ER-SFM: EFFICIENT AND ROBUST CLUSTER-BASED STRUCTURE FROM MOTION*

Zongxin Ye, Wenyu Li, Sidun Liu, Peng Qiao(✉), and Yong Dou

National Key Laboratory of Parallel and Distributed Computing, College of Computer, National University of Defense Technology, Changsha, China
{yezongxin21,wenyu18,liusidun,pengqiao,yongdou}@nudt.edu.cn

Abstract. Structure from Motion (SfM) is a fundamental computer vision technique that recovers scene structure and camera motion from multi-view images. When facing large-scale scenarios, cluster-based methods are commonly employed to improve reconstruction efficiency. However, these methods currently face challenges regarding their limited robustness, redundant computation, and drift. To address these issues, we propose a unified pipeline called ER-SfM, which enhances the three key aspects of cluster-based SfM: image clustering, local reconstruction, and merging. In terms of image clustering, we propose a three-stage image clustering method to ensure adequate and reliable connections between clusters. In the local reconstruction stage, we expedite the reconstruction process by eliminating duplicate point cloud computation. In the final merging stage, we introduce a global merging algorithm without scale ambiguity to address the drift problem. Extensive experimental results demonstrate the superior performance of our method in terms of both robustness and efficiency compared to state-of-the-art methods.

Keywords: Parallel Structure from Motion · 3D Reconstruction · Image Clustering · Global Averaging.

1 Introduction

Structure from Motion (SfM) is a vital process in computer vision that retrieves camera poses and scene point clouds from multiple images. It serves as a critical step for various computer vision tasks, including Neural Radiance Fields [12], Multi-View Stereo [19,17], and 3D Gaussian Splatting [9].

To tackle the efficiency challenge posed by large-scale scenarios, numerous cluster-based methods [22,3,1,18,2,23,6] have been proposed. Typically, these methods utilize images as nodes and construct a connected graph according to the number of feature matches between images, denoted as **image graph**.

* Supported by the Open Fund of Science and Technology on Parallel and Distributed Processing Laboratory (PDL), WDZC20235250106.

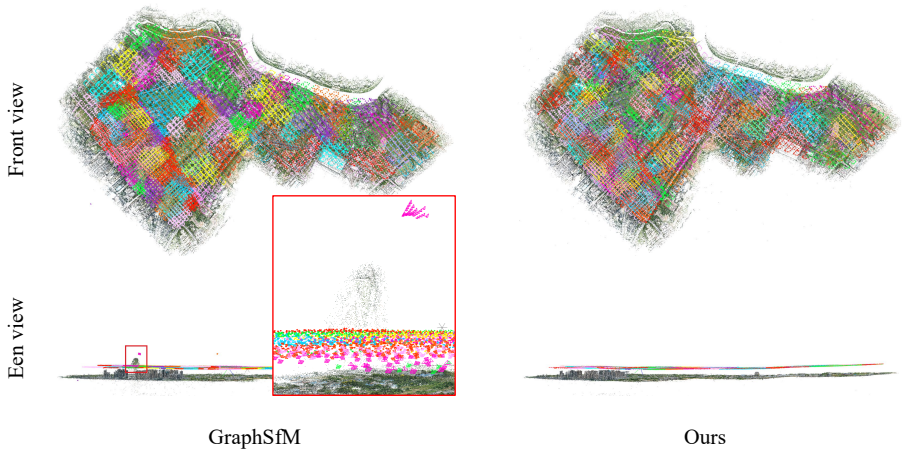


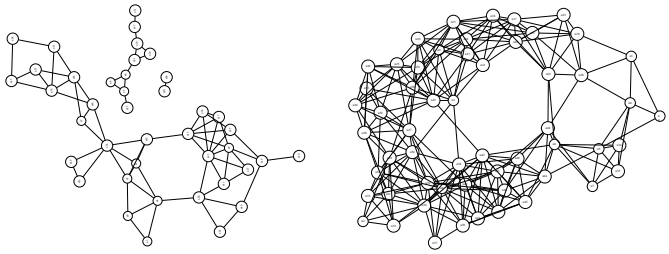
Fig. 1. Reconstructions generated from the Aerial-20k dataset (23458 images). Our method only takes 1.5 hours, while GraphSfM takes 7.5 hours.

Subsequently, the image graph is partitioned into clusters via a graph cut algorithm [16].To facilitate cluster merging, it is typically necessary to expand additional common images within the clusters. Afterwards, clusters are processed in parallel for incremental SfM. Finally, the clusters are merged by utilizing the similarity transformations calculated from the common images shared between them.

Image clustering, local SfM, and merging are three key components of parallel incremental methods, which directly impact the robustness and efficiency of the system.

1) **Image clustering**. Expanded common images are critical to the merging of adjacent clusters. Nonetheless, too many common images increase reconstruction and merging time. To decrease redundant common images and improve the merging efficiency, current methods [2,22,3] limit the number of common images within a cluster and simplify the connected graph. However, these methods do not decouple the impact of common images on efficiency and connectivity, thus often compromising connectivity while pursuing efficiency, as illustrated in Fig. 2.

2) **local SfM**. Current image expansion methods [2,1,3,22] rely solely on image matches or similarity without considering the correlation with the entire cluster. The common images expanded in these methods lack reliability. Moreover, these expansion images are typically registered using incremental SfM, which entails point cloud reconstruction and repeated Bundle Adjustment, significantly diminishing efficiency. In fact, the point clouds corresponding to these expansion images will be reconstructed in multiple clusters, resulting in a large number of unnecessary duplicate calculations.



(a) GraphSfM (b) Ours

Fig. 2. Cluster graphs for Lund Cathedral dataset when cluster size is 30.

3) **Merging**. Existing merging methods [2,1,3,6,18] are mainly path-based. Although minimized by selecting the optimal path to some extent, the cumulative error persists, and the drift problem [4] remains unavoidable, as shown in Fig. 6(a). Furthermore, incorrect selections of duplicated point clouds and camera poses introduce additional errors.

To tackle the problems above and systematically improve efficiency and robustness, we introduce a pipeline, named Efficient and Robust Structure from Motion, coined as ER-SfM. This pipeline focuses on refining the three core stages of parallel incremental SfM, as shown in Fig. 3. At **Image clustering** stage, to decouple the impact of common images on efficiency and connectivity, we determine the common image in two phases. Firstly, instead of expanding the images from a part of selected cluster edges, we expand all images to restore all the connections without limitation on the number of expanded images within each cluster. Secondly, to maintain the connectivity while improving efficiency, we filter the common images between each pair of clusters instead of removing cluster edges directly. As shown in Fig. 2(b), our image clustering strategy maintains connectivity. At **Local SfM** stage, To enhance reliability, we perform local incremental SfM on the cluster before the expansion step (Fig. 3), and then filter expansion images in light of the visibility of the reconstructed point cloud. In order to reduce redundant calculations, we only restore the poses of the expansion images through the perspective-three-point (P3P)[10]. At **Merging** stage, we replace the path-bathed method by proposing a global algorithm to handle the drift problem. However, global methods often face scale ambiguity [13]. Therefore, we use the results of local reconstruction to decouple scale and translation. In addition, we facilitate the correct selection of duplicate elements by considering both the reprojection error and the number of point cloud observations.

The evaluation on multiple datasets shows that ER-SfM is significantly more robuster than state-of-the-art clustering-based method while also achieving a speedup of 2-9 times. Fig. 1 illustrates the reconstruction outcomes of a very large-scale scene, demonstrating the faster reconstruction speed and superior performance of our method.

2 RELATED WORKS

SfM methods are commonly categorized into two categories: incremental methods [13,15] and global methods [21]. Incremental approaches register images one by one, and continuously optimize the camera poses and point cloud using Bundle Adjustment(BA) throughout the process. This makes the incremental approaches more robust and accurate while less efficiency. Additionally, although the incremental approaches typically adopt additional re-triangulation step to mitigate cumulative errors, they are still prone to drift problem. Global approaches simultaneously recover the camera poses of all images by solving rotation and translation averaging problem. Considering all images simultaneously improves the efficiency and leads to no drift. However, global methods are sensitive to outliers, and solving the translation averaging problem is challenging due to scale ambiguity. Moreover,the time and spatial complexity of Global methods is cubic and square respectively in the number of input images.

Due to efficiency constraints, both incremental and global approaches are difficult to handle large-scale scenarios. Certain cluster-based methods leverage parallel computing to enhance efficiency. Bhowmick et al. [1] attempted to solve large-scale SfM using a divide-and-conquer approach. They employed a graph cut algorithm [16] to partition the images into different clusters. Then each cluster is reconstructed separately, enabling a parallel implementation. Finally, the reconstructed clusters were merged into a unified coordinate system. However, the merging algorithm overlooked the issue of drift caused by accumulated errors. To mitigate the errors, some studies [18,6] employed minimum spanning trees (MST) to identify the optimal merging path. GraphSfM [2] further reduces merging errors by exploiting an additional minimum height spanning tree. However, these path-based merging methods cannot fundamentally avoid drift issues. To eliminate drift, some studies [22,23,3] introduce global methods into cluster-based methods. Zhu et al. [22] utilizes the results of local reconstruction for motion averaging to re-estimate the poses and reconstruct the point cloud. However, this approach entails significant repeated computation. This approach was further improved in a later work [23], which introduces a block-wise global method that iteratively applies local motion averaging and global motion averaging. Nevertheless, it faces the challenge of scale ambiguity. AdaSfM [3] performs coarse global SfM to regularize and align the results of local incremental SfM, but this coarse global SfM requires additional sensor measurements as inputs.

3 METHODOLOGY

The pipeline of ER-SfM is shown in Fig. 3. Initially, features and matches are extracted from the images. Then, the images are partitioned into clusters for separate reconstruction. The resulting reconstructions from all clusters are then merged. Finally, a global BA is executed to get obtain final result. To enhance the robustness and efficiency, we optimized three key steps of cluster-based method: image clustering, local SfM, and merging. Each of these steps will be elaborated in subsequent subsections.

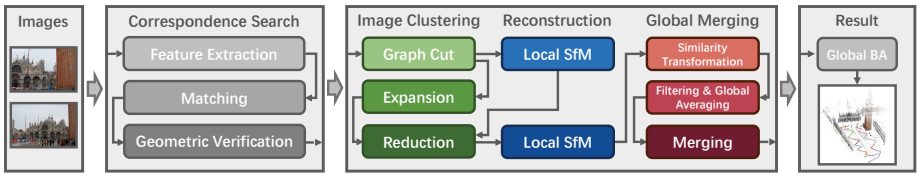


Fig. 3. Pipeline of ER-SfM.

3.1 Image Clustering

As shown in Fig. 4, our image clustering algorithm is divided into three stages: graph cut, expansion, and reduction. We will detail each step as follows.

Graph Cut. In this step, we first build a connected graph named **image graph** based on images and feature matches. In the graph, each node represents an image, while the weight on each edge represents the number of matches. In this way, the image clustering problem is converted into a graph cut problem (Fig. 4(a)). As with some previous approaches[2,1,22], we use the NCuts[16] algorithm to solve this problem.

Image Expansion. To establish connections between clusters, we expand common images to these clusters. The edges that are cut off during graph cut step are referred to as **lost edges**. The images connected by each lost edge belong to two different clusters. For each cluster, all the lost edges connected to it are collected and all images that connected to these lost edges are expanded. The images that are clustered during graph cut step are called **local images**, while the images that are added at expansion step are called **marginal images**. This can be seen in Fig. 4(b). In subsequent local reconstruction stage, the point cloud reconstruction for marginal images can be skipped to improve processing speed.

Image Reduction. We construct a **cluster graph** with cluster as node and the weight N of cluster edge as the number of reliable common images between two clusters. Since local images undergo more BA optimization, their poses are deemed more reliable compared to those of marginal images. A common image between two clusters is considered reliable if it is a local image in either of the two clusters.

To reduce the redundancy caused by image expansion and filter out outliers, previous methods [2,22,3] limit the number of marginal nodes within each cluster and only retain the most reliable cluster edges. As shown in Fig. 2(a), these methods decrease cluster graph connectivity, potentially leading to disconnections. To address this issue, our approach restricts the number of common images within cluster edges instead. In Fig. 4(c), for any image, its weight is the number of reconstruction points that observed within a cluster. Generally, the more observation points, the more reliable the camera pose is. The weight of a common image in a cluster edge is the minimum weight of the image in the two clusters connected by the edge. We set a weight threshold for common images to improve reliability. For each cluster edge, we select the s most weighted

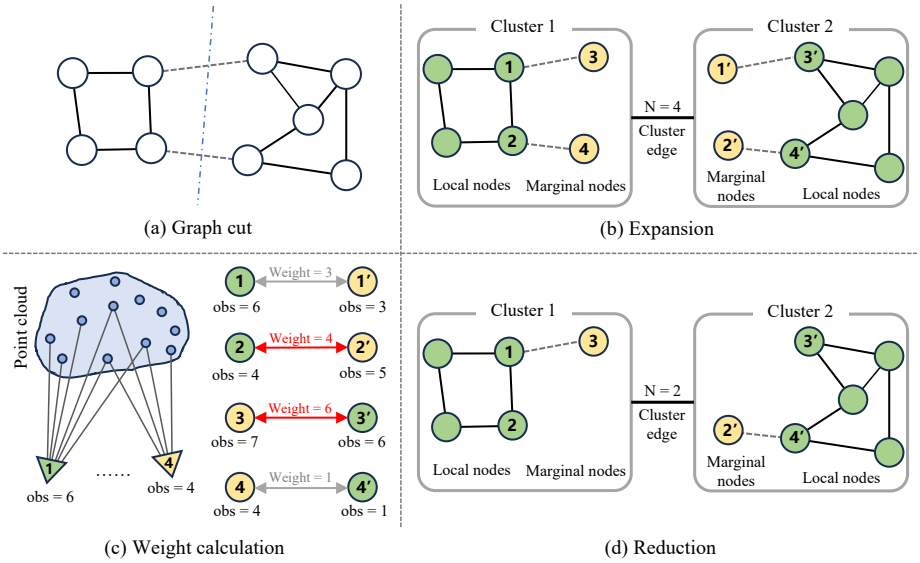


Fig. 4. Pipeline of image clustering. (a) Graph cut. Nodes are depicted as circles, edges as dashed lines, and lost edges as dotted lines. (b) Expansion. In each cluster, local nodes are labeled in green, marginal nodes are labeled in yellow. The common images shared between clusters are assigned the same number. N represents the weight of cluster edge. (c) Weight calculation. triangles represent the cameras corresponding to the images. *obs* indicates the number of points observed. The double arrows indicate the common image, where the weight is the minimum *obs* among the common images. (d) Reduction. Retain selected common images.

common images. Marginal images that are not selected by any cluster edge will be removed from their cluster. To calculate the scale, we remove cluster edges with weight $N <= 2$. By image expansion and reduction, we obtain an adequate number of reliable common images for subsequent merging step. Determining the common image through our extension and reduction steps can decouple the effect of the common image on efficiency and connectivity. Thus, efficiency and connectivity can be ensured simultaneously.

3.2 Local Parallel Reconstruction

To improve robustness, we employ incremental SfM for our local reconstruction. We observe that the point cloud derived from local images assists in selecting common images during the reduction step, while marginal images solely serve to connect clusters, as their corresponding point clouds are reconstructed in other clusters. To address this, we divide the local reconstruction into two stages alternating with image clustering. The first stage is executed before image reduction, involving the reconstruction of poses and point clouds for all local images. In the second stage, following image reduction, we exclusively employ the P3P [10]

to calculate poses for marginal images. It is worth noting that certain methods [22,3] do not guarantee direct connections between all marginal images and the local images. Therefore, the point cloud of some expansion image needs to be reconstructed to assist in restoring the poses of other marginal images.

3.3 Cluster Merging

We finally present a robust and accurate merging algorithm. The key aspect lies in employing global approach to eliminate drift and leveraging the relative scale of clusters to resolve the scale ambiguity problem encountered during global SfM approaches.

Similarity Transformation. We define the subscript ij of the similarity transformation as the transformation from j to i. Relative rotation, translation, and scale are denoted as R_{ij}, t_{ij}, and s_{ij}, respectively. By aligning the common images between clusters, we compute the similarity transformation for each cluster edge. We adopt the method introduced in GraphSfM [2] to compute the relative scale $s_{ij} = (C_j^{k_1} - C_j^{k_2})/(C_i^{k_1} - C_i^{k_2})$, where C_i^k represents the center of camera k in the coordinate system of cluster i. We use

$$R_{ji} = R_j^T R_i \tag{1}$$

$$t_{ji} = R_j^T (s_{ji} t_i - t_j) \tag{2}$$

to calculate relative translation and rotation, where R_i and t_i are the camera poses of the common images. The derivation is shown in the supplementary. Considering that each cluster edge usually corresponds to multiple pairs of common images, RANSAC [7] is used to select the best common images to calculate the similarity transformation.

Graph Filtering. We extend the rotation-based loop constraint [20] to scale and translation to filter outliers. Specifically, we validate all triplets. The triplet with a circular error greater than threshold ϵ is the outlier:

$$d(R_{ij} R_{jk} R_{ki}, I) > \epsilon_r \tag{3}$$

$$|s_{ij} s_{jk} s_{ki} - 1| > \epsilon_s \tag{4}$$

$$\frac{3||s_{ij} R_{ij} t_{ij} + s_{ik} R_{ik} t_{kj} + t_{ik}||_2}{||t_{ji}||_2 + ||t_{kj}||_2 + ||t_{ki}||_2} > \epsilon_t \tag{5}$$

Note that the loop constraint for translation is applied after the rotation and scale averaging. Actually, R_{ji} and s_{ji} in equation (5) is based on the result of rotation and scale averaging.

Global Averaging. Global scale s_i and rotation R_i are calculated independently after graph filtering. For scale averaging, we solve the L1 optimization problem for the equation system $s_{ij} s_j - s_i = 0, i \neq j \; and \; i, j \in 0, 1, .., N$. Where N is the total number of clusters. Global rotations are computed as done by OpenMVG[13], with a least-square minimization that tries to satisfy equations $R_{ij} R_i^T - R_j^T = 0$. Once the global scale and rotation are obtained, the translation averaging no longer has scale ambiguity and can be solved directly via L2

optimization: $R_j t_{ji} - s_{ji} t_i - t_j = 0$, where R_j and s_{ij} are obtained by rotation averaging and scale averaging. To remove ambiguity, we set s_0, R_0 and t_0 to 0, identity matrix I and zero vector 0, respectively.

Cluster Merging. According to the result of global averaging, all clusters are merged into cluster 0. The similarity transformation from cluster i to cluster 0 is

$$R_{0i} = R_0^T R_i = R_i \tag{6}$$

$$s_{0i} = s_0 / s_i = 1 / s_i \tag{7}$$

$$t_{0i} = s_{0i} R_0^T t_i - R_0^T t_j \tag{8}$$

There may be duplicate camera poses and structure points between different clusters. For the repeated points, we calculate their re-projection errors and retain the smallest. Any point with significantly large re-projection error is considered as outlier. For repeated camera poses, we introduce a reliability metric $m = obs/err$, where obs is the number of observed points and err is the mean reprojection error.

4 EXPERIMENTS

We evaluate our algorithm with several datasets: four small scenes (Gerrard Hall, South Building and Person Hall from [15], DTU [8]) consisting of sparse images and five large-scale scenes (Lund Cathedral, Duomo and San Marco from [14], Rubble from [11], Aerial-20k) . We compare with two widely used incremental SfM apprachs, COLMAP [15] and OpenMVG [13], and a state-of-the-art cluster-based method, GraphSfM [2]. Our local reconstruction is conducted via OpenMVG [13], while other incremental SfM methods also work. More details of the experiment settings can be found in the supplementary.

Table 1. Efficiency and accuracy evaluated on datasets with different scales. T represents the reconstruction time, without final BA. N_p, Err represents the number of 3D points and the mean reprojection error, respectively. The best results are highlighted in bold. "-" represents that the result is incorrect.

dataset	images	COLMAP			OpenMVG			GraphSfM			Ours		
		T	N_p	Err	T	N_p	Err	T	N_p	Err	T	N_p	Err
Gerrard Hall	100	105	58261	1.08	151	99195	0.26	131	53672	1.0	**74**	**121664**	**0.22**
South Building	128	298	85509	0.61	338	172007	**0.19**	127	83329	0.58	**91**	**235094**	0.23
Person Hall	337	821	143563	0.65	2386	**434574**	**0.20**	**516**	192359	1.13	528	413921	0.21
Lund	1227	11819	**686995**	0.58	20908	270203	**0.25**	814	638913	0.48	**304**	530593	0.28
Duomo	1805	27143	**1114013**	0.52	41191	231548	**0.25**	2715	893886	0.46	**293**	194305	0.27
San Marco	1657	26012	**518148**	0.69	16525	213729	**0.28**	1380	408522	0.45	**227**	171047	0.30
Rubble	1499	19593	1420417	0.64	24954	1568757	**0.28**	5064	1298571	-	**1265**	**2317768**	0.34
Aerial-20k	23458	-	-	-	-	-	-	27473	9263932	-	**4394**	**9746138**	0.93

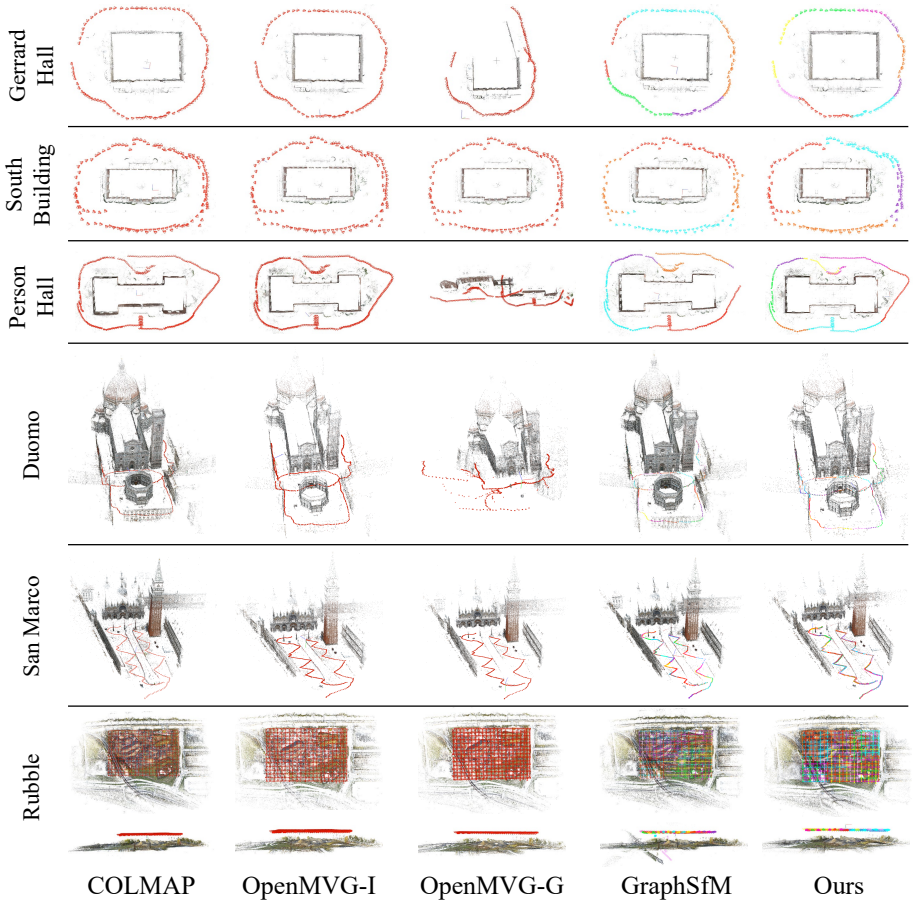


Fig. 5. Reconstruction results on public datasets. The Rubble dataset presents two different perspectives. Global OpenMVG (OpenMVG-G) and GraphSfM perform poorly in some datasets, while our approach overcomes these problems.

4.1 Efficiency and Accuracy Evaluation

As shown in Tab. 1, cluster-based approaches significantly outperform incremental ones. The time complexity of incremental SfM is approximately $O(N^4)$ [5], so clustering not only allows parallel processing, but also reduces the computational load. Importantly, our method outperforms GraphSfM in terms of reconstruction speed, despite the fact that both approaches employ cluster-based methods. On large datasets, we achieved a speedup of 2-9 times. Tab. 2 further shows the reconstruction time for each stage under different cluster sizes. Due to the need to count the number of points observed in images, our graph cutting stage takes longer. During the merging stage, we had more marginal images within the cluster, which resulted in longer time for RANSAC to select similar transformations, thus leading to lower merging speed. However, thanks to our

Table 2. The impact of cluster size on efficiency and accuracy evaluated on dataset Lund Cathedral. T_c, T_{SfM}, T_m, $T_{\sum}$ respectively denotes the time (seconds) of image clustering, local SfM, merging and total time without final BA. Err represents the mean reprojection error. "-" represents reconstruction failure

cluster	GraphSfM					Ours				
size	T_c	T_SfM	T_m	T	Err	T_c	T_SfM	T_m	$T_{\sum}$	Err
20	1	645	16	662	-	17	157	106	280	0.29
40	1	796	17	814	0.48	15	181	108	304	0.28
80	1	879	15	895	0.51	15	200	113	328	0.28
150	1	1154	60	1215	0.52	16	234	118	368	0.28

efficient local reconstruction, our overall time is shorter. In terms of accuracy, our average reprojection error is close to the optimal result. We further validated the accuracy on a DTU dataset with ground truth camera poses. As depicted in Tab. 3, our method achieves more accurate rotation calculations, while the translation results are comparable to the best method.

Table 3. Accuracy evaluated on dataset DTU. R and t respectively represent the rotation and translation of the camera pose. Err represents the mean reprojection error.

dataset	images	COLMAP			OpenMVG			GraphSfM			Ours		
		R	t	Err	R	t	Err	R	t	Err	R	t	Err
scan106	64	0.6182	0.00332	0.462	0.2470	**0.002920**	0.242	0.5312	0.00685	0.436	**0.1502**	0.00296	0.224
scan110	64	0.6759	0.01262	0.536	0.1515	**0.003453**	0.230	0.6466	0.01153	0.531	**0.1493**	0.00377	0.265
scan114	64	0.4190	0.00655	0.547	0.1448	**0.002916**	0.262	0.4388	0.00674	0.495	**0.0953**	0.00316	0.286
scan122	64	0.6356	0.00327	0.560	0.1709	**0.003146**	0.265	0.5555	0.00527	0.522	**0.1355**	0.00332	0.279

4.2 Robustness Evaluation

Fig. 5 shows some of the visualization results in Tab. 1. Note that OpenMVG in Tab. 1 is incremental, and the global OpenMVG is added in Fig. 5 for fair comparison with our global merging algorithm. Our method performs well on all datasets, while the global OpenMVG or GraphSfM have reconstruction errors. Compared to the global SfM, our global merging method shows no scale ambiguity. Furthermore, our similarity transformations go through multiple screenings, ensuring greater reliability. Therefore, our performance is more stable. Despite using the MHT to reduce cumulative errors, GraphSfM may still suffer from drift (Fig. 5 (South building)). In addition, the path selected based on MHT is not reliable, which leads to erroneous merges (Fig. 5 (Rubble)). Fig. 6 shows the reconstruction results under different cluster sizes. GarphSfM still has drift problems when the size is 150, and the results are incomplete when the size is 30, which is due to the disconnected cluster graph (Fig. 2(a)). Our method has better robustness and can still reconstruct correctly even when the size is 10.

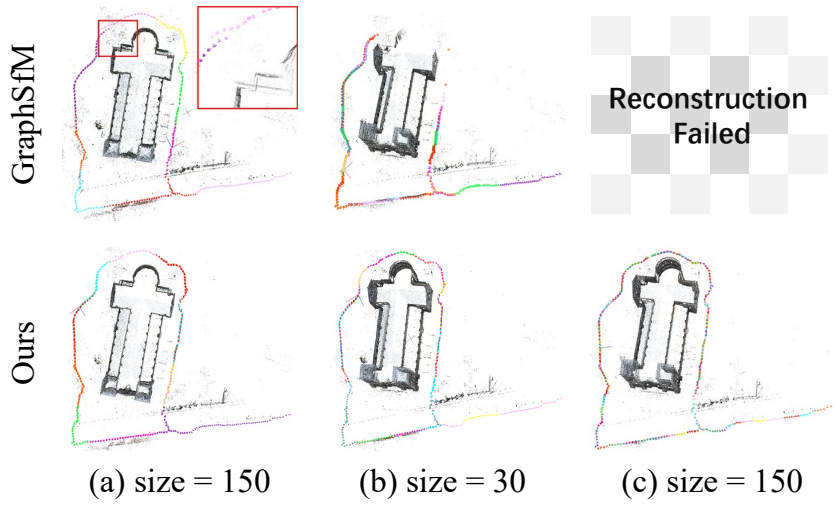


Fig. 6. Reconstructions on Lund Cathedral dataset with different cluster sizes. Each cluster is represented by a distinct color. Compared to GraphSfM, our method does not exhibit drift, and performs well under small cluster size.

5 CONCLUSIONS

In this article, we provide an analysis of the key factors that influence the robustness and efficiency of current cluster-based SfM methods. Moreover, we introduce a unified pipeline called ER-SfM, which enhances the three crucial stages of clustering-based SfM: image clustering, local reconstruction, and merging. Through extensive experiments, ER-SfM is shown to significantly enhance both efficiency and robustness compared to state-of-the-art cluster-based method and maintaining comparable accuracy to incremental approaches.

References

1. Bhowmick, B., Patra, S., Chatterjee, A., Govindu, V.M., Banerjee, S.: Divide and conquer: Efficient large-scale structure from motion using graph partitioning. In: Computer Vision – ACCV 2014. pp. 273–287. Springer (2015)
2. Chen, Y., Shen, S., Chen, Y., Wang, G.: Graph-based parallel large scale structure from motion. Pattern Recognition **107**, 107537 (2020)
3. Chen, Y., Yu, Z., Song, S., Yu, T., Li, J., Lee, G.H.: Adasfm: From coarse global to fine incremental adaptive structure from motion. In: 2023 IEEE International Conference on Robotics and Automation (ICRA). pp. 2054–2061 (2023)
4. Cornelis, K., Verbiest, F., Van Gool, L.: Drift detection and removal for sequential structure from motion algorithms. In: IEEE Transactions on Pattern Analysis and Machine Intelligence. vol. 26, pp. 1249–1259. IEEE (2004)
5. Crandall, D.J., Owens, A., Snavely, N., Huttenlocher, D.P.: Sfm with mrfs: Discrete-continuous optimization for large-scale structure from motion. In: IEEE

Transactions on Pattern Analysis and Machine Intelligence. vol. 35, pp. 2841–2853. IEEE (2013)
6. Fang, M., Pollok, T., Qu, C.: Merge-sfm: Merging partial reconstructions. In: BMVC. p. 29 (2019)
7. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In: Communications of the ACM. vol. 24, p. 381. ACM (1981)
8. Jensen, R., Dahl, A., Vogiatzis, G., Tola, E., Aanæs, H.: Large scale multi-view stereopsis evaluation. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. pp. 406–413 (2014)
9. Kerbl, B., Kopanas, G., Leimkuehler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Trans. Graph. **42**(4) (2023)
10. Kneip, L., Scaramuzza, D., Siegwart, R.: A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In: CVPR 2011. pp. 2969–2976 (2011)
11. Lindenberger, P., Sarlin, P.E., Larsson, V., Pollefeys, M.: Pixel-perfect structure-from-motion with featuremetric refinement. In: ICCV. IEEE (2021)
12. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: Computer Vision – ECCV 2020. pp. 803–806. Springer (2020)
13. Moulon, P., Monasse, P., Marlet, R.: Global fusion of relative motions for robust, accurate and scalable structure from motion. In: 2013 IEEE International Conference on Computer Vision. pp. 3248–3255. IEEE (2013)
14. Olsson, C., Enqvist, O.: Stable structure from motion for unordered image collections. In: Image Analysis. pp. 524–535. Springer (2011)
15. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4104–4113. IEEE (2016)
16. Shi, J., Malik, J.: Normalized cuts and image segmentation. In: IEEE Transactions on Pattern Analysis and Machine Intelligence. vol. 22, pp. 888–905. IEEE (2000)
17. Shi, Y., Xi, J., Hu, D., Cai, Z., Xu, K.: Raymvsnet++: Learning ray-based 1d implicit fields for accurate multi-view stereo. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(11), 13666–13682 (2023)
18. Wang, L., Ge, L.L., Luo, S., Yan, Z.J., Cui, Z., Feng, J.: Tc-sfm: Robust track-community-based structure-from-motion. ArXiv **abs/2206.05866** (2022)
19. Xi, J., Shi, Y., Wang, Y., Guo, Y., Xu, K.: Raymvsnet: Learning ray-based 1d implicit fields for accurate multi-view stereo. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8585–8595 (2022). https://doi.org/10.1109/CVPR52688.2022.00840
20. Zach, C., Klopschitz, M., Pollefeys, M.: Disambiguating visual relations using loop constraints. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 1426–1433. IEEE (2010)
21. Zhang, G., Larsson, V., Barath, D.: Revisiting rotation averaging: Uncertainties and robust losses. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 17215–17224 (2023)
22. Zhu, S., Shen, T., Zhou, L., Zhang, R., Fang, T., Quan, L.: Accurate, scalable and parallel structure from motion. In: Computer Vision – ACCV 2014. arXiv (2017)
23. Zhu, S., Zhang, R., Zhou, L., Shen, T., Fang, T., Tan, P., Quan, L.: Very large-scale global sfm by distributed motion averaging. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4568–4577. IEEE (2018)