A DISTRIBUTED BUNDLE ADJUSTMENT METHOD USING ITERATIVE MOTION AVERAGING

Wu Linghui [1*], Xiao Teng[2*], Li Honghui [1], Deng Fei [1†]

[1] School of Geodesy and Geomatics, Wuhan University, 430079 Wuhan, China – (LinghuiWu, HonghuiLi)@whu.edu.cn, fdeng@sgg.whu.edu.cn
[2] School of Computer Science, Hubei University of Technology, 430068 Wuhan, China - xiao@hbut.edu.cn

Commission 4, WG 7

ABSTRACT
With the wide use of UAVs in urban scenes making image acquisition easier, processing large-scale 3D reconstruction problems have become a basic need of the Structure from Motion(SfM) community. Because of high memory consumption and computation complexity, bundle adjustment (BA) is a significant bottleneck in large-scale 3D reconstruction problems. Distributed strategy is a current research direction. Recent research cut the view graph to optimize sub-scenes in small size and average common cameras and/or 3D points to align them into one in the end. Yet, those methods tried to simply average cameras and/or 3D points, which lack mathematical rigor and tend to get a sub-optimal result. This paper first cuts the view graph and expands it. Then we use the strategy of motion averaging instead of simply averaging points to obtain a robust result. Finally, refined results are achieved by optimizing reprojection errors in the overlapping area. We conduct experiments on UAV data, check the validity of graph cut and expansion, and analyze whether graph size influences the precision computation time. Visualization results show that our method preserves scenes well; precision and visualization of un-expanded scenes prove the necessity of graph expansion; statistical results also indicate the negative infect of large sub-scene size.

KEY WORDS: Unmanned Aerial Vehicle(UAV), distributed bundle adjustment, motion averaging, norm cut, 3D reconstruction.

1. INTRODUCTION

Wide use of UAVs in urban scenes has made image acquisition easier. Today, processing large-scale image-based 3D reconstruction problems has become an essential demand in Structure from Motion(SfM) community . As the last step of SfM, bundle adjustment has been a bottleneck in solving large-scale optimization problems because of memory usage and computational complexity. When BA problems are too enormous, the effects of standard solutions using Levenberg-Marquardt(LM) algorithm seem to be relatively weak.

Researches in recent years tried to split the problem by cutting view graph to solve sub-scenes in a much smaller scale and merge them in the end. Eriksson et al. (2016) used common 3D points to merge sub-scenes and proved feasibility of consensus-based methods mathematically. Natesan Ramamurthy et al. (2017) used both cameras and points and introduced Alternating Direction Method of Multipliers (ADMM) which is one of consensus-based methods. Zhang et al. (2017) adopted ADMM and used cameras only to ease communication consumption. Mayer (2019) improved the method of Eriksson et al. (2016) by adjust points with fixed cameras instead of averaging them. After optimizing sub-scenes locally, common elements are used to align scenes. During alignment, information from common elements is mixed to get a transformation between sub-scenes. But their methods all have a shortcoming of simply trying to exchange and average cameras and/or 3D points. This averaging strategy has defects that lack mathematical rigor and a tendency to obtain sub-optimal results (Cin et al., 2021).

Based on former research, we try to optimize large-scale scene faster than standard solution. On the other hand, we choose a mathematical-strict way to merge all sub-problems to get a robust result.

2. METHOD

Bundle adjustment is generally modelled as a least square problem as:

$$\{\mathrm{I},\mathrm{E},\mathrm{X}\} = \operatorname{argmin} \sum_{i,j} \left\| x_{i,j} - f\left(I_i, E_i, X_j\right) \right\|_2 \qquad (1)$$

where I, E, X and x represent intrinsic parameters, extrinsic parameters, coordinates of 3D points, and coordinates of 2D feature points, respectively; i and j represent indexes of cameras and 3D points. After splitting problem, the new cost function should consider projections from other graphs as

$$\mathrm{G} = \operatorname{argmin} \sum_{k} \sum_{l,l\neq k} \left\| x_{i,j,k} - f(I_i, G_k(E_i), G_l(X_{j,l})) \right\|_2 \qquad (2)$$

where G present the global transformations of sub-graphs aligning them to a unified coordinate system, k and l represent indexes of sub-scenes.

The workflow chart is presented in Figure 1. We take a complete scene as input data and establish a view graph based on matching feature pairs. Edges in the graph contain relative transformations between images, Normalized Cut (NC) is used to cut the graph into multiple sub-scenes. Graph expansion is executed to increase common 3D points number between scenes. We carry out local bundle adjustment for every sub-scene.

Having cut sub-scenes, we could build a cluster graph whose nodes denote sub-scenes and edges contain relative transformations between sub-scenes. Former researches average common 3D points to calculate global transformations for sub-

[*] Equal contribution
[†] Corresponding author fdeng@sgg.whu.edu.cn

scenes implicitly. Firstly, we use common points to update relative transformations with errors due to mismatches or insufficient elements. Secondly, we adopt the strategy of motion averaging and separate estimation of rotations, scales, and translations for more robust global transformations. Estimations are formulated as:

$$A^T \varphi A \Delta\omega_{global} = A^T \varphi \omega_j^T \omega_{i,j} \omega_i \quad (3)$$

$$\log(S_i) - \log(S_j) = \log(S_{i,j}) \quad (4)$$

$$t_j - S_{i,j} R_{i,j} t_i = t_{i,j} \quad (5)$$

where subscript (i, j) means global relations. Estimation of rotations is divided into two steps. Optimization is implemented with Formula (3) without weight matrix $A^T \varphi$ noted as L1 least squares (L1LS) followed by iteratively reweighted least squares (IRLS) to refine. While optimizing translation, rotations and scales are used.

Finally, we project 3D points to images in other sub-scenes able to see them. Distance between projected point and 2D feature is termed overlapped error. Final global transformation is obtained by minimizing sums of overlapped errors.

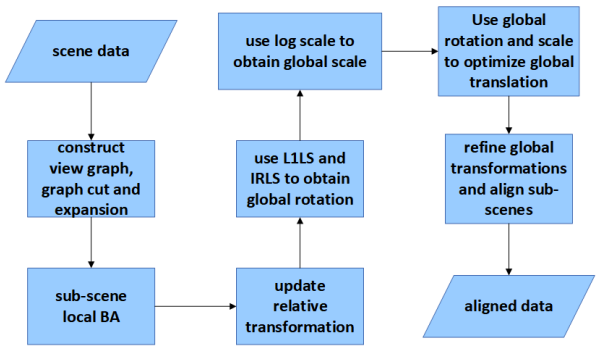


Fig 1. Workflow chart

3. RESULTS AND DISCUSSION

This paper conducts experiments on UAV data whose shooting scenes include city and country. For the experiments, we use a computer with Intel i7-10700KF and 32GB of memory.

3.1 Statistical and visualization results

In this part, we choose re-projection error to measure precision. Results are shown in table 1. Figure 5 shows visualization examples. After distributed BA, shooting scenes are well preserved.

Table 1. Statistical results

Data	image number	rmse/pixel	time used/s
AHMAS	5449	0.930526	258.38
GDZH-1	5043	0.449989	51.9
GDZH-3	3767	0.893410	202.2
HBWX	5706	0.987734	523.9

a) GDZH-1

b) GDZH-3

c) AHMAS

Fig 2. Visualization results

3.2 Graph cut and expansion

This part aims to check the validity of graph cut and graph expansion. We run experiments using the same setting as 3.1 which gives result with best precision only without graph expansion. After graph cut, sub-scenes have neat shapes to keep inside consistency. Meanwhile, overlapped areas between adjacent sub-graph grow larger after expansion which helps preserve conformity between verges of graphs. Skipping expansion makes some rise on rmse. While rmse decreases on AHMAS data, there are holes in scene, as shown in Figure 3. Comparison of rmse is listed in Table 2.

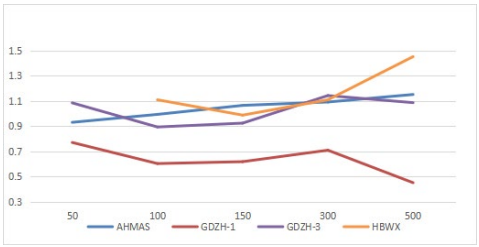
Table 2. Comparison of rmse with or without graph expansion(GE)

Data	with GE	without GE
AHMAS	0.930526	0.812362
GDZH-1	0.449989	1.199
GDZH-3	0.893410	0.91641
HBWX	0.987734	1.42939

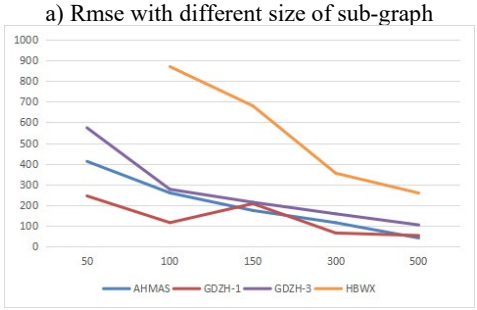
Fig 3. Comparison of results with or without graph expansion

3.3 Influence of graph size

In this part, we compare reprojection error and computational time in the same data with different maximum sizes of sub-scene. Results are shown in Figure 6. Rmse doesn’t have a clear trend to grow or descend with increasing sub-graph size, except a too big size could result in decreasing precision. As listed in Table 3, internal rmse denotes error in the non-overlap area, while overlap denotes overlapped error. Error accumulates bigger on the verge of scene, leading to more considerable overlapped error, so most data reach the best precision with a relatively small size. GDZH-1 has a much bigger size as an exception, but it simultaneously has a remarkably better overlap rmse, which means its error accumulation is moderate. On the other hand, increasing maximum size lower computational time. Therefore, medium size and a comparatively big overlap ratio may be a wise choice.



a) Rmse with different size of sub-graph



b) Time with different size of sub-graph

Fig 4. Rmse and time with different size of sub-graph

Table 3. Best rmse and its corresponding scene size

data	internal rmse/pixel	overlap rmse/pixel	scene size
AHMAS	0.308041	1.072553	50
GDZH-1	0.274357	0.593139	500
GDZH-3	0.279860	1.050771	100
HBWX	0.293848	1.138938	150

REFERENCES

[1] Eriksson, A., Bastian, J., Chin, T. J., & Isaksson, M. (2016). A consensus-based framework for distributed bundle adjustment. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1754-1762).

[2] Natesan Ramamurthy, K., Lin, C. C., Aravkin, A., Pankanti, S., & Viguier, R. (2017). Distributed bundle adjustment. In Proceedings of the IEEE International Conference on Computer Vision Workshops (pp. 2146-2154).

[3] Zhang, R., Zhu, S., Fang, T., & Quan, L. (2017). Distributed very large scale bundle adjustment by global camera consensus. In Proceedings of the IEEE International Conference on Computer Vision (pp. 29-38).

[4] Mayer H (2019) RPBA - robust parallel bundle adjustment based on covariance information.

[5] Cin, A. P. D., Magri, L., Arrigoni, F., Fusiello, A., & Boracchi, G. (2021). Synchronization of Group-labelled Multi-graphs. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 6453-6463).