applied sciences

MDPI

Article

Superpixel-Based Feature Tracking for Structure from Motion

Mingwei Cao [1,2], **Wei Jia** [1,2], **Zhihan Lv** [3], **Liping Zheng** [1,2] **and Xiaoping 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 230009, China

[3] School of Data Science and Software Engineering, Qingdao University, Qingdao 266071, China

* Correspondence: liu@hfut.edu.cn

Received: 4 July 2019; Accepted: 22 July 2019; Published: 24 July 2019

Abstract: Feature tracking in image collections significantly affects the efficiency and accuracy of Structure from Motion (SFM). Insufficient correspondences may result in disconnected structures and incomplete components, while the redundant correspondences containing incorrect ones may yield to folded and superimposed structures. In this paper, we present a Superpixel-based feature tracking method for structure from motion. In the proposed method, we first propose to use a joint approach to detect local keypoints and compute descriptors. Second, the superpixel-based approach is used to generate labels for the input image. Third, we combine the Speed Up Robust Feature and binary test in the generated label regions to produce a set of combined descriptors for the detected keypoints. Fourth, the locality-sensitive hash (LSH)-based k nearest neighboring matching (KNN) is utilized to produce feature correspondences, and then the ratio test approach is used to remove outliers from the previous matching collection. Finally, we conduct comprehensive experiments on several challenging benchmarking datasets including highly ambiguous and duplicated scenes. Experimental results show that the proposed method gets better performances with respect to the state of the art methods.

Keywords: feature tracking; superpixel; structure from motion; three-dimensional reconstruction; local feature; multi-view stereo

1. Introduction

In recent years, structure from motion (SFM) has received much attention from the computer vision and graphics communities. SFM is a collection of technologies, which is able to reconstruct 3D point-cloud model, and can estimate camera parameters (including intrinsic and extrinsic parameters) from image sequences [1]. A classic SFM framework usually consists of camera calibration, feature tracking, camera pose estimation, triangulation, and bundle adjustment [2]. It is well-known that SFM plays an important role in many research areas [3], such as augment reality, multi-view stereo [4], image-based localization [5], 3D reconstruction, image-based navigation [6], place recognition, autonomous driving, camera localization, and geographic information system (GIS) [7,8]. Based on different focuses, different types of SFM technologies have been proposed, such as incremental SFM, Global SFM, and Hybrid SFM [9].

Among existing Incremental SFMs, Bundler [10] is prestigious, which is a standard implementation of SFM, in which the scale invariant feature transform (SIFT) [11] is adopted to detect keypoints, then resulting in a highly computational cost. With the development of Graphics Process Unit (GPU), Wu et al. [12] implemented a GPU accelerated SIFT named SIFTGPU to reduce the computation time of feature tracking. Based on the SIFTGPU, Wu et al. developed a fast SFM system called Visual SFM (VSFM) [12], thus resulting in a significantly improvement in the aspect of time efficiency. In addition to the promising speed, the VSFM is user friendly due to its Graphic User Interface (GUI), and can not only work with multi-view stereo (MVS), such as the patch-based multi-view stereo (PMVS) [13], but also can be combined with Poisson surface reconstruction [14] to produce textured model of the scene. Dong et al. [15] developed a robust and real-time camera tracking system based on keyframes, called ACTS, for multi-view 3D reconstruction. The ACTS system consists of offline and online modules, both two modules work together can quickly recover the point-cloud model of the scene, and estimate camera's parameters containing intrinsic and extrinsic parameters. After a series of improvements on the ACTS, which is extended to work in large-scale surroundings [16]. Ni et al. [17] proposed a hierarchical SFM in a divide and conquer manner by using the bipartite graph structure of the scene. COLMAP [18] is an excellent incremental SFM implementation that contains many novel techniques such as scene augmentation, re-triangulation, and depth-fusion approach. All SFMs mentioned before use SIFT or SIFT's variants to locate keypoints and compute descriptors, other excellent local features may be ignored. Zach et al. [19] is the first time to use Speeded Up and Robust Features (SURF) [20] to detect keypoints and compute descriptors for feature for SFM, then leading a significantly boosting on speed.

Agarwal et al. [21] consider that feature tracking method may largely affect the quality of SFM. For example, if the captured image data contains few features, or many repeating features, the matching precision of feature tracking cloud be decreased significantly. To improve the problem of repeating features, some incomplete approaches has been proposed, such as loop constraint-based approach [22] where the observed redundancy in the hypothesized relations is used to reason the repetitive visual structures in the scene. Fan et al. [23] proposed to utilize the low distortion constraint approach to match pairs of interest points and then obtained feature correspondences from the matched pairs of interest points. Roberts et al. [24] found that the geometric ambiguities are usually caused by the presence of repeated structures and then proposed an expectation maximization (EM)-based algorithm that estimate camera poses and identifies the false match-pairs with an efficient sampling method to discover plausible data association hypotheses. Snavely et al. [25] presented a novel approach to solving the ambiguous problems by considering the local visibility structure of the repeated features and then presented a network theory-based method to score the repeated features. Recently, Ceylan et al. [26] designed an optimization framework for extracting repeated features in images of urban facades, while simultaneously calibrating the input images and estimating the 3D point-cloud model using a graph-based global analysis. Although some novel approaches have been proposed for the problem of ambiguous structures, they only work in the symmetric scenes.

To defend the ambiguous problem, we have paid much attention to investigate deeply the existing works [9,27,28], the following reasons may cause to produce ambiguous point-cloud model, that is repeated feature, untextured region where few keypoints can be found. As a result, we propose a superpixel segmentation-based feature tracking method for repeated and untextured scenes. Considering the simplicity, the superpixel-based feature tracking is abbreviated as "SPFT". The SPFT consists of feature detection, superpixel segmentation, and Markov Random Field (MRF)-based superpixel matching. Owing to the used superpixel segmentation, the SPFT can find sufficient keypoints in untextured scenes. Moreover, the SPFT can be considered as a general framework for feature tracking, which can be integrated with various local feature approaches such as SIFT, SURF, KAZE [29], and MSD [30]. Several challenging experiments made in Section 5 can efficiently prove the effectiveness and efficiency of the SPFT.

The main contributions of this work are summarized as follows:

- A Superpixel-based feature tracking method is proposed to locate keypoints and produce feature correspondences. The SPFT method has the fast speed and high matching confidence. Thus, SPFT can largely improve the quality of point-cloud model produced by SFM system.
- A combined descriptor extractor is proposed for producing robust descriptions for the detected keypoints. The proposed descriptor is robust to image rotation, lighting changes, and even can distinguish repeated features.

- We conduct a comprehensive experiment on several challenging datasets to assess the SPFT method, and comparison with the state-of-the-art methods. According to the evaluation, some valuable remarks are presented, which can be as a guide for developers and researchers.

The rest of this paper is organized as follows: related work is presented in Section 2. The proposed method is described in Section 3. In Section 4, a prototype 3D reconstruction system based on SFM is presented. Experimental results are given in Section 5. The conclusions and final remarks are given in Section 6.

2. Related Work

In this section, we will briefly review existing feature tracking methods and various SFM frameworks for better understanding the proposed feature tracking method.

2.1. Feature Tracking

Over the past years, many feature tracking methods has been proposed in the field of 3D reconstruction. The existing methods can be roughly divided into two categories, KLT-like approaches [31], and detection-matching framework (DMF)-based methods [32]. For the former, they compute displacement of keypoints between consecutive video frames when the image brightness constancy constraint is satisfied, and image motion is fairly small. However, KLT-like methods are only suitable to video data [33] in which each image frames have same resolution. To defend the drawbacks of KLTs, the DFM-based methods been proposed. In general, the DMF consists of keypoint detection, descriptor computing, and descriptor matching. For example, Snavely et al. [34] proposed a simple feature tracking method in which the SIFT and Brute-Force-Matching (BFM) were used to locate keypoints and to match descriptors respectively. Zhang et al. [35] developed a segment-based feature tracking method for camera tracking, the method can efficiently track non-consecutive video or image frames by the backend feature matching.

Moreover, researches proposed many novel local features to replace the SIFT in feature tracking procedure, such as speed up robust features (SURF) [20], Oriented Fast and Rotated Brief (ORB) [36], Binary Robust Invariant Scalable keypoints (BRISK) [37], maximally stable extremal regions (MSER) [38], and KAZE [29], features from accelerated segment test (FAST) [39], AGAST [40] and center surround detectors (CenSurE) [41]. Among these detectors, FAST and AGAST have fast speed, which are widely used in some real-time environments such as large scale simultaneous localization and mapping (SLAM) systems [42]. But they easily suffer from image rotation due to the local feature without main direction. To address this issue, Leutenegger et al. [37] proposed the BRISK detector, which is an invariant version of AGAST in multiple scale spaces. Unfortunately, BRISK has a low repeatability, which can further aggravate the drift problem in the process of feature tracking. Recently, binary descriptor has attracted much attention from the field of 3D reconstruction, such as local difference binary (LDB) [43,44], learned arrangements of three patch codes descriptors (LATCH) [45], boosting binary keypoint descriptors (BinBoost) [46], fast retina keypoint (FREAK) [47], and KAZE [29], etc. However, these binary descriptors can easily produce same descriptor in the scene with repeating structures according to [43]. Thus, the resulting ambiguous descriptors may further aggravate the ambiguity of feature matching especially in outdoors.

In addition to ambiguity, the existing local features have expensive computational cost. Even for binary local features, such as ORB, BRISK, the computational costs are also very high in large-scale scenarios. To accelerate the feature tracking method, Wu et al. [48] developed a SIFTGPU routine, which is the parallel implementation of the SIFT on GPU devices, then the SIFTGPU can achieve 10 times acceleration than that of original SIFT. Thus, the SIFTGPU is widely used in various computer tasks including SFM, simultaneous localization and mapping (SLAM), and robotic navigation. Inspired by SIFTGPU, Graves et al. [49] developed KLTGPU routines using OpenCL, then resulting in a 92% reduction in runtime compared to a CPU-based implementation. Cao et al. [50] proposed a GPU-accelerated feature tracking (GFT) method for SFM-based 3D reconstruction, which has a 20 times faster than that of SIFTGPU. Xu et al. [51] designed a GPU-accelerated image matching method with improved Cascade Hashing named CasHash-GPU, in which a disk-memory-GPU data exchange approach is proposed to optimize the load order of data, so the proposed method is able to deal with big data. According to their experiments, the CasHash-GPU can achieve hundreds of times faster than the CPU-based implementation.

2.2. Structure from Motion

Recent years, many 3D multi-view 3D reconstruction systems based on SFM technique have been proposed. For example, Snavely et al. [10] designed and implemented an excellent 3D reconstruction system, called Bundler, to reconstruct spare point-cloud model from unordered image collections. In the Bundler system, the authors employ scale invariant feature transform (SIFT) [11] to detect keypoints and compute descriptors, and use brute-force matching (BFM) strategy to match descriptors for image pair. However, owing to the usage of SIFT and BFM, the Bundler system has high computation cost. To save the computation time for 3D reconstruction based SFM, Wu et al. [12] developed a Visual SFM (VSFM) system based on Bundler, which use SIFTGPU to detect keypoints and compute descriptors for saving computation time. Micusik et al. [52] presented a novel SFM pipeline, which estimates motion and wiry 3D point clouds from imaged line segments across multiple views. The proposed SFM system tackle the problem of unstable endpoints by using relaxed constraints on their positions, both during feature tracking and in the bundle adjustment stage. Sweeney et al. [53] introduced the distributed camera model for 3D reconstruction based on SFM technique, in which, the proposed model describes image observations in terms of light rays with ray origins and directions rather than pixels. As a result, the camera model can describe a single camera or multiple cameras simultaneously as the collection of all light rays observed.

Based on the successes in solving for global camera rotations using averaging technique, Kyle et al. [54] proposed a simple, effective method for solving SFM problems by averaging epipolar geometries. The proposed unstructured SFM system (1DSFM) can overcome several disadvantages of existing sequential SFM. Moulon et al. [55] proposed a novel global calibration approach based on the global fusion of relative motions between image pairs for robust, accurate and scalable SFM. After an efficient contrario trifocal tensor estimation, the authors define an efficient translation registration method to recover accurate positions. Besides accurate camera position, Moulon et al. use KAZE [29] feature to detect keypoints in feature tracking, then resulting in a high-precision score. Based on optimized viewgraph, Chris et al. [56] designed and implemented an excellent SFM system, named Theia-SFM, to produce compact and accurate point-cloud model for both indoor and outdoor scenes. To recover the location of an object, Goldstein et al. [57] designed a scalable SFM system by utilizing ShapeFit and ShapeKick, even in the presence of adversarial outliers. Cohen et al. [58] proposed a novel solution for 3D reconstruction based on SFM to reconstruct the inside and the outside of a building into a single model by utilizing the semantic information, in which, novel cost function is proposed to determine the best alignment. To solve the degeneracies introduced by rolling shutter camera models, Albl et al. [59] show that many common camera configurations such as cameras with parallel readout directions, become critical and allow for a large class of ambiguities in 3D reconstruction based on SFM technique.

With the development of the depth camera, such as Kinect and RealSense, many RGBD datasets are publicly available for 3D reconstruction. Xiao et al. [60] developed RGBD-SFM system to produce dense point cloud model from RGBD images. Recently, Cui et al. [61] hold that SFM methods can be broadly categorized as incremental or global according to their ways to estimate initial camera poses. They proposed a unified framework to tackle the issues of efficiency, accuracy, and robustness, and developed a hybrid structure from motion (HSFM) system.

3. SLIC Method

Superpixel was first proposed by Ren et al. [62], and was used for image segmentation. In general, a superpixel in the image is a group of pixels that have continuous depths. The following properties for the superpixel are generally desirable: Superpixels should adhere well to image boundaries, and Superpixels should be fast to compute, memory efficient, and simple to use. Therefore, in the recent years, many superpixel algorithms, such as simple linear iterative clustering (SLIC) [63], superpixels extracted via energy-driven sampling (SEEDS) [64], Lattices [65], and GMMSP [66], have been proposed for various applications.

In this paper, the superpixel algorithm is selected as a preprocess step to segment tiny regions, as shown in Figure 1, the SLIC is the best choice due to its two important properties: (1) The number of distance calculations in the optimization is dramatically reduced by limiting the search space to a region proportional to the superpixel size. This reduces the complexity to be linear in the number of pixels N and independent of the number of superpixels k. (2) A weighted distance measure combines color and spatial proximity, while simultaneously providing control over the size and compactness of the superpixels. By default, the only parameter of the SLIC algorithm is k, which is the desired number of approximately equally-sized superpixels. For a given color image in the CIELAB color space, to get superpixel segmentations the following steps are required:

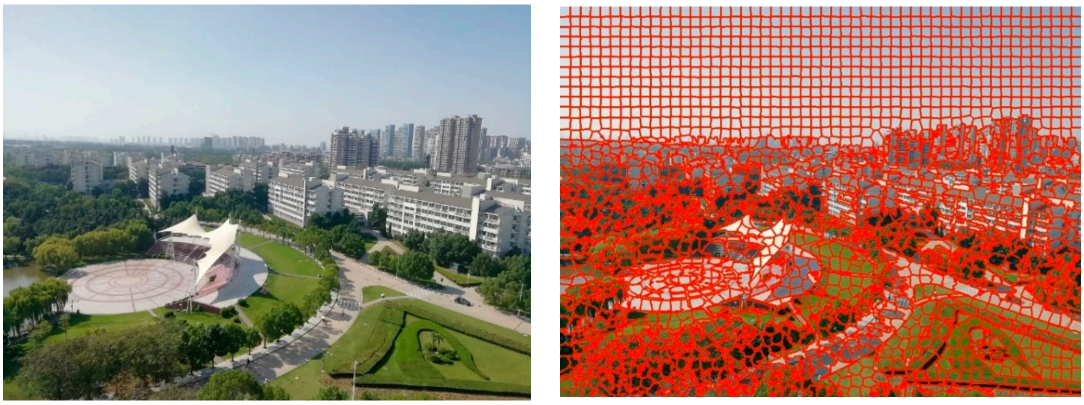
Step 1: Initialize cluster centers $C_i = \begin{bmatrix} l_i & a_i & b_i & x_i & y_i \end{bmatrix}^T$, which are sampled on the regular grid spaced S pixels apart.

Step 2: Move the cluster centers to the lowest gradient position in a 3×3 neighborhood.

Step 3: Compute the distance E between each cluster center C_k and pixel i in a $2S \times 2S$ region around C_k, if $D < d(i)$ then set $d(i) = D$, $l(i) = k$.

Step 4: Compute new cluster centers C'_k and residual error E.

Step 5: Repeat Step 3 and Step 4 until the residual E less than the threshold.



(**a**) Input (**b**) Visual results of superpixels

Figure 1. Illustration for superpixels.

4. The Proposed Method

To improve the quality of SFM, we propose a superpixel-based feature tracking method (SPFT), which consists of feature detection, descriptor computing, feature matching, and outliers removing. The flowchart of SPFT is depicted in Figure 2. For given an image, we first use SLIC algorithm to segment it to obtain non-overlapping regions C_i, and then use SIFTGPU feature detector to locate keypoints K_j, thus the total keypoints $K'_i = \left\{ C_i \cup K_j \middle| i = 1 \cdots N, j = 1 \cdots M \right\}$. Second, use ORB feature to describe the detected keypoints K'_i, and use SLIC labels to compute a patch-based description, then resulting a combined descriptor. Third, use k nearest neighboring method (KNN) to match the combined descriptors between the reference image and the query image. Finally, we use cross-check to remove incorrect matches from the KNN matching, then resulting in a set of correct correspondences as shown in Figure 2.

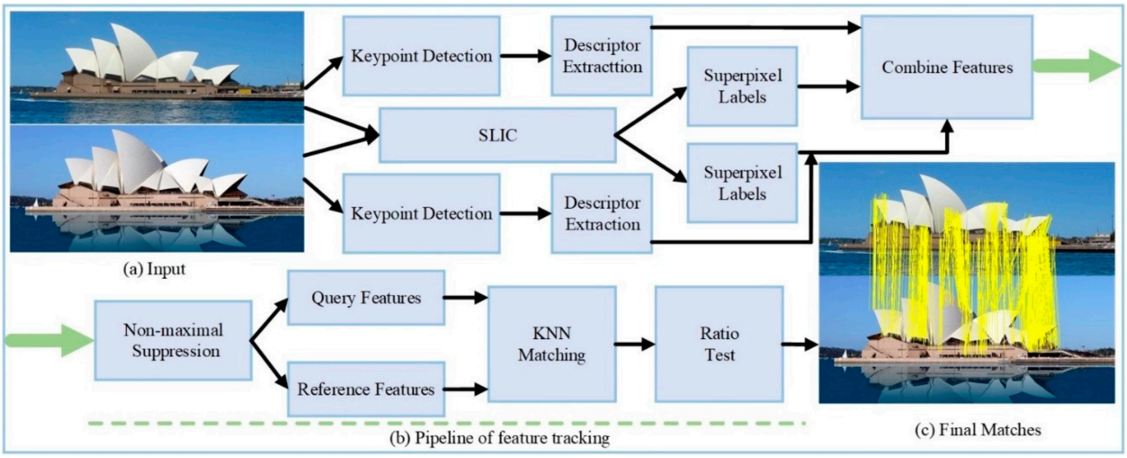


Figure 2. Flowchart of the superpixel-based feature tracking (SPFT) method.

4.1. Joint Keypoint Detector

To accelerate the speed of feature tracking, we propose a Joint Keypoint detector (JKD) that is based on FAST detector, as described in [39]. The JKD consists of two major stages: learning keypoint and superpixel-based keypoint location—each of which, in turn, takes several steps. In the stages of the learning keypoint, the input image is first convoluted. The output of convolution, known as the integral image, is then used as the basis of the scale-space analysis. The responses obtained from the scale-space analysis are utilized to detect the keypoints, $kp_i(x, y)$. In the stage of superpixel-based keypoint location, the SLIC is used to segment the input image to several labels, and then those labels have their center position, $cp_i(x, y)$. Finally, combine the $kp_i(x, y)$ and $cp_i(x, y)$, we can get the final keypoints, $k_i(x, y)$, via non-maximal suppression. The pipeline of the JKD keypoint detector is shown in Figure 3.

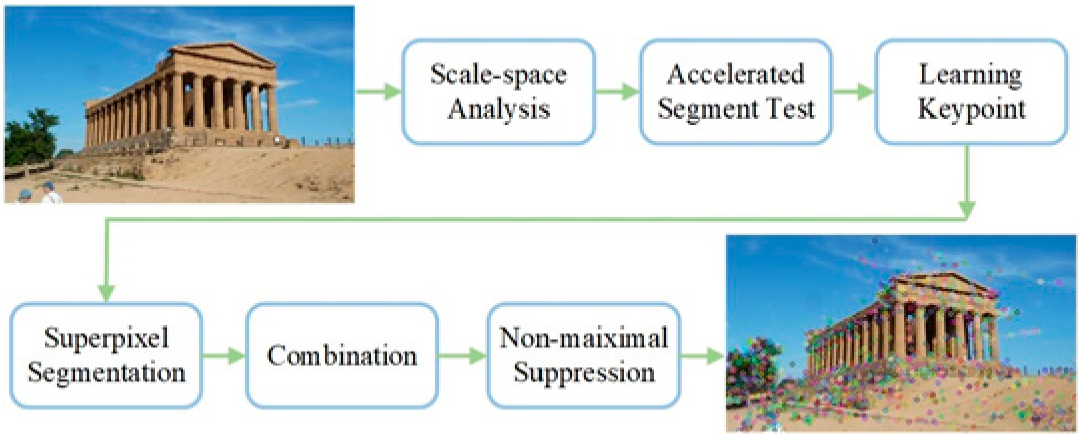


Figure 3. Joint keypoint detection.

Let $O(x, y)$ represents a candidate keypoint, and $N_O(x, y)$ represents the 7×7 neighbors of $O(x, y)$. Compute the DOG image of $R_O(x, y)$ to get $DOG_O(x, y)$ by Equation (1)

$$DOG_O(x, y) = G(x, y, k\sigma) - G(x, y, \sigma) \tag{1}$$

where k is a constant, $G(x, y, \sigma) = \frac{1}{2\pi\sigma} exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$ represents Gaussian density function with variance σ. Changing the value of σ, a set of DOG image is obtained as $DOG_{set}(x, y) = \{dog_1, \cdots, dog_5\}$ where 5 DOG images is constructed only for saving computation time.

For each location on the given DOG image I, the pixel at that position relative to O can have one of three states:

$$S_{O\rightarrow N} = \begin{cases} d, I_{o\rightarrow n} \leq I_o - t \\ s, I_o - t < I_{o\rightarrow n} < I_o + t \\ b, I_{o\rightarrow n} > I_o + t \end{cases} \tag{2}$$

where, $S_{O\rightarrow N}$ is a correlation between pixel o and n. d denotes darker, s denotes similar, and b denotes brighter. t is a threshold with a tiny value.

For all $O \in N_o$, the N_o can be divided into three subsets N_d, N_s, and N_b by computing $S_{O\rightarrow N}$. Use ID3 [67] algorithm to choose the first pixel n to compare with the candidate keypoint $O(x, y)$, and decide whether $O(x, y)$ is keypoint or not according to the entropy $H(O)$ of K_O.

$$\mathrm{H}(O) = \ (c + \bar{c}) \log_2(c + \bar{c}) - c \log_2 c - \bar{c} \log_2 \bar{c} \tag{3}$$

where c $= \left|\left\{p\middle|K_p \ is \ true\right\}\right|$ represents the number of keypoints and $\bar{c} = \left|\left\{p\middle|K_p \ is \ true\right\}\right|$ represents the number of non-keypoints.

If the selected n belongs to O_d and produce the max value of $H(O)$, then O_d can be further divided into the following five categories: O_{dd}, $O_{\overline{dd}}$, O_{ds}, O_{db}, $O_{\overline{db}}$. For O_s, divide it into O_{sd}, $O_{\overline{sd}}$, O_{ss}, O_{sb}, $O_{\overline{sb}}$. The process is applied recursively on all five subsets until $H(O)$ equals to zero. The candidate keypoint can be detected according to the value of K_O.

$$\mathrm{O}(x,\ y) = \begin{cases} true, K_O = 1 \\ false, K_O = 0 \end{cases} \tag{4}$$

where O is a keypoint if K_o is one. Repeat above process until all input images processed over, then a set of FAST keypoints can be obtained as follows:

$$K_{fast} = \{k_i | i = 1, \cdots, n\} \tag{5}$$

However, the keyoints detected by FAST are often distributed not average, then the resulting point-cloud models are discontinuous.

To avoid the in-averaging distributed of FAST keypoints, we use superpixel segmentation approach as a post-process step to find many small regions. Thus, the centers of the regions are selected as the candidate keypoints. For a given image in CIELAB color space, the candidate keypoints, K_{slic}, could be obtained by SLIC algorithm as described in Section 3.

$$K_{slic} = \left\{k^j_{slic}\middle|j = 1, \cdots, m\right\} \tag{6}$$

Once, the K_{fast} and K_{slic} are computed, the combined keypoints can be obtained as follows:

$$K_{find} = \left\{k^j_{slic} \cup k^i_{fast}\middle|j \in [1, m] \Lambda i \in [1, n]\right\} \tag{7}$$

To choose high-quality keypoints that have maximal responses, we use non-maximal suppression (NMS) [39] to eliminate the unstable keypoints that have minimal responses. The NMS is defined as

$$\mathrm{V} = max\left(\sum_{x \epsilon S_s} |I_{o\rightarrow x} - I_0| - t, \quad \sum_{x \epsilon S_d} |I_0 - I_{0\rightarrow x}| - t,\right) \tag{8}$$

As a result, by suppression the low-quality keypoints, the final keypoints that locate by the JKD is

$$K_{final} = \left\{ k_j \middle| \, j \in [1, \ m+n] \right\} \tag{9}$$

It should be note that the number of keypoints by JKD is vary, which depends on the value of σ in Equation (1). Thus, we can change σ to obtain more keypoints for special applications such as dense simultaneous localization and mapping (SLAM) [68] and face recognition [69,70].

4.2. Joint Descriptor Computing

The robustness of descriptor is very important to achieve robust feature tracking, which has been analyzed deeply in [43]. According to the last recent evaluation work made by Zhu et al. [71], the SURF feature has desirable performance on aspect of matching speed and precision. However, the SURF feature easily suffers from affine transform, this may break the compactness of point-cloud model when it is used in 3D reconstruction system. To improve the quality of 3D reconstruction system, we propose a joint computing procedure that include SURF and binary test [36], the former is use to describe the keypoints located in the texture areas, then the latter is used in the textureless areas. For convenience, we called the proposed feature descriptor as joint feature descriptor (JFD), the pipeline for computing a JFD feature descriptor is depicted in Figure 4, in which it is run on GPU device for accelerating. In addition to the matching precision and fast speed, the proposed JFD feature is also robust to various perturbations such as noise, illumination or contrast change.

For the k_j located in the texture areas, we first use SURF feature to compute a vector of 64 dimensional which is an normalized gradient statistics extracted from a spatial grid R divided into 4×4 regions. These subregions are referred to as $\mathrm{R} = \left\{ R_{i,j} \middle| 1 \le i, j \le 4 \right\}$. According to [20], the weighted gradient at point (u, v) is defined as,

$$\begin{pmatrix} d_x(u,v) \\ d_y(u,v) \end{pmatrix} = R_{-\theta_k} \begin{pmatrix} D_x^{L_k} \\ D_y^{L_k} \end{pmatrix} \varphi(x,y) \times G_1(u,v) \tag{10}$$

where $D_x^{L_k}$ and $D_y^{L_k}$ denote first order box filters, which are used to compute the gradient components.

To this end, the SURF uses first order statistical results on vertical and horizontal gradient responses to produce the good description that achieves the best performance between accuracy and efficiency, then the resulting statistical vector with respect to $R_{i,j}$ can be calculated by the following formula,

$$\mu_k(i,j) = \begin{pmatrix} \sum\limits_{u,v}^{R_{i,j}} d_x(u,v) \\ \sum\limits_{u,v}^{R_{i,j}} d_y(u,v) \\ \sum\limits_{u,v}^{R_{i,j}} \left| d_x(u,v) \right| \\ \sum\limits_{u,v}^{R_{i,j}} \left\lceil d_y(u,v) \right\rceil \end{pmatrix}, i, j \in [1, 4] \tag{11}$$

The SURF descriptor of k_i can be directly computed by concatenating the $\mu_k(i, j)$, which is defined as

$$\mu_k = vstack(\mu_k(i,j)) \tag{12}$$

where $vstack(\cdot)$ is function that represents stacking the matrix in vertical direction.

To improve the invariance to linear transform, the SURF descriptor should be normalized to a unit vector by L2 normal, the enhanced SURF descriptor can be calculated by the following formula

$$\mathrm{SURF}(k_i) = \mu_k / \|\mu_k\|_2 \tag{13}$$

However, for the keypoints distributed in textureless regions, we use binary test to produce robust descriptors in the neighbor regions that labeled by the superpixel-based segmentation. The binary test τ in [36] is defined as

$$\tau(L, x, y) = \begin{cases} 0, p(x) \geq p(y) \\ 1, p(x) < p(y) \end{cases} \tag{14}$$

where $p(x)$ represents the intensity of p at a point (x, y). Thus, the resulting feature vector is defined as

$$v_n(k_i) = v_n(p(x, y)) = \sum_{1 \leq i \leq n} 2^{i-1}(\tau(L_i, x_i, y_i)) \tag{15}$$

Note that n is set to 32 for saving computation time in the whole experiment, thus the resulting feature vector has 32 binary elements.

To this end, the JKD descriptor can be obtained by concatenating the SURF(k_i) and $v_n(k_i)$, then resulting a 96 dimensional of feature descriptor.

$$\mathrm{JKD}(k_i) = concat(\mathrm{SURF}(k_i), v_n(k_i)) \tag{16}$$

Owing to the JKD is hybrid type, namely it not only includes float type elements, but also contains binary type ones, thus, we need urgently a novel matching approach to match them.

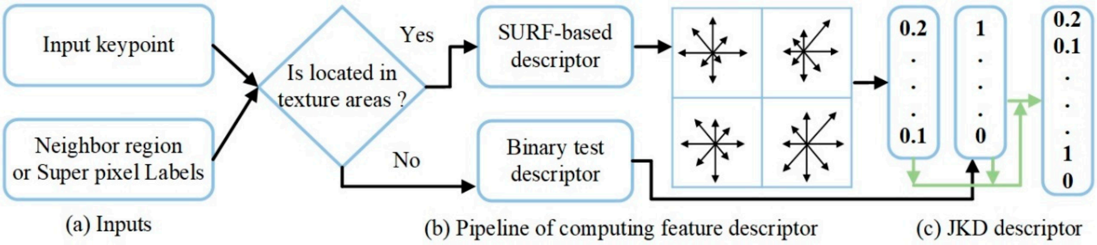

Input keypoint
Neighbor region or Super pixel Labels
Is located in texture areas ?
Yes
No
SURF-based descriptor
Binary test descriptor
0.2
0.1
1
0
0.2
0.1
1
0
(a) Inputs
(b) Pipeline of computing feature descriptor
(c) JKD descriptor

Figure 4. Flowchart of joint keypoint computing.

4.3. Fast Descriptor Matching

Feature matching aims to measure the similarity between the two feature descriptors. The float-type descriptors, such as SIFT, SURF et al. usually use Euclidean distance (L2 distance) to measure the similarity of two feature descriptors [11]. For binary descriptors such as BRISK [37] and LGHD [72], the Hamming distance is used [43]. Because our descriptor is hybrid type that not only includes float-type elements, but also contains binary-type ones. Thus, we use two metrics to measure the similarity of the proposed feature descriptors, namely Hamming distance and Euclidean distance as shown in Figure 5.

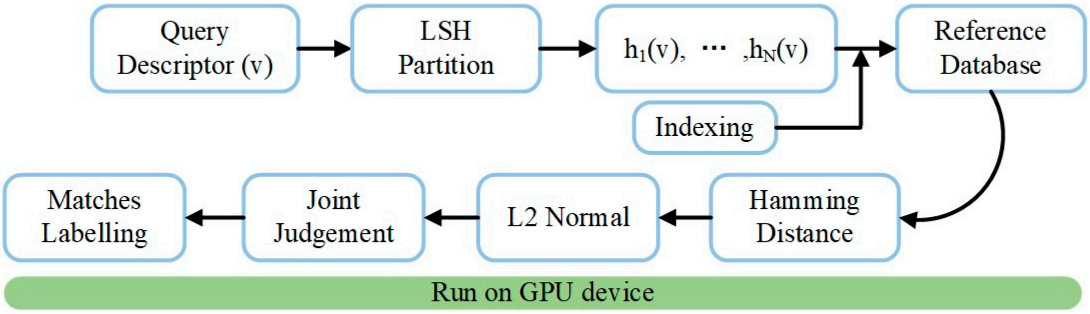


Figure 5. Flowchart of descriptor matching.

The former is utilized to measure similarity of superpixel-based feature descriptors, then the latter is exploited to handle float-type feature descriptors. For the given two binary-type descriptors,

$DB_r = \left\{d_r^1, \cdots, d_r^n\right\}$, $D_{Bq} = \left\{d_q^1, \cdots, d_q^n\right\}$, then the similarity between DB_r and DB_q can be calculated by the simple bitwise operation.

$$\mathrm{MS}_{q,r} = DB_r \; xor \; DB_q \tag{17}$$

where *xor* denotes XOR operation which returns the number of different elements between DB_r and DB_q.

However, for float-type feature descriptor, $DF_q = \{q_1, \cdots, q_m\}$ and $DF_r = \{r_1, \cdots, r_m\}$, we use the Euclidean distance (L2 normal) to estimate the similarity of them, the matching confidence can be calculated as

$$C_{qr} = \|\mathrm{p}(q_i) - \mathrm{p}(r_i)\|, \; i \in [1, m] \tag{18}$$

where $\mathrm{p}(q_i)$ and $\mathrm{p}(r_i)$ denote the descriptor for keypoint q_i and keypoint r_i respectively.

Once, the metrics are defined, we can simply loop the above procedure until the feature descriptors in the feature database is processed over, then every feature descriptor in query feature database has two potentially corresponding candidates. Let $\mathrm{p}(r_i)$ and $\mathrm{p}\left(r_j\right)$ denote the candidates with respect to the query descriptor $\mathrm{p}(q_i)$, then we can judge whether the matching is successful by the following formula.

$$C_f = \frac{\|\mathrm{p}(q_i) - \mathrm{p}(r_i)\|}{\|\mathrm{p}(q_i) - \mathrm{p}\left(r_j\right)\|} \tag{19}$$

If c < 0.7, the $\left\langle q_i, r_j \right\rangle$ is a correct match. Base on the hybrid matching approach, we can use the Brute-Force-Match (BFM) [73] to find a candidate for each query keypoint.

However, BFM-based KNN approach is a greedy algorithm and has an expensive computational cost. If the matching method is utilized in large-scale 3D reconstruction, then the process of recovering 3D model is very slow. Thus, we must improve the computation efficiency of BFM-based KNN to accelerate the feature tracking method. After a deep investigation in descriptor matching methods [74,75], we found that local sensitive hash (LSH) [51,76] is an efficient approach to achieve descriptor matching. Thus, the LSH is utilized to match feature descriptors. The core of LSH algorithm is an approximate approach to compute k-nearest neighbors, which use N hash functions $h_1(\cdot), \cdots, h_N(\cdot)$ to transform the D-dimensional space R^D into a lattice space L^D, and the original each data is distributed into one lattice cell:

$$H(v) = \{h_1(v), \cdots, (v)\} \tag{20}$$

where v denotes a vector of query descriptor.

To this end, the LSH-based KNN can use the L2 distance to measure the similarity between the query descriptor and the reference descriptor.

Algorithm 1 Superpixel-based feature tracking scheme

```
Input: image sequences, $I = \{I_1, I_2, \cdots, I_N\}$.
Output: a set of matching pairs, $S = \left\{\left\langle k_{ij}, k_{hc} \right\rangle \middle| i, h \in [1, N]\right\}$.

Step1: Compute keypoints for each image in $\{I_1, I_2, \cdots, I_N\}$, then resulting in a set of keypoints,
       $\{k_1, k_2, \cdots, k_m\}$.
Step2: Compute feature descriptor for each located keypoint, if they are located in texture areas, then use
       Equations (11) and (12) to obtain robust description, otherwise, use binary test that defined in
       Equation (15) to describe the keypoints.
Step3: Construct hash tables via Equation (20), the large set of JKD descriptors is distributed into many
       lattice cells independently.
Step4: For JKD$(k_i)$ and JKD$\left(k_j\right)$ the similarity can be measured by Equations (17) and 19. If those formulas
       are true, JKD$(k_i)$ and JKD$\left(k_j\right)$ are considered matching.
Step5: Repeat Step4 for any two keypoints in $\{k_1, k_2, \cdots, k_m\}$ the resulting matching pairs is
       $S = \left\{\left\langle k_{ij}, k_{hc} \right\rangle \middle| i, h \in [1, N]\right\}$.
```

5. Experimental Results

The proposed SPFT is developed in C++, NVIDIA CUDA SDK 10.0 and OpenCV SDK 4.0, on a PC with Intel i7 CPU processor 3.40 GHz and 32.0GB memory. We have evaluated the SPFT method on several challenging dataset, and have compared it with the state-of-the-art methods, including HPM [77], ROML [78], MODS [79], ENFT [35] and SuperPoint [80]. It should be noted that SuperPoint is deep learning-based approach to feature detection and descriptor computing, and is published on the European Conference on Computer Vision in 2018.

5.1. Evaluation of Colosseum Dataset

We have evaluated the performance of the SPFT on the Colosseum dataset which is constructed by the authors of this paper. Samples of the Oxford benchmark are shown in Figure 6 where the lighting of every images is different to each other, and they also have many repeated features and structures. In the whole process of experiment, we use a standard evaluation metric to measure the performance for each method. The evaluation metric is defined as:

$$Precision = \frac{\#correct\ matches}{\#tentative\ matches} \tag{21}$$

where *#correct matches* stands for the number of correct matches, *#tentative matches* represents the number of raw matches, namely does not have any post-process steps such as RANSAC, cross-check and ratio-test.

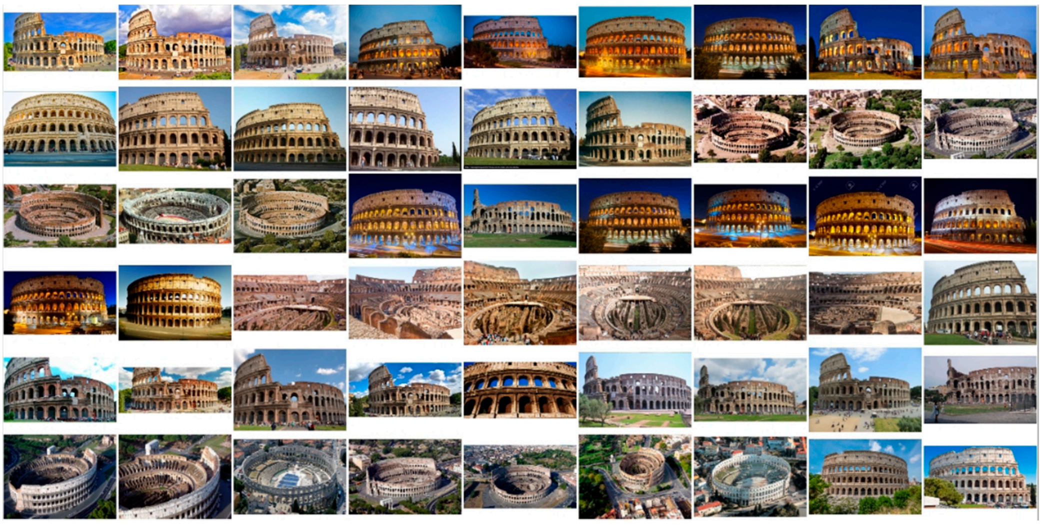


Figure 6. Samples from Colosseum dataset.

5.1.1. Matching Precession

Figure 7 presents visualized results for each method, the green lines denotes correct matches. The HPM obtained the minimal number of feature correspondences. The number of feature correspondences from ROML is more than that of HPM. The number of feature correspondences of SuperPoint is the second place. The SPFT has the maximal number of feature correspondences. According to the common sense in the field of 3D reconstruction, the more the number of feature correspondences, the denser the point-cloud model from 3D reconstruction system. Thus, the SPFT can significantly increase the density of the reconstructed point-cloud model.

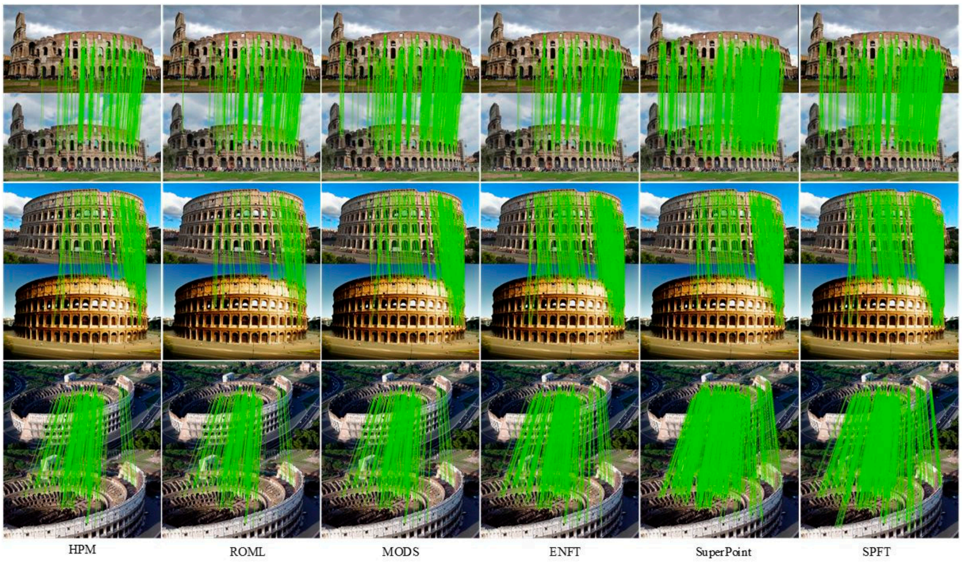


Figure 7. Matching results for the Colosseum dataset.

Moreover, we have tallied up the matching precision for each feature tracking method, the statistical results are depicted in Figure 8. Among those methods [77–80], the proposed SPFT has the highest matching precision, namely the matching precision is 85.6%; The SuperPoint is in the second place; The ENFT is in the third place; and the matching precision of HPM is the lowest. The matching performance of MODS is better that that of HMP and ROML. According to this experiment, we have the following valuable findings: (1) ENFT have robustness to rotation change due to the usage of SIFT feature; (2) The viewpoint change has a significantly impact on the matching precision of feature tracking method; (3) The scale-space has heavily impact on the matching precision because the number of keypoints in multiple scale spaces is more than that of the keypoint detector in single scale space; (4) Superpixel-based segmentation can be used to find potentially keypoints that in the textureless regions. As a result, the matching precision of the SPFT is largely attributed to the usage of multiple scale spaces and superpixel segmentation; (5) Deep learning-based method, such as SuperPoint, can improve the matching precession in the single scale space of the image.

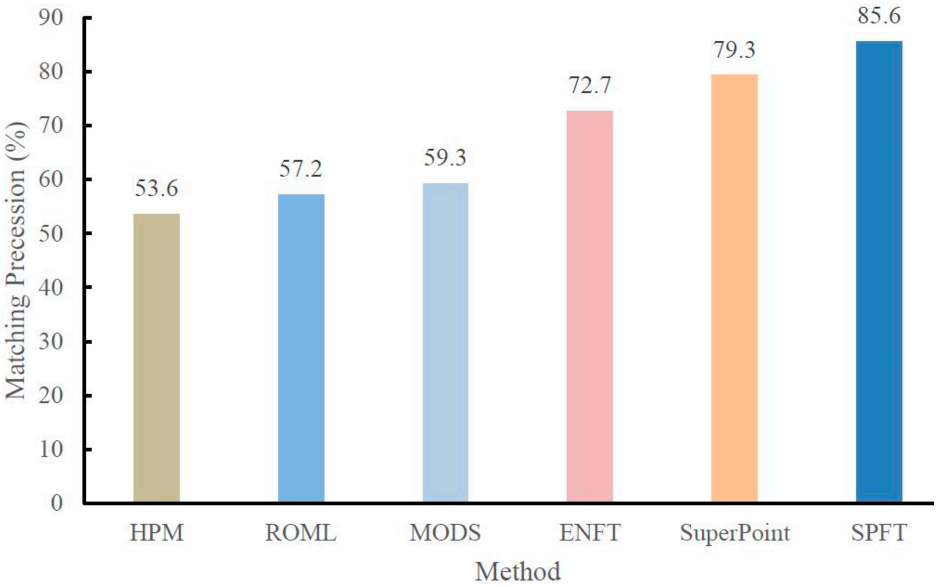


Figure 8. Averaging matching precessions for the evaluated methods, where the SPFT has the best performance, the SuperPoint is in the second place.

5.1.2. Computation Time

Computational cost is one of the most evaluation metrics for feature tracking methods, thus, we have collected the computation times for each compared feature tracking method according to the assessment that conducted on Colosseum dataset. The statistical results of computation times for each method are depicted in Figure 9 where the computation time is the sum of that spend on the whole pipeline including keypoint detection, descriptor computing, and feature matching. We can clearly see that the SPFT has the fastest speed, the averaging computation time is 6.7 s. The ENFT is in the second place, its averaging computation time is 9.2 s. Among those compared methods, the ROML has the lowest speed, which requires 21.3 s averagely for image pairs matching. After deeply investigation for ROML, we found that the main reason attributed to the highest computational cost of ROML is implementation in MATLAB routines. We hold that the ROML may be significantly accelerated when implementation in C++ programming language. As shown in Figure 9, the speed of the proposed SPFT is about 3 times faster than that of ROML, and is about 2–3 times faster than that of HPM and MODS. According to the statistical results of matching precision and computation time, we can conclude that the SPFT feature has the best performance in both accuracy and efficiency. In addition to ROML, the SuperPoint has the lowest speed, the averaging time is 18.2 s according to the experiment.

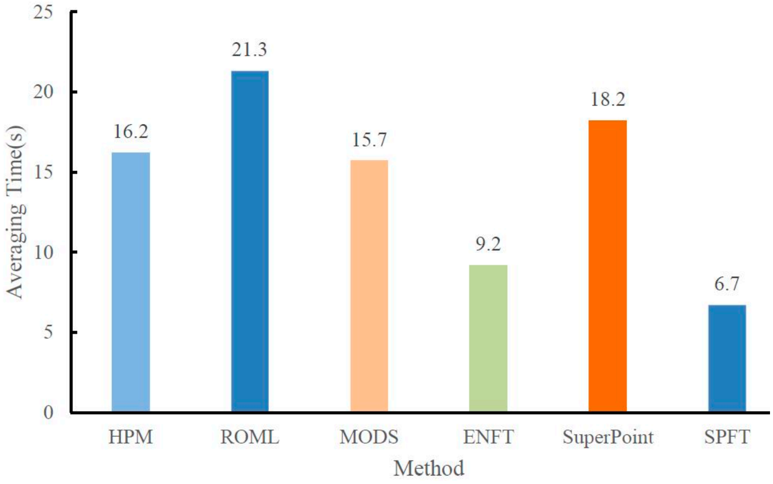


Figure 9. Averaging times for the evaluated methods.

5.2. Evaluation on HFUT Dataset

In the field of 3D reconstruction, if a feature tracking method is integrated into a 3D reconstruction system, which can produce high-quality point-cloud model, we consider the feature tracking method as a good approach to 3D reconstruction. Based on this judgement criteria, we create a new dataset captured by Canon EOS 550D camera, we named the new dataset as HFUT dataset for short. Figure 10 presents samples of the HFUT dataset, which contains 120 images and have many repeated features and repeated structures on the surface of each image. In addition to repeated features, the light for each image is very weak, which pose a new challenge for feature tracking method. In this experiment, we integrated the SPFT feature tracking method into ISFM system [2] to recover the point-cloud model, the results are shown in Figure 11. We can see that the reconstructed point-cloud model has highly geometric consistency with respect to the real scenario. Moreover, we found that the resulting point-cloud model is very dense, which is attributed to the usage of the SPFT feature tracking method. According to our record, the ISFM system with SPFT can recover a high-quality point-cloud model having 338,391 vertices for the HFUT dataset in 5.5 min. As a result, we consider the SPFT has an excellent performance in practice.

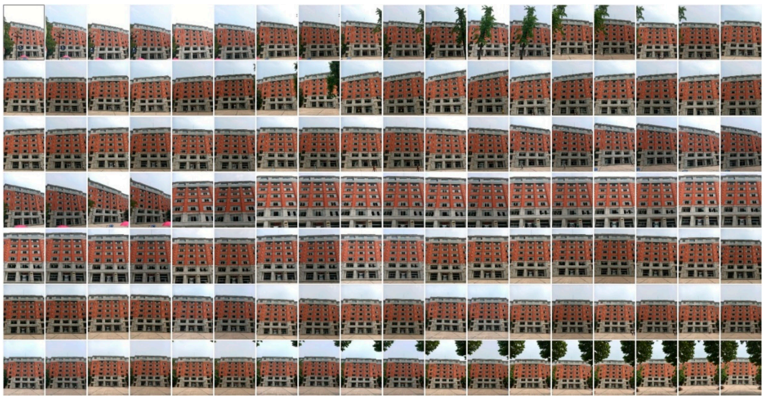
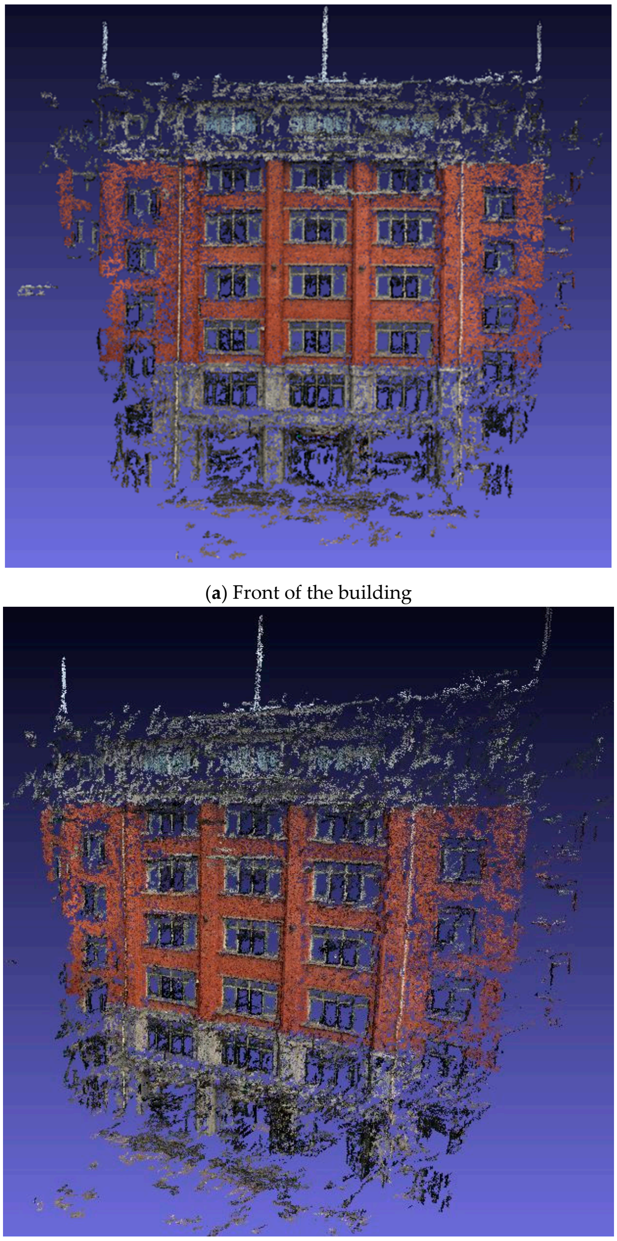


Figure 10. Samples from HFUT dataset.



(**a**) Front of the building

(**b**) Side of the building

Figure 11. The point-cloud model for HFUT dataset.

5.3. Evaluation of Forensic Dataset

To assess the scalability of the SPFT, we have evaluated it on the Forensic dataset provided by the PIX4D company. The samples of the UAV dataset are provided in Figure 12, which is captured by unmanned aerial vehicle and has large-scale resolution and many repeated features on the surface of each image. In summary, the Forensic dataset is very challenge for feature tracking method and structure from motion.

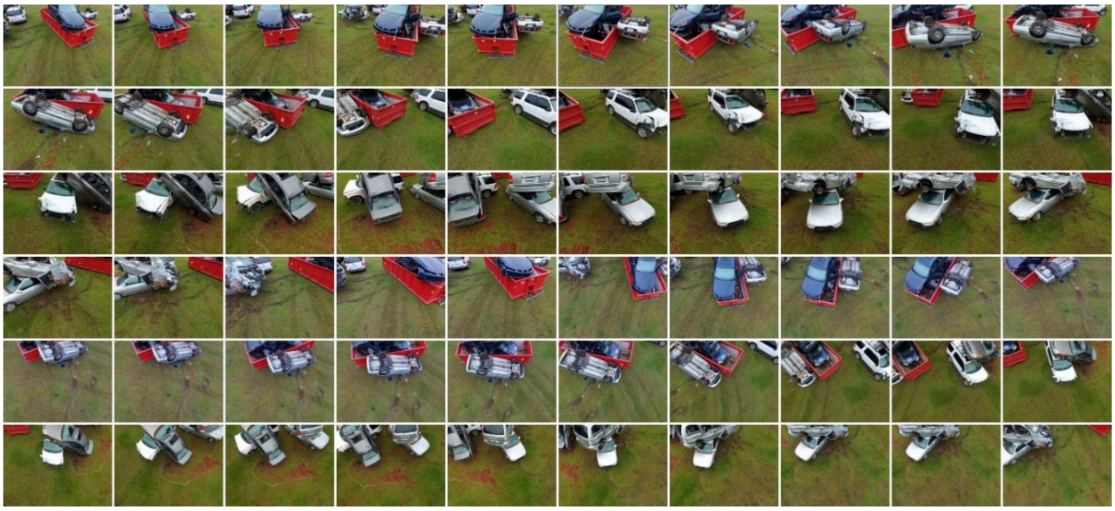


Figure 12. Samples from the Forensic dataset. Note that many repeating features are appeared on the surface of each image.

Figure 13 presents the visual correspondences of each feature tracking method for the Forensic dataset, where the SPFT has obtained the maximum number of feature matches, and has the fastest speed among the compared feature tracking methods. The HPM has the minimum number of visual correspondences, and has the lowest speed. According to our statistic, the HPM has an average of 55 feature correspondences on the Forensic dataset. The number of visual correspondences of the MODS in the second place, and it has lower speed than that of the HPM approach because of views synthesis. Although the ENFT has number of visual correspondences less than that of MODS, which has a cheap computational cost. After a deep analysis for the ENFT method, we found that the ENFT heavily dependents on the segmentation for input video or image sequences to decrease the computational burden. But, the segmented-based approach easily handicaps the quality of the point-cloud model that is constructed by the SFM system. The SuperPoint has more feature correspondences than that of ENFT, but less than that of ours. However, the proposed SPFT method not only has the cheapest computational cost but also has the highest matching precision among these compared feature tracking methods. According to our statistical results in experiment, the SPFT method has an average of 1876 correct feature matches.

In addition to making a comparison with the state-of-the-art method, we have integrated the SPFT into the ISFM system [2], and use the combinational system to estimate the point-cloud model for the Forensic dataset. Figure 14 provides the sparse point-cloud model for the Forensic dataset, which has 2,683,015 vertices and is reconstructed in 10.6 min. We can see that the constructed point-cloud model has good geometric consistency with corresponding to the real scenarios. As a result, we can draw a conclusion that the SPFT has the best performance in both accuracy and efficiency.

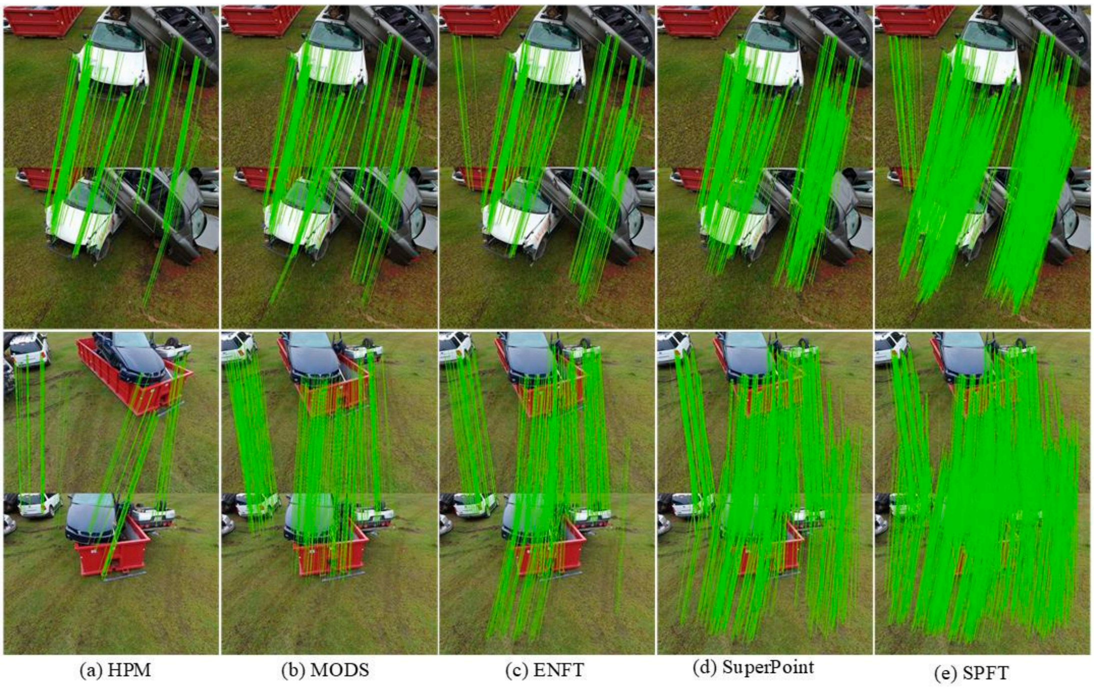


Figure 13. Visual correspondences for each method on Forensic dataset.

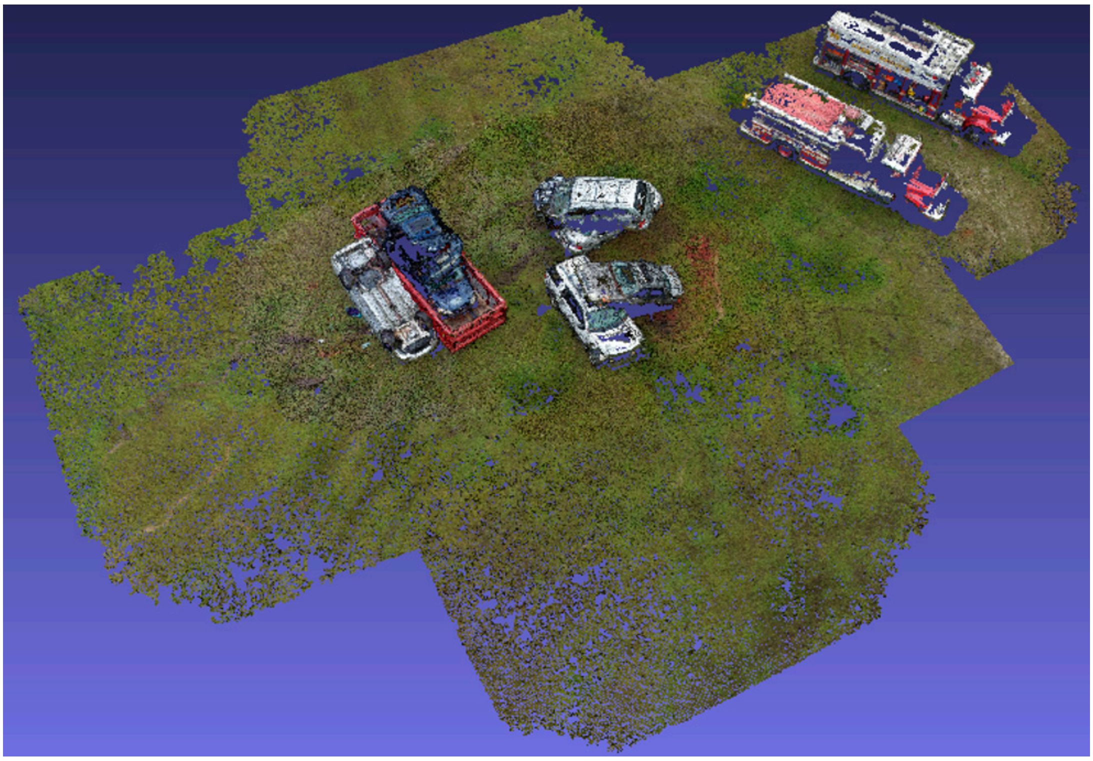


Figure 14. The sparse point-cloud model for the forensic dataset, and constructed by the ISFM system with the RTFT feature tracking method.

6. Conclusions

In this paper, we proposed an accurate, fast and robust feature tracking method for SFM-based 3D reconstruction, which is based on the superpixel segmentation to increase the number of potentially keypoint and improve the descriptor's quality. In the stage of feature detection, a multiple scale-space analysis and the superpixel-based segmentation technique is used to candidate keypoints, then using non-maximal suppression technique to remove some unstable keypoints from the initial keypoint collection. In the stage of descriptor computing, we use the segment-based binary test to produce a robust descriptor for each keypoints. In the stage of feature matching, the GPU-accelerated KNN method with ratio-test is used to measure the similarity of two descriptors for saving computation time. Finally, we have evaluated the SPFT on the several challenging datasets, and compared it with the state-of-the-arts feature tracking methods. Moreover, the SPFT is integrated into an SFM-based 3D reconstruction system, then resulting high-quality point-cloud models on the challenging datasets. I hold that the SPFT likes a unified framework of feature tracking, in which with different superpixel methods or KNN-like methods, the SPFT may produce a novel feature tracking method. Thus, the SPFT has good ex extendibility.

Besides of promising feature tracking method, we have other valuable findings according to experiments: (1) the number of located keypoints largely depends on multiple scale spaces; (2) the context information is very important to construct a robust descriptor for keypoint; (3) the usage of shared memory in GPU device is also important to accelerate the feature matching speed. In summary, we proposed a promising feature tracking method for SFM-based 3D reconstruction, the quality of point-cloud model is significantly improved when it is used. In the future, we will try to propose a novel feature tracking method based on the proposed SPFT framework for simultaneous localization and mapping.

Author Contributions: Conceptualization, M.C. and L.Z.; methodology, M.C.; software, M.C.; validation, M.C., W.J. and L.Z.; formal analysis, Z.L.; investigation, M.C.; resources, W.J.; data curation, M.C.; writing—original draft preparation, M.C.; writing—review and editing, Z.L., W.J. and X.L.; visualization, M.C.; supervision, X.L.; project administration, L.Z.; funding acquisition, X.L., M.C., W.J. and Z.L.

Funding: This research was funded by [National Science Foundation of China] grant number [61802103, 61877016, and 61673157], and [Postdoctoral Science Foundation] grant number [2018M632522], and [Fundamental Research Funds for the Central Universities] grant number [JZ2018HGBH0280, and PA2018GDQT0014], and [Natural Science Foundation of Shandong Province] grant number [ZR2017QF015], and [Key Research and Development Program in Anhui Province] grant number [1804a09020036].

Acknowledgments: The authors gratefully acknowledge the support of the National Natural Science Foundation (Grant No.: 61802103, 61877016, 61602146 and 61673157), Postdoctoral Science Foundation (Grant No.: 2018M632522), Fundamental Research Funds for the Central Universities (Grant No.: JZ2018HGBH0280 and PA2018GDQT0014), and Natural Science Foundation of Shandong Province (Grant No.: ZR2017QF015), and Key Research and Development Program in Anhui Province (Grant No.: 1804a09020036).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

References

1. Lv, Z.; Li, X.; Li, W. Virtual reality geographical interactive scene semantics research for immersive geography learning. *Neurocomputing* **2017**, *254*, 71–78. [CrossRef]
2. Cao, M.W.; Jia, W.; Zhao, Y.; Li, S.J.; Liu, X.P. Fast and robust absolute camera pose estimation with known focal length. *Neural Comput. Appl.* **2017**, *29*, 1383–1398. [CrossRef]
3. Kong, C.; Lucey, S. Prior-Less Compressible Structure from Motion. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4123–4131.
4. Cao, M.W.; Li, S.J.; Jia, W.; Li, S.L.; Liu, X.P. Robust bundle adjustment for large-scale structure from motion. *Multimed. Tools Appl.* **2017**, *76*, 21843–21867. [CrossRef]
5. Lu, H.M.; Li, Y.J.; Mu, S.L.; Wang, D.; Kim, H.; Serikawa, S. Motor Anomaly Detection for Unmanned Aerial Vehicles Using Reinforcement Learning. *IEEE Internet Things J.* **2017**, *5*, 2315–2322. [CrossRef]
6. Lu, H.M.; Li, B.; Zhu, J.W.; Li, Y.J.; Li, Y.; Xu, X.; He, L.; Li, X.; Li, J.R.; Serikawa, S. Wound intensity correction and segmentation with convolutional neural networks. *Concurr. Comput. Pract. Exp.* **2017**, *29*, e3927. [CrossRef]
7. Zhang, X.L.; Han, Y.; Hao, D.S.; Lv, Z.H. ARGIS-based Outdoor Underground Pipeline Information System. *J. Vis. Commun. Image Represent.* **2016**, *40*, 779–790. [CrossRef]

8. Serikawa, S.; Lu, H. Underwater image dehazing using joint trilateral filter. *Comput. Electr. Eng.* **2014**, *40*, 41–50. [CrossRef]
9. Ozyesil, O.; Voroninski, V.; Basri, R.; Singer, A. A Survey of Structure from Motion. *Acta Numer.* **2017**, *26*, 305–364. [CrossRef]
10. Snavely, N.; Seitz, S.M.; Szeliski, R. Photo tourism: Exploring photo collections in 3D. *ACM Trans. Graph. (TOG)* **2006**, *25*, 835–846. [CrossRef]
11. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
12. Wu, C. Towards linear-time incremental structure from motion. In Proceedings of the International Conference on 3D Vision-3DV 2013, Seattle, WA, USA, 29 June–1 July 2013.
13. Furukawa, Y.; Ponce, J. Accurate, Dense, and Robust Multi-View Stereopsis. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA, 17–22 June 2007.
14. Kazhdan, M.; Bolitho, M.; Hoppe, H. Poisson surface reconstruction. In Proceedings of the Fourth Eurographics Symposium on Geometry Processing, Cagliari, Sardinia, Italy, 26–28 June 2006.
15. Dong, Z.L.; Zhang, G.F.; Jia, J.Y.; Bao, H.J. Keyframe-based real-time camera tracking. In Proceedings of the IEEE 12th International Conference on Computer Vision, Kyoto, Japan, 29 September–2 October 2009.
16. Zhang, G.F.; Liu, H.M.; Dong, Z.L.; Jia, J.Y.; Wong, T.T.; Bao, H.J. ENFT: Efficient Non-Consecutive Feature Tracking for Robust Structure-from-Motion. *arXiv* **2015**, arXiv:1510.08012.
17. Ni, K.; Dellaert, F. HyperSfM. In Proceedings of the 2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), Zurich, Switzerland, 13–15 October 2012.
18. Schönberger, J.L.; Frahm, J.-M. Structure-from-motion revisited. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
19. Zach, C. *ETH-V3D Structure-and-Motion Software.© 2010–2011*; ETH Zurich: Zürich, Switzerland, 2010.
20. Bay, H.; Tuytelaars, T.; van Gool, L. Surf: Speeded up robust features. In Proceedings of the European Conference on Computer Vision, Graz, Austria, 7–13 May 2006; Springer: Berlin/Heidelberg, Germany, 2006; pp. 404–417.
21. Agarwal, S.; Furukawa, Y.; Snavely, N.; Simon, I.; Curless, B.; Seitz, S.M.; Szeliski, R. Building rome in a day. In Proceedings of the IEEE 12th International Conference on Computer Vision, Kyoto, Japan, 27 September–4 October 2009.
22. Zach, C.; Klopschitz, M.; Pollefeys, M. Disambiguating visual relations using loop constraints. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 1426–1433.
23. Fan, B.; Wu, F.; Hu, Z. Towards reliable matching of images containing repetitive patterns. *Pattern Recognit. Lett.* **2011**, *32*, 1851–1859. [CrossRef]
24. Roberts, R.; Sinha, S.N.; Szeliski, R.; Steedly, D. Structure from motion for scenes with large duplicate structures. In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, Colorado Springs, CO, USA, 20–25 June 2011; pp. 3137–3144.
25. Wilson, K.; Snavely, N. Network Principles for SfM: Disambiguating Repeated Structures with Local Context. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, NSW, Australia, 1–8 December 2013; pp. 513–520.
26. Ceylan, D.; Mitra, N.J.; Zheng, Y.; Pauly, M. Coupled structure-from-motion and 3D symmetry detection for urban facades. *ACM Trans. Graph.* **2014**, *33*, 57–76. [CrossRef]
27. Saputra, M.R.U.; Markham, A.; Trigoni, N. Visual SLAM and Structure from Motion in Dynamic Environments: A Survey. *ACM Comput. Surv.* **2018**, *51*, 37. [CrossRef]
28. Knapitsch, A.; Park, J.; Zhou, Q.Y.; Koltun, V. Tanks and Temples: Benchmarking Large-Scale Scene Reconstruction. *ACM Trans. Graph.* **2017**, *36*, 78. [CrossRef]
29. Alcantarilla, P.F.; Bartoli, A.; Davison, A.J. KAZE features. In Proceedings of the Computer Vision–ECCV 2012, Florence, Italy, 7–13 October 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 214–227.
30. Tombari, F.; Di Stefano, L. *Interest Points via Maximal Self-Dissimilarities*; Springer International Publishing: Cham, Switzerland, 2015.
31. Tomasi, C.; Kanade, T. Detection and tracking of point features. *Int. J. Comput. Vis.* **1991**, *20*, 110–121.

32. Cao, M.; Jia, W.; Lv, Z.; Li, Y.; Xie, W.; Zheng, L.; Liu, X. Fast and robust feature tracking for 3D reconstruction. *Opt. Laser Technol.* **2018**, *110*, 120–128. [CrossRef]
33. Sinha, S.N.; Frahm, J.M.; Pollefeys, M.; Genc, Y. GPU-based video feature tracking and matching. In Proceedings of the EDGE, Workshop on Edge Computing Using New Commodity Architectures, Chapel Hill, NC, USA, 23–24 May 2006.
34. Crandall, D.; Owens, A.; Snavely, N.; Huttenlocher, D. Discrete-continuous optimization for large-scale structure from motion. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Colorado Springs, CO, USA, 20–25 June 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 3001–3008.
35. Guofeng, Z.; Haomin, L.; Zilong, D.; Jiaya, J.; Tien-Tsin, W.; Hujun, B. Efficient Non-Consecutive Feature Tracking for Robust Structure-From-Motion. *IEEE Trans. Image Process.* **2016**, *25*, 5957–5970.
36. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. ORB: An efficient alternative to SIFT or SURF. In Proceedings of the 2011 IEEE International Conference on Computer Vision (ICCV), Barcelona, Spain, 6–13 November 2011; IEEE: Piscataway, NJ, USA, 2011.
37. Leutenegger, S.; Chli, M.; Siegwart, R.Y. BRISK: Binary robust invariant scalable keypoints. In Proceedings of the 2011 IEEE International Conference on Computer Vision (ICCV), Barcelona, Spain, 6–13 November 2011; IEEE: Piscataway, NJ, USA, 2011.
38. Forssén, P.-E.; Lowe, D.G. Shape descriptors for maximally stable extremal regions. In Proceedings of the IEEE 11th International Conference on Computer Vision (ICCV 2007), Rio de Janeiro, Brazil, 14–21 October 2007; IEEE: Piscataway, NJ, USA, 2007.
39. Rosten, E.; Drummond, T. Machine learning for high-speed corner detection. In Proceedings of the Computer Vision–ECCV 2006, Graz, Austria, 7–13 May 2006; Springer: Berlin, Germany, 2006; pp. 430–443.
40. Mair, E.; Hager, E.M.; Burschka, D.; Suppa, M.; Hirzinger, G. Adaptive and generic corner detection based on the accelerated segment test. In Proceedings of the Computer Vision–ECCV 2010, Crete, Greece, 5–11 September 2010; Springer: Berlin, Germany, 2010; pp. 183–196.
41. Agrawal, M.; Konolige, K.; Blas, M.R. CenSure: Center surround extremas for realtime feature detection and matching. In Proceedings of the Computer Vision–ECCV 2008, Marseille, France, 12–18 October 2008; Springer: Berlin, Germany, 2008; pp. 102–115.
42. Klein, G.; Murray, D. Parallel tracking and mapping for small AR workspaces. In Proceedings of the 6th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2007), Nara, Japan, 13–16 November 2007; IEEE: Piscataway, NJ, USA, 2007.
43. Yang, X.; Cheng, K.-T. LDB: An ultra-fast feature for scalable augmented reality on mobile devices. In Proceedings of the 2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Atlanta, GA, USA, 5–8 November 2012; IEEE: Piscataway, NJ, USA, 2012.
44. Yang, X.; Cheng, K.-T. Local difference binary for ultrafast and distinctive feature description. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 188–194. [CrossRef]
45. Levi, G.; Hassner, T. LATCH: Learned Arrangements of Three Patch Codes. *arXiv* **2015**, arXiv:1501.03719.
46. Trzcinski, T.; Christoudias, M.; Fua, P.; Lepetit, V. Boosting binary keypoint descriptors. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013.
47. Alahi, A.; Ortiz, R.; Vandergheynst, P. Freak: Fast retina keypoint. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012; IEEE: Piscataway, NJ, USA, 2012.
48. Wu, C. SiftGPU: A GPU Implementation of Scale Invariant Feature Transform. 2011. Available online: http://cs.unc.edu/~{}ccwu/siftgpu (accessed on 10 November 2018).
49. Graves, A. GPU-accelerated feature tracking. In Proceedings of the 2016 IEEE National Aerospace and Electronics Conference (NAECON) and Ohio Innovation Summit (OIS), Dayton, OH, USA, 25–29 July 2016.
50. Cao, M.; Jia, W.; Li, S.; Li, Y.; Zheng, L.; Liu, X. GPU-accelerated feature tracking for 3D reconstruction. *Opt. Laser Technol.* **2018**, *110*, 165–175. [CrossRef]
51. Xu, T.; Sun, K.; Tao, W. *GPU Accelerated Image Matching with Cascade Hashing*; Springer: Singapore, 2017.
52. Micusik, B.; Wildenauer, H. Structure from Motion with Line Segments Under Relaxed Endpoint Constraints. *Int. J. Comput. Vis.* **2017**, *124*, 65–79. [CrossRef]
53. Sweeney, C.; Fragoso, V.; Hollerer, T.; Turk, M. Large Scale SfM with the Distributed Camera Model. In Proceedings of the 2016 Fourth International Conference on 3D Vision (3DV), Stanford, CA, USA, 25–28 October 2016.

54. Wilson, K.; Snavely, N. Robust global translations with 1dsfm. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; Springer: Berlin, Germany, 2014.
55. Moulon, P.; Monasse, P.; Marlet, R. Global fusion of relative motions for robust, accurate and scalable structure from motion. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, NSW, Australia, 1–8 December 2013.
56. Sweeney, C.; Sattler, T.; Hollerer, T.; Turk, M.; Pollefeys, M. Optimizing the Viewing Graph for Structure-from-Motion. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015.
57. Goldstein, T.; Hand, P.; Lee, C.; Voroninski, V.; Soatto, S. ShapeFit and ShapeKick for Robust, Scalable Structure from Motion. In Proceedings of the Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; pp. 289–304.
58. Cohen, A.; Schonberger, J.; Speciale, P.; Sattler, T.; Frahm, J.; Pollefeys, M. Indoor-Outdoor 3D Reconstruction Alignment. In Proceedings of the Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; pp. 285–300.
59. Albl, C.; Sugimoto, A.; Pajdla, T. Degeneracies in Rolling Shutter SfM. In Proceedings of the Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; pp. 36–51.
60. Xiao, J.; Owens, A.; Torralba, A. SUN3D: A database of big spaces reconstructed using sfm and object labels. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, NSW, Australia, 1–8 December 2013.
61. Cui, H.; Gao, X.; Shen, S.; Hu, Z. HSfM: Hybrid Structure-from-Motion. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
62. Ren, X.; Malik, J. Learning a Classification Model for Segmentation. In Proceedings of the Ninth IEEE International Conference on Computer Vision, Nice, France, 13–16 October2003; IEEE Computer Society: Washington, DC, USA, 2003; p. 10.
63. Achanta, R.; Shaji, A.; Smith, K.; Lucchi, A.; Fua, P.; Susstrunk, S. SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 2274–2282. [CrossRef] [PubMed]
64. Van den Bergh, M.; Boix, X.; Roig, G.; De Capitani, B.; Van Gool, L. SEEDS: Superpixels Extracted Via Energy-Driven Sampling. *Int. J. Comput. Vis.* **2015**, *111*, 298–314. [CrossRef]
65. Moore, A.P.; Prince, S.J.D.; Warrell, J.; Mohammed, U.; Jones, G. Superpixel lattices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008.
66. Ban, Z.; Liu, J.; Fouriaux, J. GMMSP on GPU. *J. Real-Time Image Process.* **2018**, *13*, 1–13. [CrossRef]
67. Quinlan, J.R. Induction of decision trees. *Mach. Learn.* **1986**, *1*, 81–106. [CrossRef]
68. Salas-Moreno, R.F.; Glocker, B.; Kelly, P.H.J.; Davison, A.J. Dense planar SLAM. In Proceedings of the IEEE International Symposium on Mixed and Augmented Reality, Munich, Germany, 10–12 September 2014.
69. Ge, S.; Li, J.; Ye, Q.; Luo, Z. Detecting Masked Faces in the Wild with LLE-CNNs. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
70. Ge, S.; Zhao, S.; Li, C.; Li, J. Low-Resolution Face Recognition in the Wild via Selective Knowledge Distillation. *IEEE Trans. Image Process.* **2019**, *28*, 2051–2062. [CrossRef] [PubMed]
71. Zhu, Z.; Davari, K. Comparison of local visual feature detectors and descriptors for the registration of 3D building scenes. *J. Comput. Civ. Eng.* **2014**, *29*, 04014071. [CrossRef]
72. Aguilera, C.A.; Sappa, A.D.; Toledo, R. LGHD: A feature descriptor for matching across non-linear intensity variations. In Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, Canada, 27–30 September 2015.
73. Li, S.; Amenta, N. Brute-force k-nearest neighbors search on the GPU. In Proceedings of the International Conference on Similarity Search and Applications, Tokyo, Japan, 24–26 October 2015; Springer: Brelin, Germany, 2015.
74. Roth, L.; Kuhn, A.; Mayer, H. Wide-Baseline Image Matching with Projective View Synthesis and Calibrated Geometric Verification. *PFG J. Photogramm. Remote Sens. Geoinf. Sci.* **2017**, *85*, 85–95. [CrossRef]
75. Lin, W.Y.; Wang, F.; Cheng, M.M.; Yeung, S.K.; Torr, P.H.S.; Do, M.N.; Lu, J. CODE: Coherence Based Decision Boundaries for Feature Correspondence. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 34–47. [CrossRef]

76. Cheng, J.; Leng, C.; Wu, J.; Cui, H.; Lu, H. Fast and Accurate Image Matching with Cascade Hashing for 3D Reconstruction. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014.
77. Tolias, G.; Avrithis, Y. Speeded-up, relaxed spatial matching. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011.
78. Jia, K.; Chan, T.H.; Zeng, Z.; Gao, S.; Wang, G.; Zhang, T.; Ma, Y. ROML: A Robust Feature Correspondence Approach for Matching Objects in A Set of Images. *Int. J. Comput. Vis.* **2016**, *117*, 173–197. [CrossRef]
79. Mishkin, D.; Matas, J.; Perdoch, M. MODS: Fast and robust method for two-view matching. *Comput. Vis. Image Underst.* **2015**, *141*, 81–93. [CrossRef]
80. DeTone, D.; Malisiewicz, T.; Rabinovich, A. Superpoint: Self-supervised interest point detection and description. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.