method is in the second place, but which has consumed higher computation cost because it needs to synthesize some virtual views in feature matching for improving matching precision. Although the ENFT has number of visual correspondences less than that of MODS, it has fast speed, and is capable of dealing with non-consecutive feature tracking. The RTFT not only has the fastest speed but also obtain the maximum visual correspondences. This assures that the point-cloud model produced by SFM has high quality when the RTFT is used. Fig. 8 presents computational costs for each compared method, in which the RTFT has the fastest speed, which achieves 10 times acceleration than that of the traditional KNN method. As a result, we can consider that the RTFT has the best performance on both computational cost and matching precision.

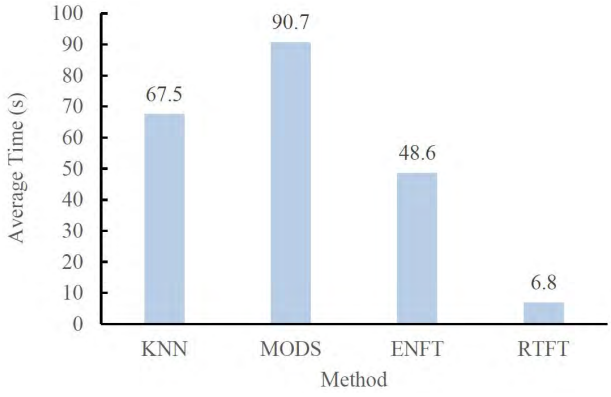
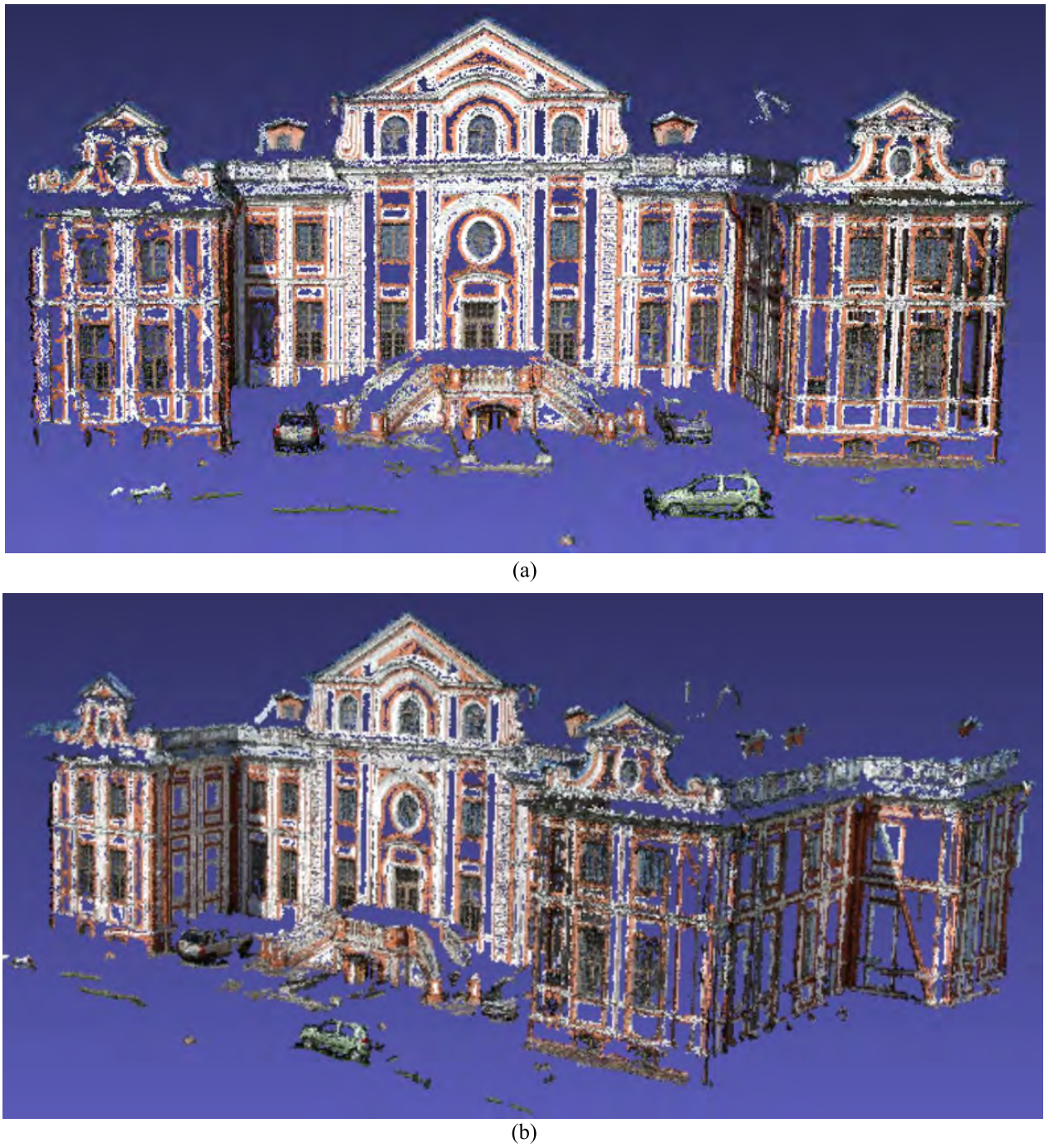


FIGURE 8. Averaging computational time for each feature tracking method that consumed on the building dataset.

We have integrated the RTFT in to ISFM system [8] to evaluate the practical ability of it. Fig. 9 illustrates the sparse



(a)

(b)

FIGURE 9. The sparse point-cloud model for the building dataset, and constructed by the ISFM system with the RTFT method. (a) Front of the reconstructed point-cloud model. (b) Side of the reconstructed point-cloud model.

point-cloud model produced by the ISFM system with the RTFT for the building dataset. The constructed point-cloud model has 496,090 vertices, and only 50 images was input to ISFM system. Thus, we can concern that the RTFT has excellent performance in practice.

V. CONCLUSION

To improve the quality of the point-cloud model that is produced by the SFM system, we designed a novel feature tracking method, and have implemented it in parallel architecture with Nvidia CUDA SDK, then resulting in a significant acceleration on computational cost. Specifically, the proposed RTFT method consists of three modules: 1) keypoint detection and descriptor computing, 2) feature matching, and 3) outliers removing. In the first stage, the ORB feature is used to find keypoints and obtain robust descriptions for the detected keypoints. In feature matching process, the Parallel-KNN is utilized to match feature descriptors for

decreasing the computational burden in large-scale 3D reconstruction where too many images are available. Moreover, the design logically behind of the Parallel-KNN is easily to generalize for other fields that also need parallel-computing technique. In the last step, we developed a novel approach to rectify the visual correspondences for resulting a set of correct feature matches, this method is only based on the Euclidean distance comparison, and is easily to implementation in programming language. Finally, we assess the RTFT method on three benchmarking datasets with some repeated structures and many repeated features, then result in a desirable performance in both feature matching precision and the quality of the point-cloud model.

In summary, the RTFT is versatile and expansible, which can be easily extended to other applications such as simultaneous localization and mapping, optical flow estimation, and robotics navigation. In the future, we will revise the Parallel-KNN and RTFT, and implement it on the multi-GPU devices for extreme fast acceleration on computation time.

REFERENCES

[1] J. Yang, K. Sim, W. Lu, and B. Jiang, "Predicting stereoscopic image quality via stacked auto-encoders based on stereopsis formation," *IEEE Trans. Multimedia*, to be published.

[2] J. Yang, B. Jiang, B. Li, K. Tian, and Z. Lv, "A fast image retrieval method designed for network big data," *IEEE Trans. Ind. Informat.*, vol. 13, no. 5, pp. 2350–2359, Oct. 2017.

[3] B. Jiang, J. Yang, Z. Lv, K. Tian, Q. Meng, and Y. Yan, "Internet cross-media retrieval based on deep learning," *J. Vis. Commun. Image Represent.*, vol. 48, pp. 356–366, Oct. 2017.

[4] M. Cao, S. Li, W. Jia, S. Li, and X. Liu, "Robust bundle adjustment for large-scale structure from motion," *Multimedia Tools Appl.*, vol. 76, no. 21, pp. 21843–21867, Nov. 2017.

[5] Q. Cui, V. Fragoso, C. Sweeney, and P. Sen. (2017). "GraphMatch: Efficient large-scale graph construction for structure from motion." [Online]. Available: https://arxiv.org/abs/1710.01602

[6] S. Ramalingam and P. Sturm, "A unifying model for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 7, pp. 1309–1319, Jul. 2017.

[7] A. Chatterjee and V. Govindu, "Robust relative rotation averaging," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 958–972, Apr. 2017.

[8] M. W. Cao, W. Jia, Y. Zhao, S. J. Li, and X. P. Liu, "Fast and robust absolute camera pose estimation with known focal length," *Neural Comput. Appl.*, vol. 29, no. 5, pp. 1383–1398, Jul. 2017.

[9] H. Lei, G. Jiang, and L. Quan, "Fast descriptors and correspondence propagation for robust global point cloud registration," *IEEE Trans. Image Process.*, vol. 26, no. 8, pp. 3614–3623, Aug. 2017.

[10] L. Kang, L. Wu, and Y.-H. Yang, "Robust multi-view L_2 triangulation via optimal inlier selection and 3D structure refinement," *Pattern Recognit.*, vol. 47, no. 9, pp. 2974–2992, 2014.

[11] N. Michael, M. Drakou, and A. Lanitis, "Model-based generation of personalized full-body 3D avatars from uncalibrated multi-view photographs," *Multimedia Tools Appl.*, vol. 76, no. 12, pp. 14169–14195, 2016.

[12] X. Li *et al.*, "WebVRGIS based traffic analysis and visualization system," *Adv. Eng. Softw.*, vol. 93, pp. 1–8, Mar. 2016.

[13] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. 6th IEEE ACM Int. Symp. Mixed Augmented Reality*, Nov. 2007, pp. 1–10.

[14] T. Kelly, J. Femiani, P. Wonka, and N. J. Mitra, "BigSUR: Large-scale Structured Urban Reconstruction," *ACM Trans. Graph.*, vol. 36, no. 6, p. 204, Nov. 2017.

[15] M. Colbert *et al.*, "Building indoor multi-panorama experiences at scale," in *Proc. ACM Siggraph Talks*, 2012, p. 24.

[16] Z. Lv, T. Yin, X. Zhang, H. Song, and G. Chen, "Virtual reality smart city based on WebVRGIS," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 1015–1024, Dec. 2016.

[17] Z. Lv *et al.*, "Managing big city information based on WebVRGIS," *IEEE Access*, vol. 4, pp. 407–415, 2016.

[18] X. Zhang, Y. Han, D. Hao, and Z. Lv, "ARGIS-based outdoor underground pipeline information system," *J. Vis. Commun. Image Represent.*, vol. 40, pp. 779–790, Oct. 2016.

[19] W. Wang *et al.*, "Spatial query based virtual reality GIS analysis platform," *Neurocomputing*, vol. 274, p. S0925231217306719, Jan. 2018.

[20] S. Song and M. Chandraker, "Robust scale estimation in real-time monocular SFM for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1566–1573.

[21] 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.

[22] C. Wu, "Towards linear-time incremental structure from motion," in *Proc. Int. Conf. 3D Vis. 3DV* 2013, pp. 127–134.

[23] K. Ni and F. Dellaert, "HyperSfM," in *Proc. 2nd Int. Conf. 3D Imag., Modeling, Process., Vis., Transmiss.*, 2012, pp. 144–151.

[24] Z. Dong, G. Zhang, J. Jia, and H. Bao, "Keyframe-based real-time camera tracking," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Oct. 2009, pp. 1538–1545.

[25] G. Zhang, H. Liu, Z. Dong, J. Jia, T.-T. Wong, and H. Bao, "Efficient non-consecutive feature tracking for robust structure-from-motion," *IEEE Trans. Image Process.*, vol. 25, no. 12, pp. 5957–5970, Dec. 2016.

[26] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4104–4113.

[27] H. Cui, X. Gao, S. Shen, and Z. Hu, "HSfM: Hybrid structure-from-motion," in *Proc. CVPR*, Jul. 2017, pp. 2393–2402.

[28] C. Wu, S. Agarwal, B. Curless, and S. M. Seitz, "Multicore bundle adjustment," in *Proc. CVPR*, 2011, pp. 3057–3064.

[29] D. J. Crandall, A. Owens, N. Snavely, and D. P. Huttenlocher, "SfM with MRFs: Discrete-continuous optimization for large-scale structure from motion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2841–2853, Dec. 2013.

[30] B. Bhowmick, S. Patra, A. Chatterjee, V. M. Govindu, and S. Banerjee, "Divide and conquer: Efficient large-scale structure from motion using graph partitioning," in *Proc. Asian Conf. Comput. Vis.*, 2014, pp. 273–287.

[31] C. Sweeney, V. Fragoso, T. Höllerer, and M. Turk, "Large scale SfM with the distributed camera model," in *Proc. 4th Int. Conf. 3D Vis.*, 2016, pp. 230–238.

[32] M. R. U. Saputra, A. Markham, and N. Trigoni, "Visual SLAM and structure from motion in dynamic environments: A survey," *ACM Comput. Surv.*, vol. 51, no. 2, pp. 1–36, Jun. 2018.

[33] O. Ozyesil, V. Voroninski, R. Basri, and A. Singer, "A survey of structure from motion," *Acta Numerica*, vol. 26, pp. 305–364, May 2017.

[34] S. N. Sinha, J.-M. Frahm, M. Pollefeys, and Y. Genc, "GPU-based video feature tracking and matching," in *Proc. Workshop Edge Comput. Using New Commodity Archit.*, 2000, pp. 189–196.

[35] A. Dai, S. Izadi, and C. Theobalt, "Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration," *ACM Trans. Graph.*, vol. 36, no. 4, p. 76a, 2017.

[36] M. Dou *et al.*, "Fusion4D: Real-time performance capture of challenging scenes," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 114:1–114:13, Jul. 2016.

[37] C. Zhang and Y. Hu, "CuFusion: Accurate real-time camera tracking and volumetric scene reconstruction with a cuboid," *Sensors*, vol. 17, no. 10, p. 2260, 2017.

[38] J. Yang, Y. Zhu, K. Li, J. Yang, and C. Hou, "Tensor completion from structurally-missing entries by low-TT-rankness and fiber-wise sparsity," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 6, pp. 1420–1434, Dec. 2018.

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

[40] M. Cao, W. Jia, S. Li, Y. Li, L. Zheng, and X. Liu, "GPU-accelerated feature tracking for 3D reconstruction," *Opt., Laser Technol.*, vol. 110, pp. 165–175, Feb. 2019.

[41] M. Cao *et al.*, "Fast and robust feature tracking for 3D reconstruction," *Opt., Laser Technol.*, vol. 110, pp. 120–128, Feb. 2019.

[42] G. Zhang, Z. Dong, J. Jia, T.-T. Wong, and H. Bao, "Efficient non-consecutive feature tracking for structure-from-motion," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 422–435.

[43] G. Zhang and P. A. Vela, "Good features to track for visual SLAM," in *Proc. CVPR*, Jun. 2016, pp. 1373–1382.

[44] B. Jiang, J. Yang, Z. Lv, and H. Song, "Wearable vision assistance system based on binocular sensors for visually impaired users," *IEEE Internet Things J.*, to be published.

[45] B. Jiang, J. Yang, H. Xu, H. Song, and G. Zheng, "Multimedia data throughput maximization in Internet-of-Things system based on optimization of cache-enabled UAV" *IEEE Internet Things J.*, to be published.
[46] B. Jiang, J. Yang, Q. Meng, B. Li, and W. Lu, "A deep evaluator for image retargeting quality by geometrical and contextual interaction," *IEEE Trans. Cybern.*, to be published.
[47] C. Peng, S. Sahani, and J. Rushing, "A GPU-accelerated approach for feature tracking in time-varying imagery datasets," *IEEE Trans. Vis. Comput. Graph.*, vol. 23, no. 10, pp. 2262–2274, Oct. 2016.
[48] M. Garrigues and A. Manzanera, "Real time semi-dense point tracking," in *Image Analysis and Recognition*. Berlin, Germany: Springer, 2012, pp. 245–252.
[49] T. Lee and T. Hollerer, "Hybrid feature tracking and user interaction for markerless augmented reality," in *Proc. IEEE Virtual Reality Conf.*, Mar. 2008, pp. 145–152.
[50] A. Buchanan and A. Fitzgibbon, "Interactive feature tracking using K-D trees and dynamic programming," in *Proc. CVPR*, Jun. 2006, pp. 626–633.
[51] G. Zhang, H. Liu, Z. Dong, J. Jia, T.-T. Wong, and H. Bao. (2015). "ENFT: Efficient non-consecutive feature tracking for robust structure-from-motion." [Online]. Available: https://arxiv.org/abs/1510.08012
[52] C. Wu, B. Clipp, X. Li, J.-M. Frahm, and M. Pollefeys, "3D model matching with viewpoint-invariant patches (VIP)," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
[53] C. Zach, M. Klopschitz, and M. Pollefeys, "Disambiguating visual relations using loop constraints," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1426–1433.
[54] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2006, pp. 404–417.
[55] L. Svärm, Z. Simayijiang, O. Enqvist, and C. Olsson, "Point track creation in unordered image collections using Gomory-Hu trees," in *Proc. 21st Int. Conf. Pattern Recognit. (ICPR)*, Nov. 2012, pp. 2116–2119.
[56] R. E. Gomory and T. C. Hu, "Multi-terminal network flows," *J. Soc. Ind. Appl. Math.*, vol. 9, no. 4, pp. 551–570, 1961.
[57] L. Roth, A. Kuhn, and H. Mayer, "Wide-baseline image matching with projective view synthesis and calibrated geometric verification," *J. Photogram., Remote Sens. Geoinf. Sci.*, vol. 85, no. 2, pp. 85–95, May 2017.
[58] K. Jia *et al.*, "ROML: A robust feature correspondence approach for matching objects in a set of images," *Int. J. Comput. Vis.*, vol. 117, no. 2, pp. 1–25, 2015.
[59] 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.
[60] W. Y. Lin, S. Liu, N. Jiang, M. N. Do, P. Tan, and J. Lu, "RepMatch: Robust feature matching and pose for reconstructing modern cities," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 562–579.
[61] L. Zhou, S. Zhu, T. Shen, J. Wang, T. Fang, and L. Quan, "Progressive Large Scale-Invariant Image Matching in Scale Space," in *Proc. ICCV*, Oct. 2017, pp. 2381–2390.
[62] T. Shen, S. Zhu, T. Fang, R. Zhang, and L. Quan, "Graph-based consistent matching for structure-from-motion," in *Computer Vision—ECCV*. Berlin, Germany: Springer, 2016, pp. 139–155.
[63] C. Zach, "ETH-V3D Structure-and-Motion software. 2010–2011," Ph.D. dissertation, School Comput. Sci., ETH Zurich, Zurich, Switzerland, 2010.
[64] C. Wu. (2011). *SiftGPU: A GPU Implementation of Scale Invariant Feature Transform*. [Online]. Available: http://cs.unc.edu/~ccwu/siftgpu
[65] P. Moulon, P. Monasse, and R. Marlet, "Global fusion of relative motions for robust, accurate and scalable structure from motion," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 3248–3255.
[66] C. Sweeney, T. Sattler, T. Hollerer, M. Turk, and M. Pollefeys, "Optimizing the viewing graph for Structure-from-motion," in *Proc. ICCV*, Dec. 2015, pp. 801–809.
[67] M. Cao *et al.*, "Fast and robust local feature extraction for 3D reconstruction," *Comput., Elect. Eng.*, vol. 71, pp. 657–666, Oct. 2018.
[68] J. Xiao, A. Owens, and A. Torralba, "SUN3D: A database of big spaces reconstructed using SfM and object labels," in *Proc. ICCV*, 2013, pp. 1625–1632.
[69] W. Yu and H. Zhang, "3D reconstruction of indoor scenes based on feature and graph optimization," in *Proc. ICVRV*, Sep. 2016, pp. 126–132.
[70] M. Zeng, F. Zhao, J. Zheng, and X. Liu, "Octree-based fusion for realtime 3D reconstruction," *Graph. Models*, vol. 75, no. 3, pp. 126–136, 2013.
[71] S. Y. Bao and S. Savarese, "Semantic structure from motion," in *Proc. CVPR*, 2011, pp. 2025–2032.
[72] T. Y. Wang, P. Kohli, and N. J. Mitra, *Dynamic SFM: Detecting Scene Changes From Image Pairs*. Hoboken, NJ, USA: Wiley, 2015, pp. 177–189.
[73] T. Schöps, T. Sattler, C. Häne, and M. Pollefeys, "Large-scale outdoor 3D reconstruction on a mobile device," *Comput. Vis. Image Understand.*, vol. 157, pp. 151–166, Apr. 2017.
[74] I. Brilakis, H. Fathi, and A. Rashidi, "Progressive 3D reconstruction of infrastructure with videogrammetry," *Autom. Construct.*, vol. 20, no. 7, pp. 884–895, 2011.
[75] H. Cui, S. Shen, W. Gao, and Z. Wang, "Progressive large-scale structure-from-motion with orthogonal MSTs," in *Proc. Int. Conf. 3D Vis.*, 2018, pp. 79–88.
[76] S. Zhu *et al.*, "Very large-scale global SfM by distributed motion averaging," in *Proc. CVPR*, 2018, pp. 4568–4577.
[77] 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.
[78] M. Cao, W. Jia, Z. Lv, W. Xie, L. Zheng, and X. Liu, "Two-pass K nearest neighbor search for feature tracking," *IEEE Access*, to be published.
[79] K. Wilson and N. Snavely, "Network principles for SfM: Disambiguating repeated structures with local context," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 513–520.
[80] T. Xu, K. Sun, and W. Tao, "GPU accelerated image matching with cascade hashing," in *Computer Vision*. Berlin, Germany: Springer, 2017, pp. 91–101.
[81] O. Chum and J. Matas, "Matching with PROSAC—Progressive sample consensus," in *Proc. CVPR*, 2005, pp. 220–226.
[82] Z. Zhang, Q. Shi, J. McAuley, W. Wei, Y. Zhang, and A. van den Hengel, "Pairwise matching through max-weight bipartite belief propagation," in *Proc. CVPR*, Jun. 2016, pp. 1202–1210.
[83] 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. CVPR*, Jun. 2017, pp. 2828–2837.
[84] J. Zhao, J. Ma, J. Tian, J. Ma, and D. Zhang, "A robust method for vector field learning with application to mismatch removing," in *Proc. CVPR*, 2011, vol. 32, no. 14, pp. 2977–2984.
[85] R. Hartley, and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2003.
[86] M. James and S. Robson, "Straightforward reconstruction of 3D surfaces and topography with a camera: Accuracy and geoscience application," *J. Geophys. Res., Earth Surf.*, vol. 117, no. F3, pp. 1–16, 2012.
[87] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," *VISAPP*, vol. 2, nos. 331–340, p. 2, 2009.
[88] J. Cheng, C. Leng, J. Wu, H. Cui, and H. Lu, "Fast and accurate image matching with cascade hashing for 3D reconstruction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1–8.
[89] W.-Y. Lin *et al.*, "CODE: Coherence based decision boundaries for feature correspondence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 1, pp. 34–47, Jan. 2018.

Authors' photographs and biographies not available at the time of publication.