Efficient Feature Matching and Pose-graph Initialization for SfM

Wen Tian Gan , Zong Qian Zhan , Xin Wang*

School of Geodesy and Geomatics , Wuhan University , Wuhan 430079, China
2019302141148@whu.edu.cn, (zqzhan, xwang)@sgg.whu.edu.cn

Keywords: Structure from Motion(SfM), Pose-graph initialization, Efficient feature match, Two-view geometry

Abstract

Over the last decades, Structure from Motion (or image orientation) has been widely studied in the fields of photogrammetry and computer vision, mainly thanks to its feasibility for dealing with various image datasets (such as crowdsourced or UAV images). However, due to the fact that images are becoming easy to obtain, nowadays, it is challenging to deal with large scale of datasets, wherein the feature matching and pose-graph generation are the key limitations in terms of time efficiency. In this work, we proposed an efficient method to accelerate the generation of correspondences and two-view geometries. Specifically, based on some already estimated two-view geometries, unknown two-view geometries can be computed via A* algorithm. Then, the corresponding feature matching can be perform in a guided way using an epiploar-hash bins that is derived from the estimated two-view epipolar geometries. The experimental results demonstrated that, our method can improve the speed of generating pose-graph by 3–4 times comparing to two popular packages (colmap and OpenMVG) and is also faster than one SOTA method of Barath et al., (2021), yet the results of SfM are typically on par with them and reprojection error of our works are even better.

1. Introduction

In the past few decades, Structure-from-Motion (SfM) has been intensively studied in the field of computer vision, photogrammetry. Nowadays, SfM methods can be mainly categorized as incremental and global ones according to their ways to initialize BA (bundle adjustment). The global methods, which consider all images simultaneously, can obtain the similar accuracy as the incremental method, while are much more time efficient. However, when processing large set of disordered images, there still exists limitations regarding consuming time, which are difficult to meet the requirements of quick production of high precision geographic information in large-scale and complex scenes.

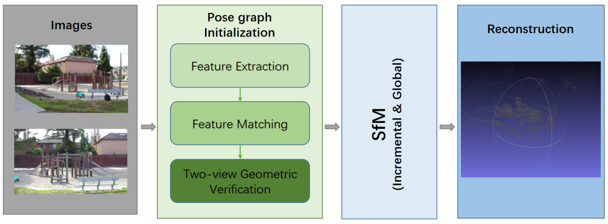
Images
Pose graph Initialization
Feature Extraction
Feature Matching
Two-view Geometric Verification
SfM (Incremental & Global)
Reconstruction

Figure 1. The workflow of conventional SfM pipeline.

As Fig.1 illustrates, the input of conventional SfM typically consist of the following main steps: feature extraction, feature matching, two-view epipolar geometric verification, in which feature matching and two-view epipolar geometric verification is the most time-consuming when dealing with large-scale of image datasets (both of them exhibit quadratic complexity with respect to the number of images). Furthermore, feature matching possesses a quadratic worst-case time complexity, as it is contingent upon the number of extracted local features on each image. Recently, ample endeavors have been tried to improve the efficiency of feature matching. For example, the well-known ORB-SLAM (Simultaneously Localization and Mapping) often accelerates this process via employing the binary descriptors Rublee et al., (2011) or thresolding the number of input features Mur-Artal et al., (2015). However, for precise SfM reconstruction, this approach frequently results in imprecise camera pose estimations. In general, approximate nearest neighbor algorithms such as kd-trees (Muja and Lowe, 2009) or product quantization Jegou et al., (2010) are employed. By organizing feature descriptors in a hierarchical tree structure, kd-trees accelerate the process of identifying potential matches by efficiently reducing search branches. This can accelerates the matching process, particularly in scenarios involving large datasets or high-dimensional feature spaces, In addition, there also exits Hardware-based speed-ups include using a GPU (Johnson et al., 2019). Nevertheless, these methods fail to consider the fact that the matching process is executed across multiple pairs of images, wherein the relative pose might be pre-estimated via propagation, at least a closed approximation can be calculated before the matching procedure. On the other hand, there is scarce works that address the limitation of inefficient pose graph generation (pose graph consists of node as images and edge between two images that survive from two-view geometric verfication). Barath et al., (2021) investigated the heuristic A*-based (Hart et al., 1968) traversal algorithm, estimate two-view geometry for unsolved image pair via propagation on a visible path, which then make feature matching more ”light-weight” by leveraging the pre-estimated two-view epipolar geometry.

To cope the discussed limitations, this paper proposes a method to improve the feature matching and initial pose graph generation algorithms for Structure from Motion (SfM). Similar to Barath et al. (2021), our main idea is to avoid unnecessary massive two-view epipolar geometry of unsolved pairs via using the information of already estimated two-view epipolar geometry and improve feature matching using a guided hashing matching strategy, but, we improve the pose graph via using sevaral orthogonal minimum spanning trees (MSTs) to build a more complete pose graph. In particular, on the contrary to Barath et al. (2021), our pose graph generation begins two-view epipolar geometry based on the image pairs that are on ortho-

[*] Corresponding author

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLVIII-1-2024-165-2024

gonal MSTs. And then, for all the remained unsolved pairs, A* is used to find a visible path to propagate the two-view geometry. Finally, feature matching is accelerated using a guided hashing matching with estimated two-view geometry. The main contributions of our work are listed as follows:
First, to generate a more complete pose graph faster, based on image similarity degree, several orthogonal MSTs are built as an pre-built initialization of pose graph.
Second, to verify unknow two-view geometry more time efficient, a heuristic A* algorithm is employed to find visible path between these two views, based on which the corresponding two-view geometry is estimated via propagation along the path.
Third, to speed up the time efficiency of feature matching, a hashing strategy based on epipolar lines (calculated from the estimated relative pose) is utilized to narrow down the potential candidate matchable points.

The rest of the paper is organized as follows. Some related works are studied in section 2. In the next section 3, we provide a detailed explanation of our method. Section 4 report our experimental results. Finally, conclusion and future work are outlined in section 5.

2. Related work

In this section, three relevant topics in the context of pose graph generation are reviewed. Firstly we introduce the related work of matchable image pair retrieval method (section 3.1), then the estimating poses methods are investigated (section 3.2), in the end, state of the art approaches in accelerating image matching are introduced (section 3.3).

2.1 Efficient match pair retrieval

Large-scale unordered image data set always exhibit the characteristics of high flexibility, high timeliness, and high resolution (Jiang et al., 2021), thus results in a significant amount of redundant computation overhead. Ample image retrieval methods are used to select most similar image pairs to avoid this problem. Generally speaking, the image match pair retrieval issue can divided into two part: image representing with local or global features and efficient indexing structure.
Firstly, in the field of image representing, some handcrafted local features such as SIFT (Lowe, 2004), SURF (Bay et al., 2006) and ORB (Rublee et al., 2011) are used to detect descriptors that is invariant for scaling, rotation, and partial invariant for viewpoint changes. However, the feature descriptors computed by these methods is of high dimension, resulting in a complex computation process. On the other hand, the global feature extraction method based on CNN has been widely studied. VGG (Simonyan and Zisserman, 2014) leverages several stacking convolution layers to extract high dimensional features from images. ResNet (He et al., 2016) using residual layers to mitigating vanishing gradient issues, through skip connections, it enables training of deeper networks and extracting hierarchical image features. Recently, the transformer model based approach is used for a wide variety of tasks. The ViT (Dosovitskiy et al., 2020) utilizes self-attention mechanisms to transform image patches into sequence embedding, effectively extracting global features to representing image.
Secondly, after presenting the image with features, we need to calculate the similarity between images in order to rank and retrieve related view pairs. The most popular ANN searching (Arya et al., 1998) is an NN (nearest neighbor) searching problem based on a vocabulary-tree image retrieval method. In 2003, Sivic and Zisserman introduced the BoW (bag-of-words) approach by considering the local features (Sivic and Zisserman, 2003), which creating a relationship between visual words and local features and has been used extensively in many software such as COLMAP and ORB-SLAM. However, the efficiency of searching high-dimensional descriptors using tree-based methods will be significantly reduced, sometimes even without improvement compared to brute-force search. The locality-sensitive hashing LSH (Indyk and Motwani, 1998) transforming high-dimensional vectors into binary codes via hash functions. These methods above require a mount of memory and initialize time and they only consider local features and there are very few efficient retrieval method which can applied to the global features based on CNNs.

2.2 Robust pose estimate

Using effective sampling strategies can accelerate the process of robust pose estimation. Many algorithms aim to improve the RANSAC method and have achieved certain success because they can quickly find samples that happen to contain all inliers. The PROSAC (Chum and Matas, 2005) algorithm, which sampling from the continuously increasing set of best corresponding points, can reduce computation resource and improve processing speed. The MAGSAC (Barath et al., 2019) is also an enhanced version of RANSAC, introduces the concept of adaptive guided sampling. This means that in each iteration, MAGSAC dynamically selects samples based on the current estimate of model quality, rather than entirely random selection. Based on MAGSAC algorithm, MAGSAC++ (Barath et al., 2020) was proposed to extends MAGSAC's capabilities, it is more accurate and faster by an order of magnitude compared to the original method. The NAPSAC (N Adjacent Points Sample Consensus) (Torr et al., 2002) algorithm assuming that the real-world data often are spatially coherent and initially localized minimal samples are more likely to be all inliers. P-NAPSAC combines the advantages of PROSAC and NAPSAC, which sampling locally and then gradually transitioning to global sampling. All above the methods are based on the isolated two-view robust pose estimation scene, instead, our method use the information on some subset of the hole image pairs where some images are matched more than once.

2.3 Accelerating feature matching

There are many ways to accelerating the speed of feature matching process. Such as using a binary descriptors like ORB (Rublee et al., 2011), or reducing descriptors dimension like PCA method.However, this usually leads to lower precision in camera pose estimation results. The other solution is use index-based methods, etc, k-d tree, LSH (Locality Sensitive Hashing). Utilizing tree-based or hash-based index structures can accelerate the feature matching process. A faster approach is to use algorithms that support hardware acceleration such as CUDA. However, none of these methods aiming at reducing the number of feature matching executions. In fact, the approximate relative pose can be infer from the existing pose graph, and then guiding the feature matching process, we only need to match feature points that are truly likely to be helpful for estimating the pose.

3. Methods

The goal of this work is to propose an solution that can significantly improve the time efficiency for generating pose graph for image dataset while maintaining the performance of subsequential SfM processing, i.e., our method can be expected to more time efficiently yield input for SfM.

The overall workflow of our method is shown as Fig.2, the green parts denote our main works. Firstly, based on the similarity matrix estimated by the inner-product of global feature ResNet50 and GeM (Radenović et al., 2018) aggregation, several orthogonal MSTs are found if there exist no MSTs in the pose graph. Secondly, a starting pose graph is generated from the edges of these orthogonal MSTs, vanilla matching using SIFT and KNN is applied for correspondences which are then employed to compute two-view geometry. Third, unsolved image pairs are input according to the similar degree from high to low, visible check is then run to determine whether unsolved image pair can be connected by the already solved edges, and A* method is used to find the visible paths, from which the relative orientations is propagated to estimate unsolved image pair. If the visible check fails, then vanilla match is again used, otherwise, we alternate to the guided feature matching. In general, the third step is repeated until all image pairs are explored. In the next subsections, we will introduce three relevant main components.

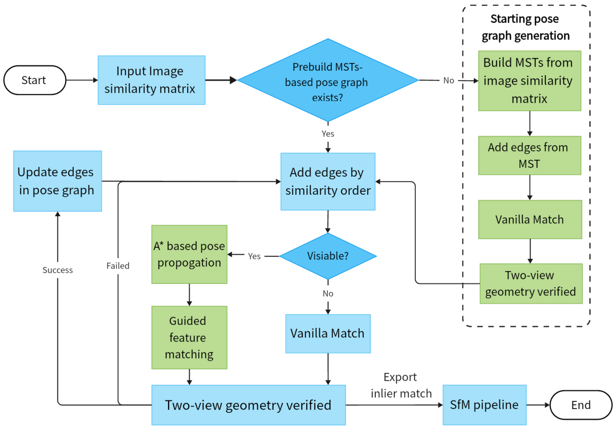
Start
Input Image similarity matrix
Prebuild MSTs-based pose graph exists?
No
Yes
Starting pose graph generation
Build MSTs from image similarity matrix
Add edges from MST
Vanilla Match
Two-view geometry verified
Update edges in pose graph
Add edges by similarity order
A* based pose propogation
Yes
Visiable?
No
Success
Failed
Guided feature matching
Vanilla Match
Two-view geometry verified
Export inlier match
SfM pipeline
End

Figure 2. The efficient pose graph construction and feature matching pipeline.

3.1 Generation of orthogonal MSTs and starting pose graph

To generate a complete pose graph which means more images are included as possible as it can be, in the beginning of our method, based on image similarity matrix, the Minimum Spanning Tree (MST) is used to obtain an optimal set of edges that can connect all images. Specifically, assuming that all images form an undirected graph G with weighted edges (via image similarity degree, S_{ij} represents the similarity score between the images i and j in the similarity matrix). We assign each edge a weight of $1/S_{ij}$, which means the higher image pair similarity score corresponds to lower cost. The total cost can be presented as $cost(G) = \sum_{(i,j)\in G} 1/S_{ij}$, the goal of MST is to find an optimal path that can: first, connect all images; second, the corresponding $cost(MST)$ is as minimum as possible.

To obtain a MST, firstly, each image is considered as a root node of different trees in the graph G, and one edge (image pair) has two vertices of root node. Secondly, retrieve the edge (image pair) with the highest similarity degree from the estimated similarity matrix, if two vertices of the edge belong to different trees, then we merge the corresponding root nodes into one tree and add the edge into the MST that is to be generated, otherwise it continue to process the next image pair with second highest similarity degree. Thirdly, repeat the second step until only one tree left in the graph G, which is the final MST. Finally, a set of edges $T = \{(i, j)|i, j \in G\}$ that includes all images and has the lowest cost is found.

In our work, instead of just using one MST, we propose to use multiple orthogonal MSTs, the main reasons are: first, for only single MST, the visible path retrieval might degrade due to the limited length of explored connecting edge via A* method, as a consequence, the vanilla matching is applied due to no visible path is found, which is very time-consuming; Second, the single MST might be corrupted by outliers of two-view geometry estimation along the edges, this may lead to poor accuracy of estimated relative poses, thereby increasing the epipolar hashing (see section 3.3) cost time. Therefore, to generate a robust pose graph and further improve the time efficiency, we proposed to multiple orthogonal MSTs to build more MSTs in pose graph, in which "orthogonal" means that any two MSTs do not share any single same edge. In our work, at least three orthogonal MSTs are used to improve the possibility that any new image pair can be successfully traversed via the A*.

As shown in Fig.3, when the first MST tree is built, all the edges in the tree are added to the excluded set, the second MST tree is then constructed by traversing the rest image pairs in descending order of similarity score. In this way, we can get multiple orthogonal MSTs by repeating the above steps several times.

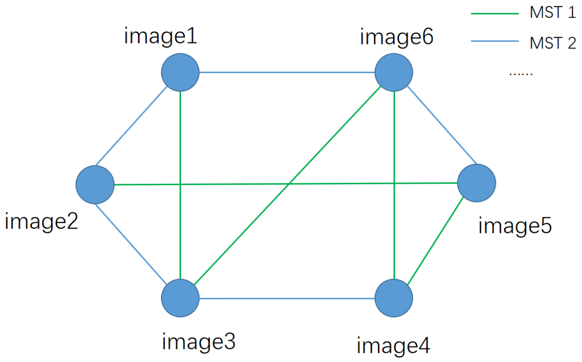

Figure 3. Multiple orthogonal MSTs generation according to the similarity degree among images.

After generating the orthogonal MSTs, the feature matching of each image pair in MSTs uses the vanilla match strategy (such as FLANN), based on which two-view epipolar geometry is estimated, and the starting pose graph is formed.

3.2 Visible path determination based on A* and two-view geometry propagation

Generally speaking, estimating the relative pose among image pairs can be a time-consuming process, In the worst case, the maximum number of iterations in RANSAC needs to be run. However, if a pose graph with t edges is built beforehand, the process can be accelerated and a large number of redundant two-view epipolar geometry calculations can be avoided. An effective path can be found between two views using the A* algorithm, which is a well-designed heuristic method. By propagating relative rotation and translation, we can predict a closed relative pose for the new $(t+1)$ image pair. As shown in Fig.4, when the image pair $view_{1,3}$ is added, we can first estimate the pose via path $view_1 - view_2 - view_3$. Note that we’re not aiming for a very high accurate relative pose in this part. Instead, the approximate pose relative can help guide feature matching to be completed more quickly and avoid the need to run heavy RANSAC.

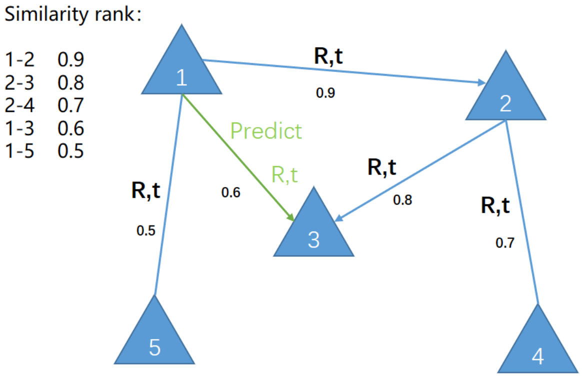

Figure 4. Pose propagation: $view(1,3)$ image pair relative pose can be estimated by $view(1,2)$ and $view(2,3)$.

Based on the starting pose graph built in section 3.1, the rest of image pairs are added by descending order of corresponding similarity degree. According to the characteristics of MST, relative poses among all image pairs can be estimated by the existing starting pose graph. Let us define $W = (I_{v_1}, I_{v_2}, ..., I_{v_n})$ as a path between an unknown image pair (I_{v_1}, I_{v_n}) found by A* algorithm, which I_{v_1} denotes the start view and I_{v_n} denotes the destination view, The pose $\Phi(e_{v_1,v_n})$ can be represented as $(R_{1n}, t_{1n}) \in SE(3)$, which v_1, v_n means the start and destination node in the edge. We can calculate the estimated relative pose between the unknown edge in the pose graph by:

$$\begin{aligned}\Phi(W) &= \Phi(I_{v_1}, I_{v_2}, ..., I_{v_n}) \\ &= \Phi(e_{v_1,v_2})\Phi(e_{v_2,v_3})...\Phi(e_{v_{n-1},v_n}) \qquad (1)\end{aligned}$$

As the number of walking steps between views increases, the error of estimated relative pose are prone to become larger, and it is even possible for the search path to fall into a deadlock. Therefore, we set a threshold for the search depth, and paths that exceed this threshold are considered invalid search results.

3.3 Guided feature Matching with propagated two-view geometry

Feature matching is typically the most time-consuming task, and it possesses a complexity of $O(n^2)$. After the relative posed of start and destination image are solved in the prebuilt pose graph, for one selected keypoint in destination image, it is possible to find matched feature point near the corresponding epipolar line in the source image, thus, a hashing strategy based on epipolar lines is proposed. Once we identify potential candidate features, the speed of feature matching process is expected to be greatly improved.

Specifically, the guided feature matching method contains two parts which are epipolar hash and quick match. As Fig.5 shows, the epipolar hash method use the essential matrix estimated in section 3.2 to find the possible points which lies nearby the epipolar lines projected on the source image. In practice, the features in the corresponding hashed bin (sector of covering the epipolar line on the source image) are considered, in which this bin is centered by the epipole within a certain angle. Due to the characteristic of epipolar geometry, the angle value range may be $[0, \pi)$ (the epipole e_1 is inside image shown as Fig.5) or $[a, b]$ (the epipole e_1 is outside image), which a,b denote the minimum and maximum angle value of the line connecting between the epipole and the corners of image. After getting the range of epipolar line angles, all features on the images can be hashed into different bins. Each bin shares the same epipole and the span of bin’s angle is $\frac{[0,\pi)}{bins\ number}$ or $\frac{[a,b]}{bins\ number}$. Among all bins formed by the line with the epipole, the one that hits the epipolar line is the candidate to be explored. Ultimately, the vanilla feature matching are performed via considering the features on the candidate bin. It should be emphasized that the vanilla feature matching only runs on a small feature set in the image, because feature that are obviously far away from the epipolar line have been excluded.

In sum, by combining the approaches of predict propagated two-view geometry and quick feature matching, we can narrow down the potential candidate points for feature matching, thus speed up the process.

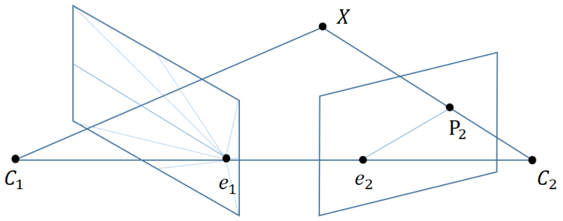

Figure 5. Epipolar hash: feature $p2$ in image $C2$ is assigned to the bin in $C1$ which corresponds to epipolar line.

4. Experiments

To validate the performance of the proposed method, we test several different image datasets (see section 4.1 for more details). To evaluate the time efficiency, two popular package (Colmap and openMVG) and one state-of-the-art method Barath et al., (2021) are compared (section 4.2). Finally, the results of SfM are reported using the evaluation metric of mean track length, mean reprojection error and number of registered images (section 4.3).

4.1 Experimental datasets and setups

Our method is tested on three image datasets of Knapitsch et al., (2017), named as *Family*, *Lighthouse* and *Playground*, Tab.1 lists the information of image size and number of images and Fig.6 shows sample image of these three datasets. Before testing our method, the first step is to compute the similarity degree matrix among images, which is normalized into (0, 1) and with dimensions of n × n, higher value indicates that the two corresponding images are more likely to overlap with each other. To do this, we use ResNet50 as backbone and GeM (Radenović et al., 2018) as aggregation layer that is pre-trained on GLD-v1 (Noh et al., 2017) to extract global feature. After that, similarity degree is computed via cosine distance, and in our experiment image pairs with similarity over 0.5 are eligible for subsequential processing and 3 orthogonal MSTs are used to generate the starting pose graph.

Datasets Name	Image Width	Image Height	Image Numbers
Family	1920	1080	152
Lighthouse	2048	1080	200
Playground	1920	1080	307

Table 1. The detailed information of tested datasets.

Figure 6. A sample image of *Family*, *Lighthouse* and *Playground*.

The number of extracted SIFT features is set at around 6000. For the A* algorithm, the weight of the heuristic parameter is set to 0.8, and the maximum path search depth is set to 5 layers.

4.2 Performance of time efficiency

In this part, we compared the cost time taken for generating pose graph by the state-of-the-art SfM pipelines including Colmap and openMVG, feature extraction time is also reported here. In addition, we conducted a comprehensive comparison with Barath et al., (2021), in which the cost time of each step is discussed.

Datasets	Method	Cost time(s)	
		extract feature	pose graph
Family	our method	128.25	**106.14**
	Barath et al., (2021)	91.83	305.82
	openMVG	273.99	139.13
	Colmap	13.98(GPU)	381.36
Lighthouse	our method	121.18	**104.12**
	Barath et al., (2021)	120.00	149.47
	openMVG	172.13	115.40
	Colmap	15.96(GPU)	439.68
Playground	our method	189.06	**277.31**
	Barath et al., (2021)	180.56	391.10
	openMVG	331.21	308.24
	Colmap	31.86(GPU)	1465.32

Table 2. Comparison of time consuming by different methods. Best results are highlighted in bold.

From Table.2, we can find that our method is always the fast solution when generating pose graph and exhibits significantly reduction in processing time compared to Colmap. Colmap computes pose graph time slowest, primarily due to its utilization of BF-search (brute-force) matching strategy. The second fastest pipeline is the method of OpenMVG, which utilizes the Fast-Cascade-Hashing-L2 matching algorithm. While this approach results in a significant speed improvement, but it typically need a greater amount of memory (Cheng et al., 2014). As the feature extraction step in Colmap benefits from GPU acceleration, the total time for the initialization part was not included in the comparative analysis. Comparing to Barath et al., (2021), we are in general faster, this can be explained by the benefit of the initialization stage using orthogonal MSTs.

To further explore how the proposed method improve Barath et al., (2021), the cost time of several key stages are investigated, including the A*-based visible path-finding, Epipolar Hash matching, vanilla matching, two-view geometric verification. The relevant results are qualitatively shown in Fig.7-9.

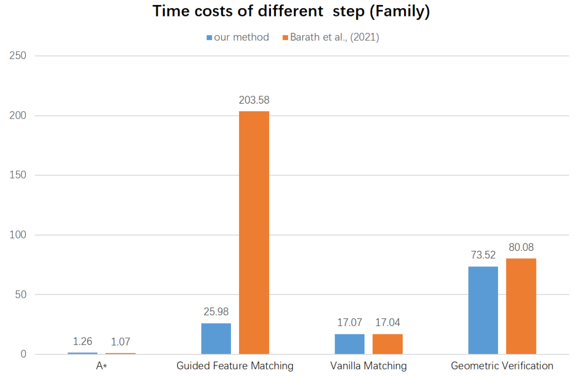

Figure 7. Cost time of *Family*.

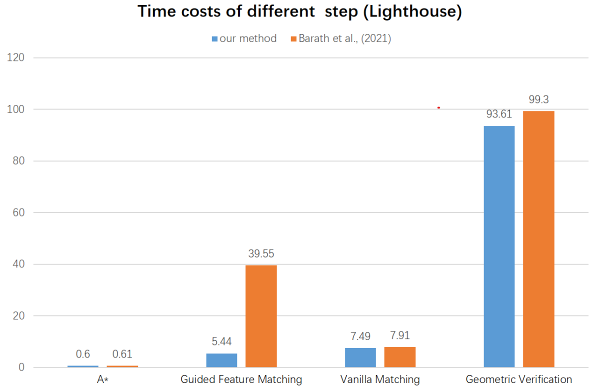

Figure 8. Cost time of *Lighthouse*.

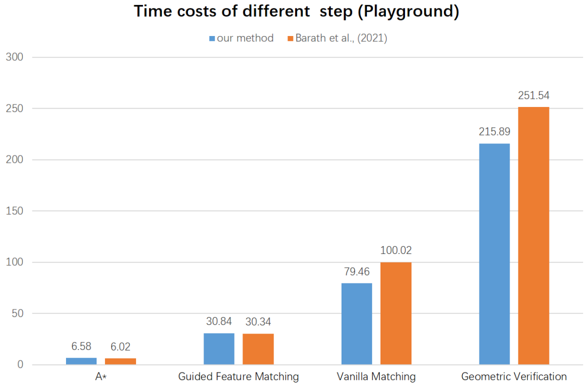

Figure 9. Cost time of *Playground*.

From Fig.7 to Fig.9, the experiment results indicate that the incorporation of multiple orthogonal MSTs can speed up the processing of the corresponding key stages. This improvement arises from the fact that any unknown image pairs can be retrieved at least one visible paths after initializing the pose graph which can avoids some vanilla matching. It is noteworthy that on the Playground datasets, the improvement of epipolar hashing does not happen, whereas the cost time of vanilla matching decreases slightly. After an investigation on the dataset itself, we find that the Playground dataset was captured in a sequential way, i.e., two adjacent images typically overlap with each other, which are more likely to be included by MSTs. Consequently, the image pairs of generated orthogonal MSTs that are built using similarity degree are basically consistent with the image pairs that are used by Barath et al., (2021) for initializing pose graph. Nevertheless, our method still demonstrates lowest processing time.

In sum, our method provide a time efficient solution for feature matching and generating pose graph, which are expected to improve the whole SfM pipeline by fast input estimation.

4.3 Performance of SfM

The performance of SfM relies on the quality of feature matching and two-view geometric results. Therefore, in this section, the popular framework - openMVG is employed as a SfM engine which take the output of our method and Barath et al. (2021) as input to estimate image poses and 3D sparse point cloud. Three evaluation metrics are computed:

Mean Track length (MTL). Track length denotes the number of images that a 3D point can be viewed on. Mean Track length is the averaging track length for all triangulated 3D points. A MTL value can typically imply the quality of feature matching, i.e., higher MTL means better feature matching.
Mean Reprojection error (MRE). The reprojection error represents a geometric discrepancy measured as the distance between a reprojected 2D point and its corresponding feature point on the image. MRE can be used to indicate how good the whole SfM result is consistent to the projection model, such as pin-hole.
Registered images number (RIN). The number of successfully registered images in the photogrammetric block. A higher number of successfully registered images indicates a more complete pose graph.

We primarily employed the pipeline of openMVG and COLMAP for our experiments. For the algorithms mentioned in this paper, we utilized feature points extracted using openMVG as our method's input. Tab.3 presents the quantitative results of these three evaluation metrics. It can be observed that the method proposed in this paper exhibits the highest reconstruction accuracy regarding MRE, while the pipeline based on Colmap always exhibit the longest MTL. In terms of the successfully registered number of images, the method proposed is just slightly inferior to Colmap. For colmap, the inherent feature extraction module is used which generate approximately 3000 more feature points per image, as a result, the corresponding MTL and RIN is always the best as more correspondences are found and more edges are likely to pass the two-view geometric verification. Compared to the original pose graph method Barath et al., (2021), our method shows superior results on all three metrics, which demonstrate the efficacy of the proposed integration of multiple orthogonal MSTs. The visual SfM reconstruction results are shown in Fig.10-12.

5. Conclusion

This paper proposed an efficient solution for accelerating the speed of finding correspondences between images and generating pose graph, thus shrinking the feature matching process time. In details, we made some improvements based on the Barath et al., (2021), before we start the pose graph construction process, a sorting method based on the multiple orthogonal MSTs is employed. The experimental results indicate that compared to several major pipelines (Colmap and openMVG), our method can achieve an speed improvement of average 3 times or even more. Moreover, compared with the Barath et al., (2021), our method exhibiting a higher pose-graph initialize efficiency. Regarding the accuracy of 3D reconstruction, our method has the best performance in terms of reprojection error, indicating that our method's output pose graph is robust and of high quality. However, there is still a lot of space for the improvement of our work, which lies in two directions. On the one hand, we will

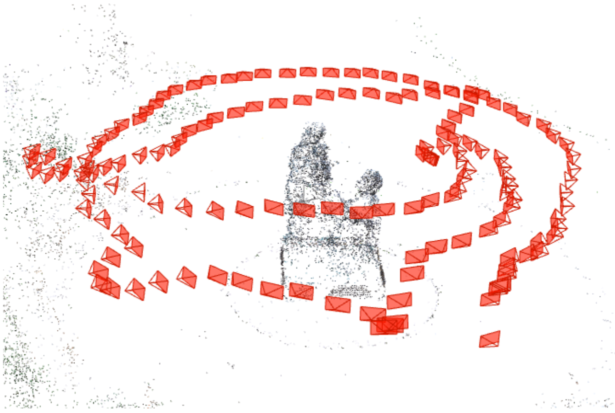
Figure 10. SfM result of *Family*.

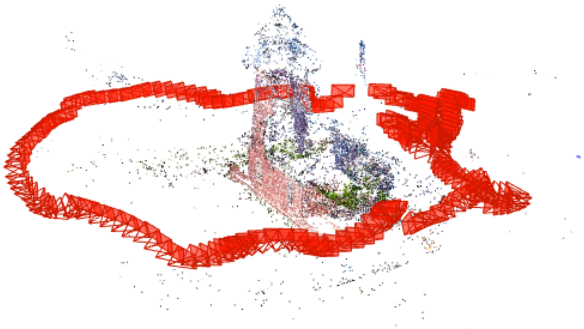
Figure 11. SfM result of *Lighthouse*.

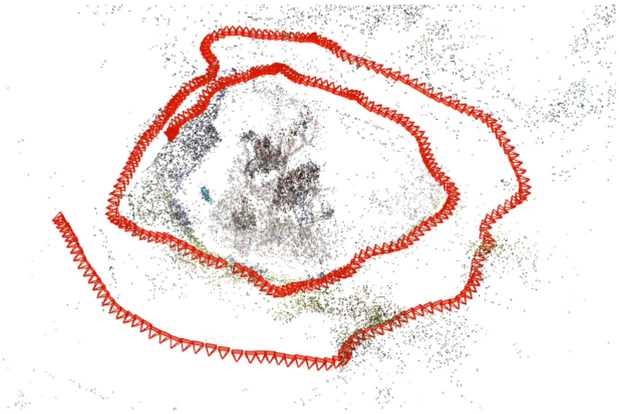
Figure 12. SfM result of *Playground*.

Dataset	Method	MTL	MRE(px)	RIN
Family	our method	7.53	**0.50**	**152/152**
	Barath et al., (2021)	3.80	0.83	141/152
	openMVG	7.02	0.67	152/152
	Colmap	**8.06**	0.87	152/152
Lighthouse	our method	5.41	**0.61**	198/200
	Barath et al., (2021)	5.35	0.63	186/200
	openMVG	5.21	0.90	196/200
	Colmap	**7.74**	0.66	**200/200**
Playground	our method	5.73	**0.49**	**307/307**
	Barath et al., (2021)	5.67	0.68	307/307
	openMVG	4.20	0.99	302/307
	Colmap	**6.03**	0.65	307/307

Table 3. Comparison of SfM results. Best is highlighted in bold.

conduct tests on more challenging datasets, such as those contains over a thousand images or datasets with even greater disorder to test the robustness of our method. On the other hand, instead of calculating the similarity by the cosine distance between extracted global features, the end-to-end method such as NetVLAND can also be used to calculate similarity degree matrix and input into the pose graph.

Acknowledgments

This work was jointly supported Natural Science Foundation of Hubei Province, China (2022CFB727), National Natural Science Foundation of China (42301507) and ISPRS Scientific Initiatives 2023.

References

Arya, S., Mount, D. M., Netanyahu, N. S., Silverman, R., Wu, A. Y., 1998. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM (JACM)*, 45(6), 891–923.

Barath, D., Matas, J., Noskova, J., 2019. Magsac: marginalizing sample consensus. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10197–10205.

Barath, D., Mishkin, D., Eichhardt, I., Shipachev, I., Matas, J., 2021. Efficient initial pose-graph generation for global sfm. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14546–14555.

Barath, D., Noskova, J., Ivashechkin, M., Matas, J., 2020. Magsac++, a fast, reliable and accurate robust estimator. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1304–1312.

Bay, H., Tuytelaars, T., Van Gool, L., 2006. Surf: Speeded up robust features. *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9*, Springer, 404–417.

Cheng, J., Leng, C., Wu, J., Cui, H., Lu, H., 2014. Fast and accurate image matching with cascade hashing for 3d reconstruction. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1–8.

Chum, O., Matas, J., 2005. Matching with prosac-progressive sample consensus. *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, 1, IEEE, 220–226.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S. et al., 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Hart, P. E., Nilsson, N. J., Raphael, B., 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2), 100–107.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Indyk, P., Motwani, R., 1998. Approximate nearest neighbors: towards removing the curse of dimensionality. *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, STOC '98, Association for Computing Machinery, New York, NY, USA, 604–613.

Jegou, H., Douze, M., Schmid, C., 2010. Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence*, 33(1), 117–128.

Jiang, S., Jiang, W., Wang, L., 2021. Unmanned Aerial Vehicle-Based Photogrammetric 3D Mapping: A survey of techniques, applications, and challenges. *IEEE Geoscience and Remote Sensing Magazine*, 10(2), 135–171.

Johnson, J., Douze, M., Jégou, H., 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3), 535–547.

Lowe, D. G., 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60, 91–110.

Muja, M., Lowe, D. G., 2009. Fast approximate nearest neighbors with automatic algorithm configuration. *VISAPP (1)*, 2(331-340), 2.

Mur-Artal, R., Montiel, J. M. M., Tardos, J. D., 2015. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE transactions on robotics*, 31(5), 1147–1163.

Noh, H., Araujo, A., Sim, J., Weyand, T., Han, B., 2017. Large-scale image retrieval with attentive deep local features. *Proceedings of the IEEE international conference on computer vision*, 3456–3465.

Radenović, F., Tolias, G., Chum, O., 2018. Fine-tuning CNN image retrieval with no human annotation. *IEEE transactions on pattern analysis and machine intelligence*, 41(7), 1655–1668.

Rublee, E., Rabaud, V., Konolige, K., Bradski, G., 2011. Orb: An efficient alternative to sift or surf. *2011 International conference on computer vision*, Ieee, 2564–2571.

Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Sivic, Zisserman, 2003. Video google: A text retrieval approach to object matching in videos. *Proceedings ninth IEEE international conference on computer vision*, IEEE, 1470–1477.

Torr, P. H., Nasuto, S. J., Bishop, J. M., 2002. Napsac: High noise, high dimensional robust estimation-it's in the bag. *British Machine Vision Conference (BMVC)*, 2, 3.