Zhuo-hao LIU, Chang-yu DIAO, Wei XING, Dong-ming LU, 2020. A low-overhead asynchronous consensus framework for distributed bundle adjustment. *Frontiers of Information Technology & Electronic Engineering*, 21(10):1442-1454.
https://doi.org/10.1631/FITEE.1900451

A low-overhead asynchronous consensus framework for distributed bundle adjustment

Key words: Structure-from-motion; Distributed bundle adjustment; Overhead; Asynchronous consensus; Partial barrier; Bipartite graph summarization

Corresponding author: Chang-yu DIAO
E-mail: dcy@zju.edu.cn
ORCID: https://orcid.org/0000-0001-7744-0889

Motivation

1. Large-scale data pose great challenges to current multi-view 3D reconstruction methods and the bundle adjustment (BA) step exhibits a high computational complexity.

2. Distributed bundle adjustment (DBA) methods break the computation power bottleneck associated with a single computer but has substantial overhead.

3. The overhead originates mainly from synchronous waiting and block partitioning. Synchronous waiting occurs because the optimization time of the subproblems can vary considerably. Block partitioning on a large scene need to deal with thousands of cameras and millions of scene points.

Main idea

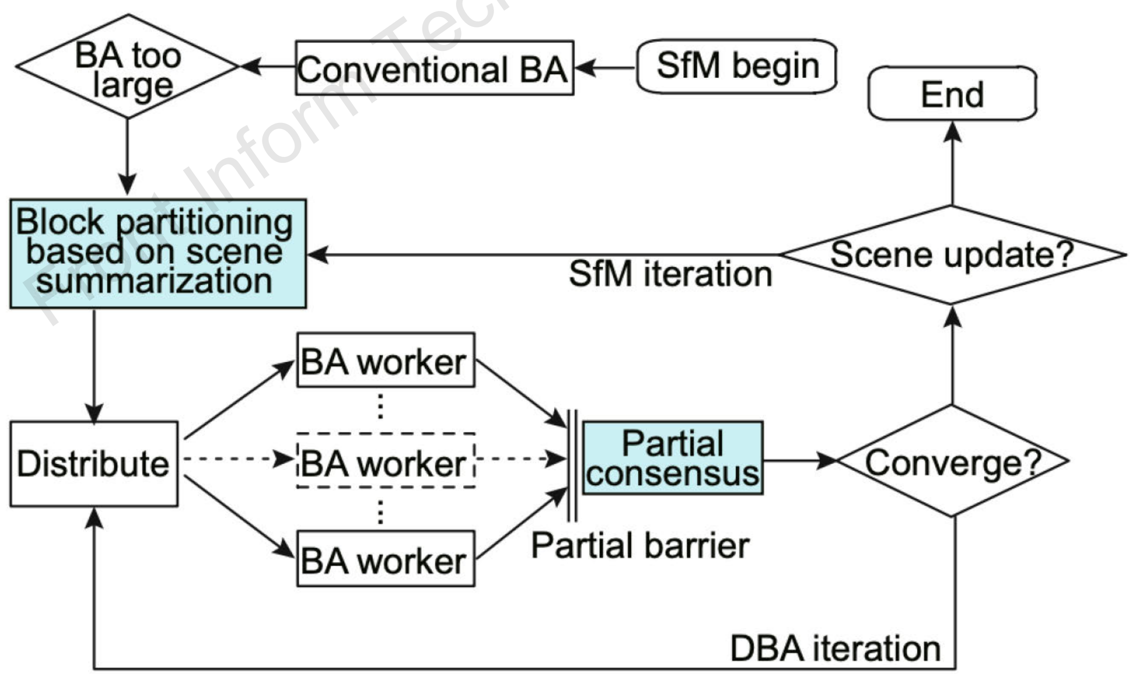
1. Consensus-based distributed bundle adjustment (CDBA) methods assign overlapping parameters to the local subproblems. The information of subproblems is passed to each other with the overlapping parameters being fused and distributed via an iterative process, which can result in an optimal solution to the original BA problem.

2. The subproblems are distributed to multiple computing nodes. Instead of waiting for all the worker nodes to complete the computation, partial consensus is performed to fuse results from faster nodes and reduce waiting time. The improvement is supported by the asynchronous alternating direction method of multipliers (ADMM) theory.

3. To improve the efficiency of the block partitioning step, we summarize similar points and images and then perform clustering on the summarized scene.

Method

1. Our low-overhead consensus framework has two major improvements: the block partitioning step based one scene summarization and the partial consensus step. It can be readily integrated into the current incremental structure-from-motion pipeline.



Method

2. Employing the augmented Lagrangian method and ADMM, the consensus problem can be solved by alternatively optimizing local parameters, the consensus value, and the Lagrange multiplier. In the alternative optimization process, all variables are fixed as constants except for the target variables.

$$
\begin{aligned}
\min \quad & \sum_{k=1}^{K} f(C_k, P_k) \\
\text{s.t.} \quad & \boldsymbol{c}_i^k = \boldsymbol{c}_i, \boldsymbol{p}_j^k = \boldsymbol{p}_j.
\end{aligned}
\longrightarrow
\begin{aligned}
x_k^{t+1} &= \arg\min_{x_k} L_\rho(\boldsymbol{x}, \boldsymbol{y}^t, \bar{\boldsymbol{x}}^t), \\
\bar{\boldsymbol{x}}^{t+1} &= \arg\min_{\bar{\boldsymbol{x}}} L(\boldsymbol{x}^{t+1}, \boldsymbol{y}^t, \bar{\boldsymbol{x}}^t) \\
&= \frac{1}{K} \sum_{k=1}^{K} x_k^{t+1}, \\
\boldsymbol{y}^{t+1} &= \boldsymbol{y}^t + \rho(\boldsymbol{x}^{t+1} - \bar{\boldsymbol{x}}^{t+1}).
\end{aligned}
$$

Method

3. In each iteration, the master node waits until *S* local updates are received. The consensus value is computed in an asynchronous manner. Then, the Lagrangian *y* is updated and sent back to local worker nodes together with the consensus value.

Algorithm 2 Processing in the master node

distribute block data to each worker node
initialize iteration count $t = 0$
repeat
  repeat
    wait
  until receive local parameter update x_k^{t+1} from all worker nodes $k \in \{k_s\}$
  compute consensus $\bar{\boldsymbol{x}}^{t+1}$ as Eq. (5)
  update $\boldsymbol{y}^{t+1}$ as Eq. (6)
  send y_k^{t+1} and $\bar{x}_k^{t+1}$ back to worker node k
  $t \leftarrow t + S/K$
until $t \geq t_\theta$
retrieve non-overlapping parameters from all workers

Method

4. Our block partitioning method takes the bottom-to-up approach, first summarizing similar points and cameras and then cutting the graph on the summarized graph to make the subsequent graph-cut computation more efficient. It is formulated as a bipartite graph summarization problem.

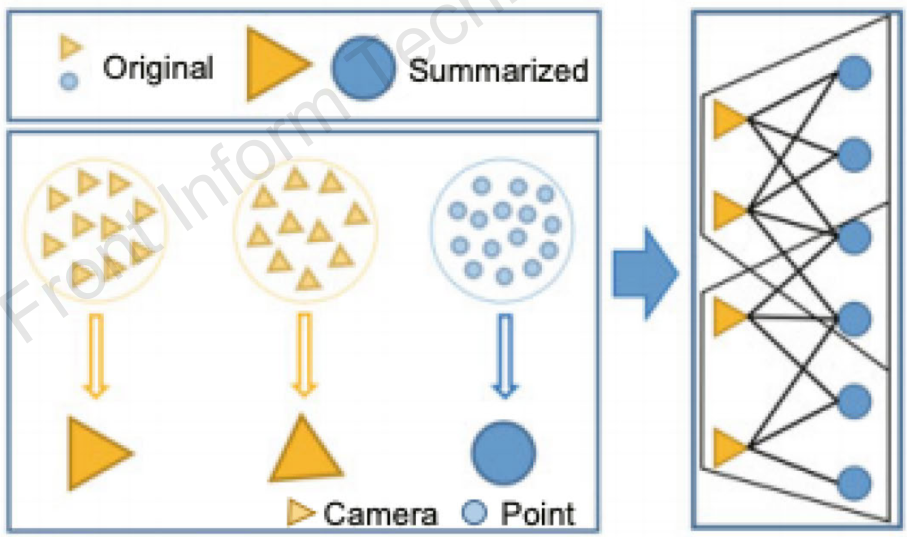


Fig. 2 Illustration of our scene summarization based joint block partitioning method

Major results

- Improved working nodes utilization

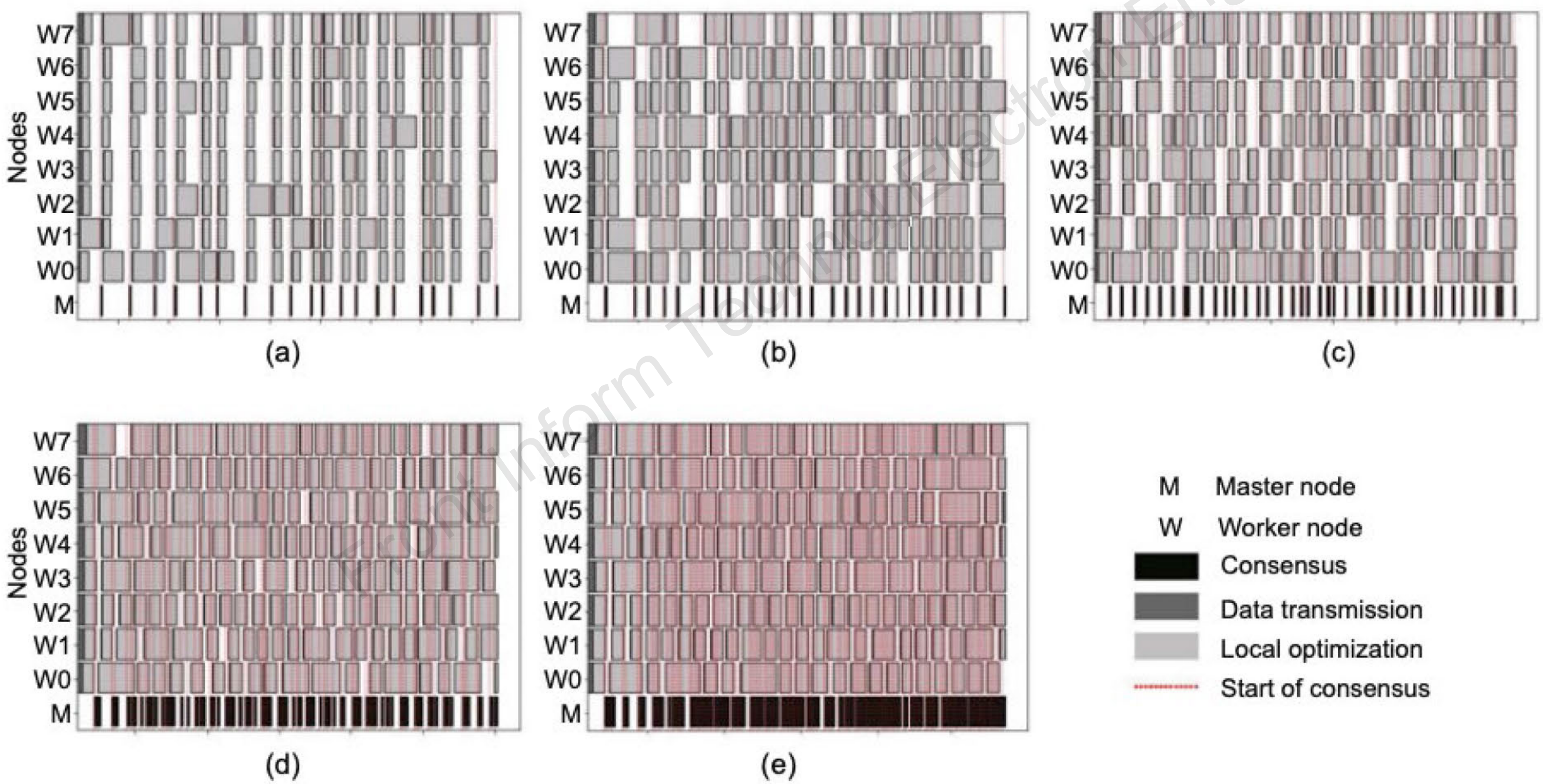
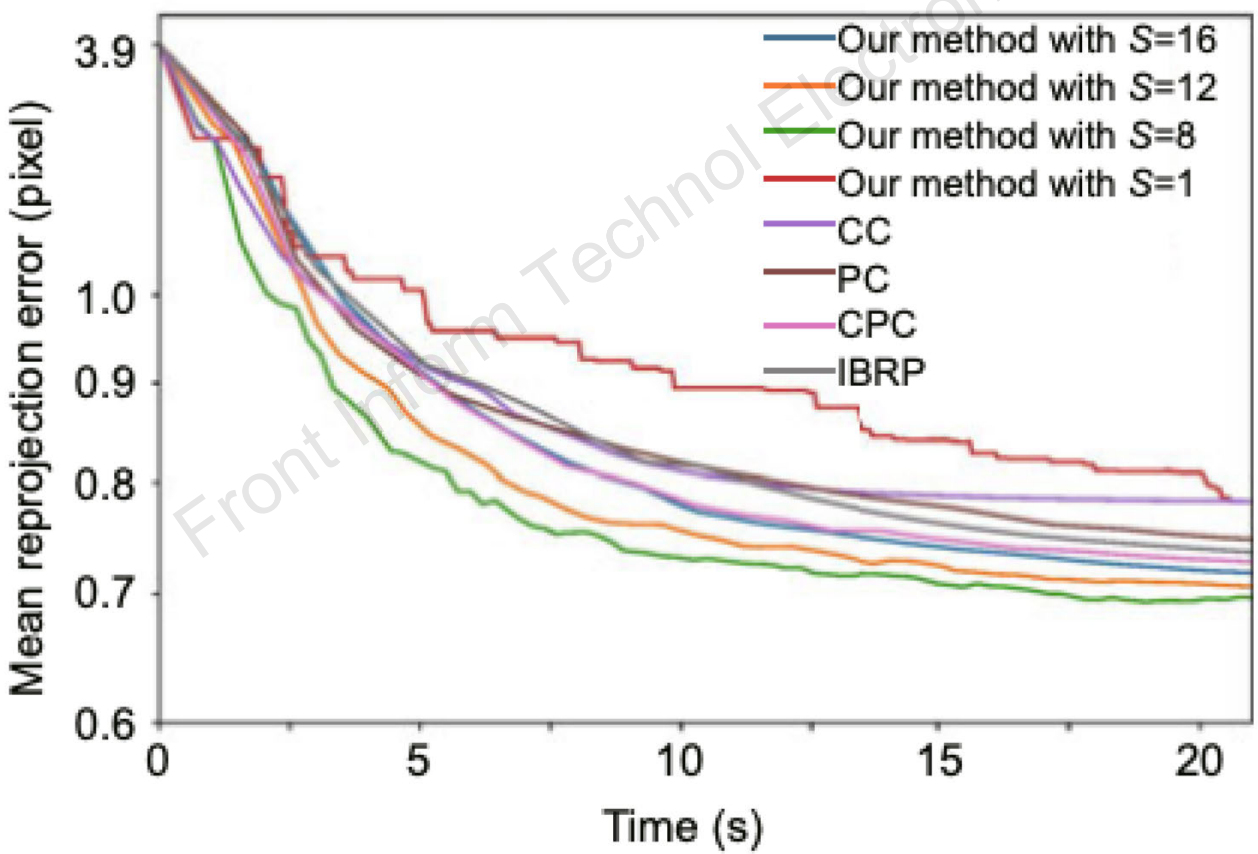


Fig. 3 Workload of the computing nodes in our asynchronous CDBA method: (a) $S = 8$; (b) $S = 6$; (c) $S = 4$; (d) $S = 2$; (e) $S = 1$

Major results

- Higher convergence speed w.r.t. time



Major results

- Quantitative comparison with existing methods

Table 2 Comparison of our method with existing methods

Dataset	K	$\bar{t}_{\mathrm{BA}}$ (s)	$\bar{t}$ (s)	Utilization rate					Mean reprojection error (pixel)				
				Ours	CPC	CC	PC	IBRP	Ours	CPC	CC	PC	IBRP
trafalgar	8	1.4	0.45	0.82	0.36	0.35	0.38	0.38	0.62	0.75	0.81	0.79	0.77
ladybug	16	4.9	0.64	0.71	0.36	0.35	0.36	0.34	0.78	0.88	0.91	0.89	0.90
venice	16	31	2.60	0.53	0.23	0.20	0.18	0.22	0.81	0.85	0.96	0.95	0.91
final	64	179	4.30	0.55	0.33	0.28	0.24	0.34	1.02	1.25	1.18	1.14	1.14
yg_cave3	64	159	3.80	0.63	0.36	0.35	0.43	0.39	1.81	1.90	2.04	1.98	1.94
yg_cave12	128	265	3.00	0.55	0.35	0.32	0.31	0.35	1.83	1.99	1.94	2.05	1.92

K: number of worker nodes; $\bar{t}_{\mathrm{BA}}$: average time for running one iteration of the conventional BA solver; $\bar{t}$: average time for running one iteration of our method

Application

Below are the results of SfM reconstructions of the two scenes in the CH datasets. For the yg_cave12 dataset, DBA is performed as the final optimization step in the hierarchical SfM pipeline. The global BA is executed 61 times and about 4 min is saved each time compared with the conventional BA solvers.

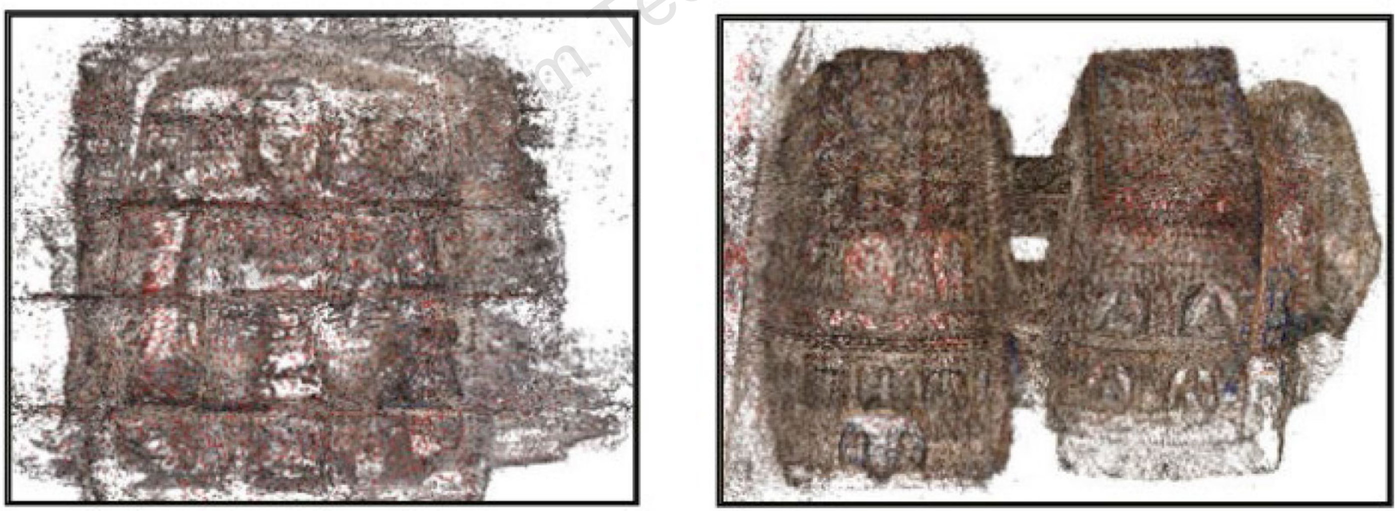


Fig. 6 Visualization of SfM results with our method as the global BA solver: (a) yg_cave3; (b) yg_cave12

Conclusions

1. This study explores to reduce the additional overhead in consensus processing and block partitioning to improve the efficiency of CDBA.

2. We analyze the asynchronous consensus method for DBA and introduce the partial barrier based method to improve the worker node utilization rate. An efficient scene-summarization method for block partitioning is also proposed.

3. Experiments show that our method outperforms existing methods in terms of the worker node utilization rate and convergence speed. Also, the applications in large-scale heritage reconstruction show that our method can be readily plugged into the modern SfM pipeline.