

# Decentralization and Acceleration Enables Large-Scale Bundle Adjustment

Taosha Fan<sup>1</sup>, Joseph Ortiz<sup>2</sup>, Ming Hsiao<sup>3</sup>, Maurizio Monge<sup>3</sup>, Jing Dong<sup>3</sup>, Todd Murphey<sup>4</sup>, Mustafa Mukadam<sup>1</sup>

<sup>1</sup>Meta AI, <sup>2</sup>Imperial College London, <sup>3</sup>Reality Labs Research, <sup>4</sup>Northwestern University

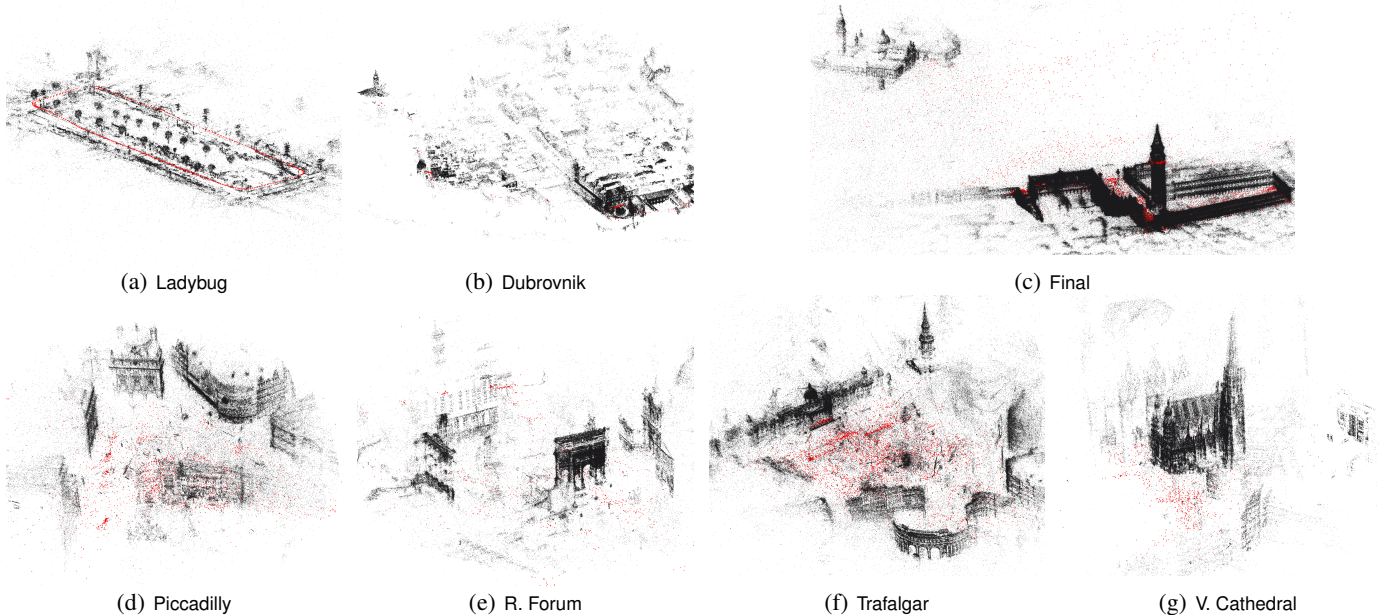


Fig. 1: 3D reconstruction (black point cloud and red camera frames) of our decentralized bundle adjustment method DABA, with 8 devices and the Huber loss, on BAL [1] (top row) and 1DSfM [2] (bottom row) datasets.

**Abstract**—Scaling to arbitrarily large bundle adjustment problems requires data and compute to be distributed across multiple devices. Centralized methods in prior works are only able to solve small or medium size problems due to overhead in computation and communication. In this paper, we present a fully decentralized method that alleviates computation and communication bottlenecks to solve arbitrarily large bundle adjustment problems. We achieve this by reformulating the reprojection error and deriving a novel surrogate function that decouples optimization variables from different devices. This function makes it possible to use majorization minimization techniques and reduces bundle adjustment to independent optimization subproblems that can be solved in parallel. We further apply Nesterov’s acceleration and adaptive restart to improve convergence while maintaining its theoretical guarantees. Despite limited peer-to-peer communication, our method has provable convergence to first-order critical points under mild conditions. On extensive benchmarks with public datasets, our method converges much faster than decentralized baselines with similar memory usage and communication load. Compared to centralized baselines using a single device, our method, while being decentralized, yields more accurate solutions with significant speedups of up to 953.7x over Ceres and 174.6x over DeepLM. Code: <https://github.com/facebookresearch/DABA>.

## I. INTRODUCTION AND RELATED WORK

With the boom of photos and videos in recent decades, bundle adjustment has become one of the most fundamental and useful techniques in robotics [3]–[5], computer vision [6], [7], autonomous driving [8], [9], AR/VR [10] and other areas.

Bundle adjustment is the nonlinear optimization problem of estimating camera parameters and point positions from a collection of images [11]. In the last decade, as image datasets are getting increasingly larger while the computing power of a single device reaches saturation, large-scale bundle adjustment on multiple devices has become more critical than ever.

Even though efficient solvers such as Ceres [12], g2o [13], GTSAM [14], Theseus [15], SymForce [16], etc. have successfully solved small- to medium-scale bundle adjustment by exploiting the problem structure, they all operate on a single central device with a global view of the problem. However, these single-device centralized methods [12]–[16] are unable to leverage parallelism and fail to scale to large problems due to the time and memory limitations.

Even though numerous multi-device methods have been proposed for bundle adjustment to exploit parallel computing, most of them [17]–[22] have to use a central device to maintain consistency. Ni *et al.* [17] propose an out-of-core solution that alternates between solving for independent clusters in parallel and overlapping regions. PBA [20] implements multiple devices to compute preconditioned conjugate gradient steps. STBA [18] stochastically decomposes bundle adjustment using constraint relaxation to approximate Gauss-Newton directions. DeepLM [19] is an efficient GPU-based Levenberg-Marquardt solver with a novel backward jacobian network. MegBA [21] uses fast distributed preconditioned conjugate gradient method

and Schur elimination on multiple GPUs. With a central device to collect information, these multi-device centralized methods, albeit more scalable than single-device ones [12]–[16], are still unsuitable for arbitrarily large bundle adjustment due to the communication bottlenecks of centralization.

Different from single/multi-device centralized methods requiring a central device [12]–[21], decentralized methods for bundle adjustment maintain consistency among devices via purely peer-to-peer communication. Decentralized methods can therefore scale to large problems by avoiding communication bottlenecks associated with a central device. Ortiz *et al.* [23] use Gaussian Belief Propagation to solve bundle adjustment problems on a graph processor. Another popular family of methods are based on Douglas-Rachford (DR) and Alternating Direction Method of Multipliers (ADMM) [24]–[26]. Eriksson *et al.* [24] uses point consensus among subproblems with Douglas-Rachford proximal splitting of the cameras. To reduce the communication overhead, camera consensus and point splitting can be more efficient [25]. Demmel *et al.* [26] use a similar consensus method of parallel block coordinate descent for photometric bundle adjustment. Although decentralized methods [23]–[26] are more scalable, they yield less accurate solutions than centralized methods while being slower and requiring careful parameter tuning. Moreover, they either lack provable convergence or make strict assumptions for it.

We present Decentralized and Accelerated Bundle Adjustment (DABA) to address the compute and communication bottleneck for bundle adjustment of arbitrary scale. Unlike prior work [23]–[26], DABA yields more accurate solutions than centralized methods with greater efficiency and less sensitivity to parameter tuning. DABA also provides convergence guarantees to first-order critical points under less strict assumptions. By reformulating the reprojection error and deriving a novel surrogate function, we decouple optimization variables from different devices to reduce bundle adjustment to independent subproblems on a single device. This is in contrast to [24]–[26] that makes local copies of optimization variables to formulate subproblems. We also implement Nesterov’s acceleration [27], [28] and adaptive restart [29] to improve convergence without loss of theoretical guarantees. On extensive benchmarks with public datasets, DABA converges much faster than decentralized baselines with similar memory usage and communication load. Compared to centralized baselines using a single device, DABA, while being decentralized, yields more accurate solutions with significant speedups of up to 953.7x over Ceres and 174.6x over DeepLM.

## II. BACKGROUND

Bundle adjustment is the problem of jointly estimating camera extrinsics/intrinsics and point positions that represent the scene geometry, given a set of images showing several points from different views. Large-scale problems necessitate data to be distributed across multiple devices. We focus on decentralized bundle adjustment that can work solely via peer-to-peer communication without the need for a central device [23]–[26]. Decentralized methods are preferred if commu-

nication latency and bandwidth are much more expensive than computation, for instance in large-scale bundle adjustment.

Decentralized bundle adjustment can be formulated as an optimization problem minimizing reprojection errors, i.e., residuals between predicted and observed light reprojections [11]. Given  $M$  cameras and  $N$  points partitioned onto  $S$  devices  $\mathcal{S} \triangleq \{1, 2, \dots, S\}$  and the set  $\mathcal{E}$  of reprojection pairs for cameras and points, the optimization problem finds variables  $\mathbf{x}$  which are camera extrinsics/intrinsics  $\{\mathbf{c}_i\}_{i=1}^M$  and point positions  $\{\mathbf{l}_j\}_{j=1}^N$ , and constructs the objective function from individual penalties  $F_{ij}(\mathbf{c}_i, \mathbf{l}_j)$  on reprojection errors:

$$F_{ij}(\mathbf{c}_i, \mathbf{l}_j) \triangleq \frac{1}{2} \rho(\|\mathbf{e}_{ij}\|^2) \quad (1)$$

where  $\mathbf{e}_{ij}$  is any reprojection error defined on the reprojection pair  $(i, j) \in \mathcal{E}$  for camera  $i$  and point  $j$ , and  $\rho(\cdot) : \mathbb{R}^+ \rightarrow \mathbb{R}$  is a robust loss function for outlier rejection. We assume that  $\rho(\cdot)$  is differentiable, concave and nondecreasing. This applies to a broad class of robust loss functions like Huber and Welsch [30]. Then,  $\mathbf{x} \triangleq \{\mathbf{c}_i\}_{i=1}^M \cup \{\mathbf{l}_j\}_{j=1}^N$  partitioned onto multiple devices can be found by minimizing the sum of all penalties over the set  $\mathcal{E}$  of reprojection pairs.

**Problem 1** (Decentralized Bundle Adjustment).

$$\min_{\mathbf{x}} F(\mathbf{x}) \triangleq \sum_{(i,j) \in \mathcal{E}} F_{ij}(\mathbf{c}_i, \mathbf{l}_j) \quad (2)$$

where the optimization variables  $\mathbf{x} \triangleq \{\mathbf{c}_i\}_{i=1}^M \cup \{\mathbf{l}_j\}_{j=1}^N$  are partitioned onto multiple devices.

## III. A NOVEL REPROJECTION ERROR

Even though there are various types of reprojection errors, all of them have cameras and points inseparable, and thus, are difficult to use in decentralized bundle adjustment where optimization variables are on multiple devices. In this section, we present a novel reprojection error to decouple camera extrinsics/intrinsics and point positions that in the next section is leveraged to formulate a surrogate function and create independent optimization subproblems solved in parallel.

In multi-view geometry, the undistorted reprojection ray is equal to the point position in the camera frame up to a scale factor [31]. Therefore, with the undistortion model in [32]–[34], we might assume that there exists a scalar  $\lambda_{ij} \in \mathbb{R}$  for each reprojection pair  $(i, j) \in \mathcal{E}$  such that

$$\underbrace{\left[ \frac{1}{f_i} \mathbf{u}_{ij} \right]}_{\text{undistorted reprojection ray}} - \lambda_{ij} \cdot \underbrace{\mathbf{R}_i^\top (\mathbf{l}_j - \mathbf{t}_i)}_{\text{point position in the camera frame}} = \mathbf{0} \quad (3)$$

where  $\mathbf{u}_{ij} \in \mathbb{R}^2$  is the observed distorted reprojection point on the camera plane,  $f_i$  is the focal length,  $k_{i,1}$ ,  $k_{i,2}$  are the radial undistortion coefficients,  $(\mathbf{R}_i, \mathbf{t}_i) \in \text{SE}(3)$  is the camera extrinsics with  $\mathbf{R}_i \in \text{SO}(3)$  and  $\mathbf{t}_i \in \mathbb{R}^3$ , and  $\mathbf{l}_j \in \mathbb{R}^3$  is the point position. If we represent the camera intrinsics as  $d_{i,1} = f_i$ ,  $d_{i,2} = f_i \cdot k_{i,1}$ ,  $d_{i,3} = f_i \cdot k_{i,2}$ , the undistorted reprojection ray on the left-hand side of Eq. (3) can be reformulated as

$$\mathbf{p}_{ij} = \left[ d_{i,1} + d_{i,2} \|\mathbf{u}_{ij}\|^2 + d_{i,3} \|\mathbf{u}_{ij}\|^4 \right] \in \mathbb{R}^3. \quad (4)$$

With Eq. (4) and a slight abuse of notation for  $\lambda_{ij}$ , Eq. (3) is

$$\mathbf{p}_{ij} - \lambda_{ij} \cdot \mathbf{R}_i^\top (\mathbf{l}_j - \mathbf{t}_i) = \mathbf{0}. \quad (5)$$

In Eq. (5), it is possible to find  $\lambda_{ij} \in \mathbb{R}$  by solving

$$\lambda_{ij} \leftarrow \arg \min_{\lambda_{ij} \in \mathbb{R}} \|\mathbf{p}_{ij} - \lambda_{ij} \cdot \mathbf{R}_i^\top (\mathbf{l}_j - \mathbf{t}_i)\|^2. \quad (6)$$

Since  $\mathbf{R} \in \text{SO}(3)$  and  $\mathbf{R}_i \mathbf{R}_i^\top = \mathbf{I}$  where  $\mathbf{I} \in \mathbb{R}^{3 \times 3}$  is the identity, we have  $\|\mathbf{R}_i^\top (\mathbf{l}_j - \mathbf{t}_i)\| = \|\mathbf{l}_j - \mathbf{t}_i\|$ . If we assume  $\|\mathbf{R}_i^\top (\mathbf{l}_j - \mathbf{t}_i)\| = \|\mathbf{l}_j - \mathbf{t}_i\| \neq 0$  that is common in bundle adjustment [11], then Eq. (6) has a unique solution at

$$\lambda_{ij} = \frac{(\mathbf{l}_j - \mathbf{t}_i)^\top \mathbf{R}_i \mathbf{p}_{ij}}{\|\mathbf{l}_j - \mathbf{t}_i\|^2}. \quad (7)$$

Substituting Eq. (7) into the left-hand side of Eq. (5) yields the following reprojection error:

$$\mathbf{e}_{ij} = \left( \mathbf{I} - \frac{\mathbf{R}_i^\top (\mathbf{l}_j - \mathbf{t}_i) (\mathbf{l}_j - \mathbf{t}_i)^\top \mathbf{R}_i}{\|\mathbf{l}_j - \mathbf{t}_i\|^2} \right) \mathbf{p}_{ij} \in \mathbb{R}^3. \quad (8)$$

where  $\|\mathbf{e}_{ij}\| = 0$  if and only if Eq. (5) holds. Furthermore,  $\mathbf{e}_{ij}$  geometrically results from projecting  $\mathbf{p}_{ij}$  onto the normal plane of  $\mathbf{R}_i^\top (\mathbf{l}_j - \mathbf{t}_i)$ ; see Fig. 2.

Recall that the undistorted reprojection ray  $\mathbf{p}_{ij}$  in Eq. (4) depends on the camera intrinsics  $\mathbf{d}_i \triangleq [d_{i,1} \ d_{i,2} \ d_{i,3}]^\top \in \mathbb{R}^3$ . Then, we conclude from Eq. (8) that  $\mathbf{e}_{ij}$  is a function of camera extrinsics/intrinsics  $\mathbf{c}_i \triangleq (\mathbf{R}_i, \mathbf{t}_i, \mathbf{d}_i) \in \text{SE}(3) \times \mathbb{R}^3$  and point positions  $\mathbf{l}_j \in \mathbb{R}^3$ , with which Problem 1 is well formulated for optimization variables  $\mathbf{x} = \{\mathbf{c}_i\}_{i=1}^M \cup \{\mathbf{l}_j\}_{j=1}^N$ .

The reprojection error above is used in the next section to decouple optimization variables from different devices such that decentralized bundle adjustment is reduced to independent optimization subproblems on a single device.

#### IV. DECOUPLE VARIABLES, REDUCE TO SUBPROBLEMS

Since objective function  $F(\mathbf{x})$  in Problem 1 has optimization variables  $\mathbf{x}$  on multiple devices, it is difficult to minimize with restricted communication. We address this with a surrogate function  $E(\mathbf{x}|\mathbf{x}^{(k)})$  that is an upper bound of  $F(\mathbf{x})$  but equal to it at the current iterate  $\mathbf{x}^{(k)}$ :

$$F(\mathbf{x}) \leq E(\mathbf{x}|\mathbf{x}^{(k)}) \text{ and } E(\mathbf{x}^{(k)}|\mathbf{x}^{(k)}) = F(\mathbf{x}^{(k)}). \quad (9)$$

If  $E(\mathbf{x}^{(k+1)}|\mathbf{x}^{(k)}) \leq E(\mathbf{x}^{(k)}|\mathbf{x}^{(k)})$ , then Eq. (9) suggests

$$F(\mathbf{x}^{(k+1)}) \leq E(\mathbf{x}^{(k+1)}|\mathbf{x}^{(k)}) \leq E(\mathbf{x}^{(k)}|\mathbf{x}^{(k)}) = F(\mathbf{x}^{(k)})$$

where such a  $\mathbf{x}^{(k+1)}$  can be found by solving

$$\mathbf{x}^{(k+1)} \leftarrow \min_{\mathbf{x}} E(\mathbf{x}|\mathbf{x}^{(k)}). \quad (10)$$

Therefore, even though  $F(\mathbf{x})$  is not optimized, Eq. (10) still yields  $F(\mathbf{x}^{(k+1)}) \leq F(\mathbf{x}^{(k)})$ . Furthermore, if the surrogate function  $E(\mathbf{x}|\mathbf{x}^{(k)})$  is in the form of

$$E(\mathbf{x}|\mathbf{x}^{(k)}) \triangleq \sum_{\alpha \in \mathcal{S}} E^\alpha(\mathbf{x}^\alpha|\mathbf{x}^{(k)}) \quad (11)$$

where  $E^\alpha(\mathbf{x}^\alpha|\mathbf{x}^{(k)})$  is a function of  $\mathbf{x}^\alpha$ , i.e., the camera extrinsics/intrinsics and point positions on a single device  $\alpha$ , then Eq. (10) is equivalent to multiple independent optimization subproblems on a single device:

$$\mathbf{x}^{\alpha(k+1)} \leftarrow \arg \min_{\mathbf{x}^\alpha} E^\alpha(\mathbf{x}^\alpha|\mathbf{x}^{(k)}) \text{ for device } \alpha \in \mathcal{S} \quad (12)$$

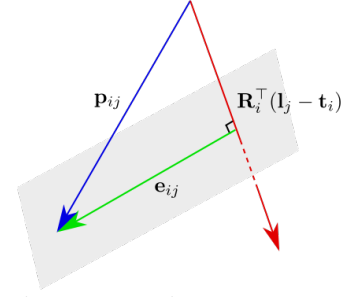


Fig. 2: Reprojection error  $\mathbf{e}_{ij}$  in Eq. (8) geometrically results from projecting  $\mathbf{p}_{ij}$  onto the normal plane of  $\mathbf{R}_i^\top (\mathbf{l}_j - \mathbf{t}_i)$ .

that can be solved in parallel without inter-device communication once  $E^\alpha(\mathbf{x}^\alpha|\mathbf{x}^{(k)})$  is constructed.

The technique above of minimizing an upper bound of the objective function is referred as majorization minimization [35], [36]. As its name suggests, majorization minimization has two steps: 1) constructing a surrogate function satisfying Eq. (9) as “majorization” and 2) optimizing the surrogate function in Eq. (10) as “minimization”. Compared to belief propagation and consensus techniques, majorization minimization is usually faster for empirical implementation and easier for theoretical analysis as long as a proper surrogate function has been found. However, such a proper surrogate function is usually nontrivial. This is even more challenging for decentralized bundle adjustment where the surrogate function is required to satisfy not only Eq. (9) but also Eq. (11). This means that  $E(\mathbf{x}|\mathbf{x}^{(k)})$  must majorize objective function  $F(\mathbf{x})$  while decoupling optimization variables  $\{\mathbf{x}^\alpha\}_{\alpha \in \mathcal{S}}$  that are camera extrinsics/intrinsics and point positions on different devices. Fortunately, such a surrogate function  $E(\mathbf{x}|\mathbf{x}^{(k)})$  exists and its derivation is the main contribution of this section.

Recall from Eq. (2) that the objective function  $F(\mathbf{x})$  is the sum of  $F_{ij}(\mathbf{c}_i, \mathbf{l}_j)$ . If we can majorize each individual  $F_{ij}(\mathbf{c}_i, \mathbf{l}_j)$  while decoupling  $\mathbf{c}_i$  and  $\mathbf{l}_j$ , a surrogate function  $E(\mathbf{x}|\mathbf{x}^{(k)})$  satisfying Eqs. (9) and (11) is yielded. This is in fact possible with the reprojection error in Eq. (8) as long as  $\|\mathbf{l}_j - \mathbf{t}_i\| \neq 0$ , which results in the following proposition.

**Proposition 1.** Suppose  $(\cdot)^{(k)}$  is the  $k$ -th iterate of  $(\cdot)$ . Let

$$P_{ij}(\mathbf{c}_i|\mathbf{x}^{(k)}) \triangleq w_{ij}^{(k)} \cdot \left\| \mathbf{R}_i \mathbf{p}_{ij} + \lambda_{ij}^{(k)} \cdot \mathbf{t}_i - \mathbf{g}_{ij}^{(k)} \right\|^2 + \frac{1}{2} a_{ij}^{(k)}, \quad (13)$$

$$Q_{ij}(\mathbf{l}_j|\mathbf{x}^{(k)}) \triangleq w_{ij}^{(k)} \cdot \left\| \lambda_{ij}^{(k)} \cdot \mathbf{l}_j - \mathbf{g}_{ij}^{(k)} \right\|^2 + \frac{1}{2} a_{ij}^{(k)} \quad (14)$$

where

$$a_{ij}^{(k)} \triangleq \frac{1}{2} \rho(\|\mathbf{e}_{ij}^{(k)}\|^2) - \frac{1}{2} \nabla \rho(\|\mathbf{e}_{ij}^{(k)}\|^2) \cdot \|\mathbf{e}_{ij}^{(k)}\|^2, \quad (15)$$

$$w_{ij}^{(k)} \triangleq \nabla \rho(\|\mathbf{e}_{ij}^{(k)}\|^2), \quad (16)$$

$$\lambda_{ij}^{(k)} \triangleq \frac{(\mathbf{l}_j^{(k)} - \mathbf{t}_i^{(k)})^\top \mathbf{R}_i \mathbf{p}_{ij}^{(k)}}{\|\mathbf{l}_j^{(k)} - \mathbf{t}_i^{(k)}\|^2}, \quad (17)$$

$$\mathbf{g}_{ij}^{(k)} \triangleq \frac{1}{2} \mathbf{R}_i \mathbf{p}_{ij}^{(k)} + \frac{1}{2} \lambda_{ij}^{(k)} \cdot \mathbf{t}_i^{(k)} + \frac{1}{2} \lambda_{ij}^{(k)} \cdot \mathbf{l}_j^{(k)}. \quad (18)$$

Then,

$$F_{ij}(\mathbf{c}_i, \mathbf{l}_j) \leq P_{ij}(\mathbf{c}_i|\mathbf{x}^{(k)}) + Q_{ij}(\mathbf{l}_j|\mathbf{x}^{(k)}) \quad (19)$$



and the equality “=” holds if  $\mathbf{c}_i = \mathbf{c}_i^{(k)}$  and  $\mathbf{l}_j = \mathbf{l}_j^{(k)}$ .

*Proof.* Please refer to [37, App. C.1].  $\square$

With Proposition 1, we can majorize the objective function  $F(\mathbf{x})$  in Eq. (2). Suppose  $\mathcal{E}'$  and  $\mathcal{E}''$  are the sets of reprojection pairs  $(i, j) \in \mathcal{E}$  with cameras and points from the same/different devices. With  $\mathcal{E}'$  and  $\mathcal{E}''$ , we might split these penalty functions  $F_{ij}(\mathbf{c}_i, \mathbf{l}_j)$  in Eq. (2) into two parts:

$$F(\mathbf{x}) = \sum_{(i,j) \in \mathcal{E}'} F_{ij}(\mathbf{c}_i, \mathbf{l}_j) + \sum_{(i,j) \in \mathcal{E}''} F_{ij}(\mathbf{c}_i, \mathbf{l}_j).$$

If we apply Eq. (19) on  $F_{ij}(\mathbf{c}_i, \mathbf{l}_j)$  where  $(i, j) \in \mathcal{E}''$  are inter-device reprojection pairs, the equation above results in

$$F(\mathbf{x}) \leq \sum_{(i,j) \in \mathcal{E}'} F_{ij}(\mathbf{c}_i, \mathbf{l}_j) + \sum_{(i,j) \in \mathcal{E}''} \left( P_{ij}(\mathbf{c}_i | \mathbf{x}^{(k)}) + Q_{ij}(\mathbf{l}_j | \mathbf{x}^{(k)}) \right) \quad (20)$$

where the right-hand side has camera extrinsics/intrinsics  $\mathbf{c}_i$  and point positions  $\mathbf{l}_j$  from different devices decoupled as  $P_{ij}(\mathbf{c}_i | \mathbf{x}^{(k)})$  and  $Q_{ij}(\mathbf{l}_j | \mathbf{x}^{(k)})$ . On the right-hand side of Eq. (20), collecting  $F_{ij}(\mathbf{c}_i, \mathbf{l}_j)$ ,  $P_{ij}(\mathbf{c}_i | \mathbf{x}^{(k)})$ ,  $Q_{ij}(\mathbf{l}_j | \mathbf{x}^{(k)})$  where  $\mathbf{c}_i$  and  $\mathbf{l}_j$  are on the same device  $\alpha$  yields a function  $E^\alpha(\mathbf{x}^\alpha | \mathbf{x}^{(k)})$  where  $\mathbf{x}^\alpha$  is the camera extrinsics/intrinsics and point positions on a single device  $\alpha$ . Then, summing  $E^\alpha(\mathbf{x}^\alpha | \mathbf{x}^{(k)})$  over all the devices  $\alpha \in \mathcal{S}$  and applying the right-hand side of Eq. (20) leads to a surrogate function  $E(\mathbf{x} | \mathbf{x}^{(k)})$  in the form of:

$$E(\mathbf{x} | \mathbf{x}^{(k)}) \triangleq \sum_{\alpha \in \mathcal{S}} E^\alpha(\mathbf{x}^\alpha | \mathbf{x}^{(k)}) = \frac{1}{2} \xi \sum_{\alpha \in \mathcal{S}} \|\mathbf{x}^\alpha - \mathbf{x}^{\alpha(k)}\|^2 + \sum_{(i,j) \in \mathcal{E}'} F_{ij}(\mathbf{c}_i, \mathbf{l}_j) + \sum_{(i,j) \in \mathcal{E}''} \left( P_{ij}(\mathbf{c}_i | \mathbf{x}^{(k)}) + Q_{ij}(\mathbf{l}_j | \mathbf{x}^{(k)}) \right) \quad (21)$$

where  $\xi > 0$  related with  $\|\mathbf{x}^\alpha - \mathbf{x}^{\alpha(k)}\|^2$  is any positive number close to zero and introduced for the purpose of convergence analysis. Furthermore, Eqs. (20) and (21) suggest the following proposition about the resulting  $E(\mathbf{x} | \mathbf{x}^{(k)})$ .

**Proposition 2.**  $E(\mathbf{x} | \mathbf{x}^{(k)}) = \sum_{\alpha \in \mathcal{S}} E^\alpha(\mathbf{x}^\alpha | \mathbf{x}^{(k)})$  and  $F(\mathbf{x}) \leq E(\mathbf{x} | \mathbf{x}^{(k)})$  where the equality “=” holds if  $\mathbf{x} = \mathbf{x}^{(k)}$ , i.e.,  $\mathbf{c}_i = \mathbf{c}_i^{(k)}$  and  $\mathbf{l}_j = \mathbf{l}_j^{(k)}$  for all the cameras and points.

*Proof.* Please refer to [37, App C.2].  $\square$

Proposition 2 indicates that  $E(\mathbf{x} | \mathbf{x}^{(k)})$  in Eq. (21) satisfies Eqs. (9) and (11). Then, with majorization minimization, Problem 1 is reduced to independent optimization subproblems on a single device in Eq. (12) that result in  $F(\mathbf{x}^{(k+1)}) \leq F(\mathbf{x}^{(k)})$ . Moreover, the update rule of Eq. (12) only has peer-to-peer communication between neighboring devices sharing inter-device reprojection pairs when constructing  $E^\alpha(\mathbf{x}^\alpha | \mathbf{x}^{(k)})$  with  $P_{ij}(\mathbf{c}_i | \mathbf{x}^{(k)})$ ,  $Q_{ij}(\mathbf{l}_j | \mathbf{x}^{(k)})$ ; see Eqs. (13), (14) and (21). Besides, camera extrinsics/intrinsics  $\mathbf{c}_i$  and point positions  $\mathbf{l}_j$  are exchanged with other devices if and only if having inter-device reprojection pairs. Thus, Eq. (12) using majorization minimization enables decentralized bundle adjustment.

## V. SPEEDUP AND GUARANTEE CONVERGENCE

The majorization minimization technique in the previous section always decreases the objective function but has slow convergence compared to centralized methods due to its underlying first-order method. This is in fact a well-known issue for first-order methods, for which Nesterov’s acceleration has been applied to get significant speedup [27], [28]. Nesterov’s accelerated methods have shown convergence guarantees only for convex optimization, but if adaptive restart [29] is used the guarantees can be extended to nonconvex optimization [30], [38]–[40]. Thus, we are inspired to implement Nesterov’s acceleration for empirical speedup and adaptive restart for theoretical guarantees. This strategy seems straightforward but the execution is not trivial since the adaptive restart requires a central device to evaluate the objective function, which is impossible for decentralized methods. In this section, we present Nesterov’s acceleration and a novel adaptive restart scheme for decentralized bundle adjustment to speedup and guarantee the convergence.

Nesterov’s acceleration [27] extrapolates the iterate  $\mathbf{x}^{\alpha(k)}$  with the momentum  $\mathbf{x}^{\alpha(k)} - \mathbf{x}^{\alpha(k-1)}$ , which yields the extrapolated intermediate  $\bar{\mathbf{x}}^{\alpha(k)}$ :

$$\bar{\mathbf{x}}^{\alpha(k)} = \text{Proj} \left( \mathbf{x}^{\alpha(k)} + \gamma^{\alpha(k)} (\mathbf{x}^{\alpha(k)} - \mathbf{x}^{\alpha(k-1)}) \right) \quad (22)$$

where  $\text{Proj}(\cdot)$  is an operator projecting optimization variables to corresponding manifolds, and  $\gamma^{\alpha(k)} \in \mathbb{R}$  is the extrapolation ratio of the momentum  $\mathbf{x}^{\alpha(k)} - \mathbf{x}^{\alpha(k-1)}$  and updated with

$$s^{\alpha(k)} = \begin{cases} 1, & k = 0, \\ \frac{\sqrt{4s^{\alpha(k-1)^2} + 1} + 1}{2}, & k > 0 \end{cases} \quad \text{and} \quad \gamma^{\alpha(k)} = \frac{s^{\alpha(k)} - 1}{s^{\alpha(k+1)}}. \quad (23)$$

In terms of bundle adjustment,  $\bar{\mathbf{x}}^{\alpha(k)}$  has camera extrinsics/intrinsics and point positions extrapolated with

$$\bar{\mathbf{R}}_i^{(k)} = \text{ProjRot3D} \left( \mathbf{R}_i^{(k)} + \gamma^{\alpha(k)} (\mathbf{R}_i^{(k)} - \mathbf{R}_i^{(k-1)}) \right), \quad (24a)$$

$$\bar{\mathbf{t}}_i^{(k)} = \mathbf{t}_i^{(k)} + \gamma^{\alpha(k)} (\mathbf{t}_i^{(k)} - \mathbf{t}_i^{(k-1)}), \quad (24b)$$

$$\bar{\mathbf{d}}_i^{(k)} = \mathbf{d}_i^{(k)} + \gamma^{\alpha(k)} (\mathbf{d}_i^{(k)} - \mathbf{d}_i^{(k-1)}), \quad (24c)$$

$$\bar{\mathbf{l}}_j^{(k)} = \mathbf{l}_j^{(k)} + \gamma^{\alpha(k)} (\mathbf{l}_j^{(k)} - \mathbf{l}_j^{(k-1)}) \quad (24d)$$

where  $\text{ProjRot3D}(\cdot) : \mathbb{R}^{3 \times 3} \rightarrow \text{SO}(3)$  is an operator projecting  $3 \times 3$  matrices to  $\text{SO}(3)$ :

$$\text{ProjRot3D}(\mathbf{M}) \triangleq \arg \min_{\mathbf{R} \in \text{SO}(3)} \|\mathbf{R} - \mathbf{M}\|^2 \quad (25)$$

and has a closed-form solution [41], [42]. With  $\bar{\mathbf{R}}_i^{(k)}$ ,  $\bar{\mathbf{t}}_i^{(k)}$ ,  $\bar{\mathbf{d}}_i^{(k)}$ ,  $\bar{\mathbf{l}}_j^{(k)}$ , we obtain  $P_{ij}(\mathbf{c}_i | \bar{\mathbf{x}}^{(k)})$  and  $Q_{ij}(\mathbf{l}_j | \bar{\mathbf{x}}^{(k)})$  in Eqs. (13) and (14). This further yields  $E^\alpha(\mathbf{x}^\alpha | \bar{\mathbf{x}}^{(k)})$  in Eq. (21) as well as the accelerated update rule:

$$\mathbf{x}^{\alpha(k+1)} \leftarrow \arg \min_{\mathbf{x}^\alpha} E^\alpha(\mathbf{x}^\alpha | \bar{\mathbf{x}}^{(k)}) \quad \text{for device } \alpha \in \mathcal{S}. \quad (26)$$

Note that  $E^\alpha(\mathbf{x}^\alpha | \mathbf{x}^{(k)})$  and  $E^\alpha(\mathbf{x}^\alpha | \bar{\mathbf{x}}^{(k)})$  in Eqs. (12) and (26) only differ on whether  $\mathbf{x}^{(k)}$  or  $\bar{\mathbf{x}}^{(k)}$  is conditioned on. Moreover, Eqs. (22) and (24) indicate that the extrapolated intermediate terms  $\bar{(\cdot)}^{(k)}$  requires no inter-device communication, and thus, Nesterov’s acceleration remains decentralized.

Nesterov's acceleration has no provable convergence due to the nonconvexity of bundle adjustment. Fortunately, such an issue can be resolved with adaptive restart [30], [38]–[40]. If the objective value  $F(\mathbf{x}^{(k)})$  is not improved by  $\mathbf{x}^{\alpha(k)}$  from Eq. (26), adaptive restart updates  $\mathbf{x}^{\alpha(k)}$  again with Eq. (12). Recall that Eq. (12) always decreases  $F(\mathbf{x}^{(k)})$ , with which the convergence is guaranteed. However, it is impossible to evaluate  $F(\mathbf{x}^{(k)})$  without a central device. Thus, other metrics than  $F(\mathbf{x}^{(k)})$  are needed to trigger adaptive restart for decentralized bundle adjustment. Inspired by [30], we present a novel adaptive restart scheme by introducing several local per device metrics. This adaptive restart scheme is decentralized but still guarantees the convergence. Due to space limitation, only the main procedure of the adaptive restart scheme is presented while the full analysis is in [37, Apps. C.3 and D.1].

For notational simplicity, let  $\Delta E^\alpha(\mathbf{x}|\mathbf{x}^{(k)})$  be defined by:

$$\Delta E^\alpha(\mathbf{x}|\mathbf{x}^{(k)}) \triangleq -\frac{1}{2}\xi\|\mathbf{x}^\alpha - \mathbf{x}^{\alpha(k)}\|^2 + \frac{1}{2} \sum_{(i,j) \in \mathcal{E}''_\alpha} \left( F_{ij}(\mathbf{c}_i, \mathbf{l}_j) - P_{ij}(\mathbf{c}_i|\mathbf{x}^{(k)}) - Q_{ij}(\mathbf{l}_j|\mathbf{x}^{(k)}) \right) \quad (27)$$

where  $\mathcal{E}''_\alpha$  is the set of inter-device reprojection pairs that has either the camera or the point from device  $\alpha$  but not both. Note that  $F_{ij}(\mathbf{c}_i, \mathbf{l}_j) - P_{ij}(\mathbf{c}_i|\mathbf{x}^{(k)}) - Q_{ij}(\mathbf{l}_j|\mathbf{x}^{(k)})$  in Eq. (27) is the surrogate gap for majorization minimization. With  $\Delta E^\alpha(\mathbf{x}|\mathbf{x}^{(k)})$ , we introduce adaptive restart metrics  $F^{\alpha(k)}$ ,  $\bar{F}^{\alpha(k)}$ ,  $E^{\alpha(k+1)}$  recursively updated on each device  $\alpha$ :

1) At initialization  $k = -1$ , we set  $\mathbf{x}^{\alpha(-1)} = \mathbf{x}^{\alpha(0)}$  and

$$\begin{aligned} F^{\alpha(-1)} &= E^\alpha(\mathbf{x}^{\alpha(-1)}|\mathbf{x}^{(-1)}), \\ \bar{F}^{\alpha(-1)} &= F^{\alpha(-1)}, \quad E^{\alpha(0)} = F^{\alpha(-1)}. \end{aligned} \quad (28)$$

2) For  $k \geq 0$ ,  $F^{\alpha(k)}$ ,  $\bar{F}^{\alpha(k)}$ ,  $E^{\alpha(k+1)}$  are updated with

$$F^{\alpha(k)} = E^\alpha(\mathbf{x}^{(k)}|\mathbf{x}^{(k-1)}), \quad (29)$$

$$\bar{F}^{\alpha(k)} = (1 - \eta) \cdot \bar{F}^{\alpha(k-1)} + \eta \cdot F^{\alpha(k)}, \quad (30)$$

$$E^{\alpha(k+1)} = E^\alpha(\mathbf{x}^{\alpha(k+1)}|\mathbf{x}^{(k)}) + F^{\alpha(k)} - E^\alpha(\mathbf{x}^{\alpha(k)}|\mathbf{x}^{(k)}). \quad (31)$$

Note that  $\eta \in (0, 1]$  in Eq. (30).

With local metrics  $F^{\alpha(k)}$ ,  $\bar{F}^{\alpha(k)}$ ,  $E^{\alpha(k+1)}$  in Eqs. (28) to (31), the adaptive restart scheme on each device  $\alpha$  independently executes the following procedure:

- 1) Solve Eq. (26) to update  $\mathbf{x}^{\alpha(k+1)}$ ;
- 2) If  $E^{\alpha(k+1)} > \bar{F}^{\alpha(k)}$ , update  $\mathbf{x}^{\alpha(k+1)}$  again with Eq. (12).

In spite of no central device to evaluate the objective value  $F(\mathbf{x}^{(k)})$ , such a local adaptive restart scheme actually suffices for provable convergence to first-order critical points; see [37, App. C.3]. Moreover, Eqs. (27) to (31) suggest that  $F^{\alpha(k)}$ ,  $\bar{F}^{\alpha(k)}$ ,  $E^{\alpha(k+1)}$  can be updated if each device  $\alpha$  can communicate with its neighbors. Thus, the adaptive restart scheme is well-suited for decentralized bundle adjustment.

By applying Nesterov's acceleration and adaptive restart while maintaining decentralization, we have achieved empirical speedup without losing theoretical guarantees. The resulting improvements are evaluated in Sec. VII-B.

---

### Algorithm 1 The DABA Method

---

- 1: **Input:** An initial iterate  $\mathbf{x}^{(0)} \in \mathcal{X}$ ,  $\xi > 0$ ,  $0 < \eta \leq 1$ .
  - 2: **Output:** A sequence of iterates  $\{\mathbf{x}^{(k)}\}$ .
  - 3: **for** each device  $\alpha \in \mathcal{S}$  **do**
  - 4:    $\mathbf{x}^{\alpha(-1)} \leftarrow \mathbf{x}^{\alpha(0)}$  and  $s^{\alpha(0)} \leftarrow 1$
  - 5:   initialize  $F^{\alpha(-1)}$ ,  $\bar{F}^{\alpha(-1)}$ ,  $E^{\alpha(0)}$  with Eq. (28)
  - 6: **end for**
  - 7: **for**  $k \leftarrow 0, 1, 2, \dots$  **do**
  - 8:   **for** each device  $\alpha \in \mathcal{S}$  **do**
  - 9:     // Nesterov's acceleration
  - 10:      $s^{\alpha(k+1)} \leftarrow \frac{\sqrt{4s^{\alpha(k)^2} + 1} + 1}{2}$  and  $\gamma^{\alpha(k)} \leftarrow \frac{s^{\alpha(k)} - 1}{s^{\alpha(k+1)}}$
  - 11:     update  $\bar{\mathbf{x}}^{\alpha(k)}$  with Eq. (24) on device  $\alpha$
  - 12:     // Inter-device communication
  - 13:     retrieve  $\mathbf{x}^{\beta(k)}$ ,  $\bar{\mathbf{x}}^{\beta(k)}$  from neighboring devices  $\beta$
  - 14:     // Majorization
  - 15:     construct  $E^\alpha(\mathbf{x}^\alpha|\mathbf{x}^{(k)})$ ,  $E^\alpha(\mathbf{x}^\alpha|\bar{\mathbf{x}}^{\alpha(k)})$  with Eq. (21)
  - 16:     // Minimization
  - 17:     solve  $\mathbf{x}^{\alpha(k+1)} \leftarrow \arg \min_{\mathbf{x}^\alpha} E^\alpha(\mathbf{x}^\alpha|\bar{\mathbf{x}}^{\alpha(k)})$
  - 18:     // Adaptive restart
  - 19:     update  $F^{\alpha(k)}$ ,  $\bar{F}^{\alpha(k)}$ ,  $E^{\alpha(k+1)}$  with Eqs. (29) to (31)
  - 20:     **if**  $E^{\alpha(k+1)} > \bar{F}^{\alpha(k)}$  **then**
  - 21:         solve  $\mathbf{x}^{\alpha(k+1)} \leftarrow \arg \min_{\mathbf{x}^\alpha} E^\alpha(\mathbf{x}^\alpha|\mathbf{x}^{(k)})$
  - 22:         update  $E^{\alpha(k+1)}$  with Eq. (31)
  - 23:     **end if**
  - 24:   **end for**
  - 25: **end for**
- 

## VI. DABA: PUTTING IT ALL TOGETHER

We have presented a novel reprojection error in Sec. III, decoupled optimization variables to reduce decentralized bundle adjustment to independent optimization subproblems in Sec. IV, and improved the convergence with Nesterov's acceleration and adaptive restart in Sec. V. All of these result in DABA, i.e., Decentralized and Accelerated Bundle Adjustment; see Algorithm 1. As previously discussed, DABA is fully decentralized and requires limited peer-to-peer communication. Furthermore, DABA is guaranteed to converge to first-order critical points under mild conditions as the following proposition states.

**Proposition 3.** *If  $\xi > 0$  and  $0 < \eta \leq 1$ , the iterates  $\{\mathbf{x}^{(k)}\}$  from DABA (Algorithm 1) converge to first-order critical points under Assumptions 1 to 4 in [37, App. B].*

*Proof.* Please refer to [37, App. C.3].  $\square$

Compared to DABA, decentralized methods [24], [25] have much stricter assumptions for provable convergence, making them sensitive to parameter tuning in practice. Even though Algorithm 1 requires  $\mathbf{x}^{\alpha(k+1)}$  to solve  $E^\alpha(\mathbf{x}^\alpha|\mathbf{x}^{(k)})$  or  $E^\alpha(\mathbf{x}^\alpha|\bar{\mathbf{x}}^{\alpha(k)})$  for convergence guarantees, DABA still empirically achieves good performances with approximate solutions, and thus, consumes fewer computational resources. Furthermore, as is shown in the next section, DABA is much faster to converge to more accurate solutions than the consensus

TABLE I: **Mean reprojection errors** with the **Trivial loss** and **Huber loss** on the datasets in Table II. Decentralized methods DR [24], ADMM [25], DABA (ours) are run for 1000 iterations with 4, 8, 16, 32 devices. Centralized methods Ceres [12] and DeepLM [19] are run for 40 iterations with single device as reference (DeepLM does not support Huber loss). On each dataset (row), any decentralized method with best result is **bold**, and outperforming Ceres and DeepLM is **red**. DABA (**ours**) achieves lowest reprojection error between decentralized methods and mostly outperforms centralized methods.

		Mean Reprojection Error with the Trivial Loss														
Dataset		Init	Ceres	DeepLM	4 Devices			8 Devices			16 Devices			32 Devices		
					DR	ADMM	DABA	DR	ADMM	DABA	DR	ADMM	DABA	DR	ADMM	DABA
BAL [1]	Ladybug	10.48	0.707	0.710	0.837	<b>0.698</b>	<b>0.690</b>	0.846	<b>0.703</b>	<b>0.690</b>	0.850	0.711	<b>0.690</b>	0.859	0.723	<b>0.690</b>
	Venice	26.33	0.468	0.466	0.515	0.516	<b>0.465</b>	0.515	0.525	<b>0.465</b>	0.516	0.532	<b>0.466</b>	0.517	0.544	<b>0.473</b>
	Final	12.57	0.855	0.848	2.017	0.876	<b>0.828</b>	2.042	0.903	<b>0.828</b>	2.053	0.908	<b>0.829</b>	2.123	0.921	<b>0.833</b>
IDSfM [2]	Gen. Markt	8.181	4.024	4.028	4.835	4.390	<b>3.977</b>	4.870	4.616	<b>3.986</b>	4.900	4.646	<b>3.988</b>	4.942	4.688	<b>4.017</b>
	Piccadilly	11.26	4.379	4.424	6.040	4.634	<b>4.218</b>	6.073	4.761	<b>4.217</b>	6.148	4.850	<b>4.249</b>	6.193	5.027	<b>4.312</b>
	R. Forum	6.407	1.211	1.177	2.194	1.381	<b>1.152</b>	2.211	1.446	<b>1.161</b>	2.232	1.516	<b>1.171</b>	2.251	1.585	<b>1.168</b>
	Trafalgar	10.77	3.702	3.886	5.065	3.994	<b>3.604</b>	5.263	4.111	<b>3.601</b>	5.169	4.185	<b>3.624</b>	5.193	4.395	<b>3.657</b>
	U. Square	10.56	3.992	3.977	5.376	4.233	<b>3.893</b>	5.426	4.401	<b>3.888</b>	5.485	4.519	<b>3.925</b>	5.538	4.619	<b>3.975</b>
	V. Cathedral	9.506	1.957	1.959	3.119	2.068	<b>1.777</b>	3.139	2.021	<b>1.770</b>	3.273	2.057	<b>1.831</b>	3.328	2.216	<b>1.834</b>

		Mean Reprojection Error with the Huber Loss														
Dataset		Init	Ceres	DeepLM	4 Devices			8 Devices			16 Devices			32 Devices		
					DR	ADMM	DABA	DR	ADMM	DABA	DR	ADMM	DABA	DR	ADMM	DABA
BAL [1]	Ladybug	3.267	0.704	-	0.834	<b>0.698</b>	<b>0.690</b>	0.841	<b>0.703</b>	<b>0.690</b>	0.844	0.712	<b>0.690</b>	0.848	0.723	<b>0.690</b>
	Venice	4.750	0.468	-	0.515	0.516	<b>0.465</b>	0.515	0.524	<b>0.465</b>	0.516	0.530	<b>0.465</b>	0.517	0.544	<b>0.473</b>
	Final	7.573	0.815	-	1.995	0.859	<b>0.796</b>	2.020	0.882	<b>0.795</b>	2.032	0.886	<b>0.796</b>	2.101	0.898	<b>0.815</b>
IDSfM [2]	Gen. Markt	8.031	3.942	-	4.777	4.320	<b>3.893</b>	4.812	4.525	<b>3.914</b>	4.842	4.561	<b>3.925</b>	4.886	4.598	<b>3.942</b>
	Piccadilly	10.34	4.136	-	5.739	4.334	<b>3.951</b>	5.776	4.441	<b>3.963</b>	5.851	4.564	<b>3.961</b>	5.892	4.689	<b>3.978</b>
	R. Forum	6.247	1.136	-	2.149	1.296	<b>1.120</b>	2.163	1.406	<b>1.127</b>	2.187	1.471	<b>1.133</b>	2.204	1.516	<b>1.133</b>
	Trafalgar	9.900	3.544	-	4.816	3.722	<b>3.363</b>	4.939	3.876	<b>3.377</b>	5.000	3.952	<b>3.401</b>	4.944	4.149	<b>3.438</b>
	U. Square	10.18	3.987	-	5.242	4.096	<b>3.755</b>	5.291	4.261	<b>3.763</b>	5.348	4.366	<b>3.763</b>	5.400	4.470	<b>3.818</b>
	V. Cathedral	9.085	1.748	-	3.008	1.928	<b>1.709</b>	3.027	1.923	<b>1.681</b>	3.156	1.956	<b>1.740</b>	3.206	2.088	<b>1.734</b>

TABLE II: Largest Bundle adjustment datasets of more than 700 cameras in BAL [1] and IDSfM [2].

Dataset		# Cameras	# Points	# Observations
BAL [1]	Ladybug	1723	156502	678718
	Venice	1778	993923	5001946
	Final	13682	4456117	28987644
IDSfM [2]	Gen. Markt	706	93672	364029
	Piccadilly	2289	209504	999878
	R. Forum	1063	265047	1292756
	Trafalgar	5032	388956	1826071
	U. Square	796	46066	230811
	V. Cathedral	836	265553	1333280

methods [24], [25], which suggests that DABA is preferable for large-scale decentralized bundle adjustment.

## VII. EVALUATION

We perform extensive benchmarks on the BAL [1] and IDSfM [2] datasets and compare our method DABA (Algorithm 1) with centralized (single device) methods Ceres [12] and DeepLM [19], and decentralized methods DR [24] and ADMM [25]. For all decentralized methods, the datasets are partitioned by evenly distributing measurements to each device if there are more than 700 cameras. Otherwise, the partitioning is determined according to the number of cameras per device. DR, ADMM and DABA are implemented in CUDA, Ceres in C++, and DeepLM in CUDA & Python. All experiments are conducted on a computer with 80 Intel Xeon 2.2GHz

CPU cores and 8 Nvidia V100 GPUs, and OpenMPI is used for inter-device communication. We compare all methods across metrics on accuracy, efficiency, memory usage and communication load.

### A. Accuracy

We evaluate the mean reprojection errors of all methods. The centralized methods are run for 40 iterations and decentralized methods for 1000 iterations to ensure sufficient accuracy. In the default solver options, Ceres and DeepLM have a maximum of 50 and 20 iterations, respectively. However, we found that these centralized methods usually exhibit limited improvement after 30 iterations, and the optimization time per iteration tends to increase significantly beyond this point. Thus, we run Ceres and DeepLM for 40 iterations to balance accuracy and time. We report in Table I the mean reprojection errors with the trivial loss and Huber loss on the nine largest datasets of more than 700 cameras in BAL & IDSfM (see Table II), while the results for the remaining datasets of less than 700 cameras are in [37, App. A.1]. The 3D reconstruction results of DABA (ours) with 8 devices and the Huber loss on some of these datasets are shown in Fig. 1.

As shown in Table I and [37, App. A.1], DABA yields the most accurate results compared to the other decentralized methods by a large margin across the board—all 20 datasets, both types of losses, and all numbers of devices. Additionally, DABA, while being fully decentralized, outperforms central-

TABLE III: The number of datasets out of All in BAL & 1DSfM (20 datasets) and Largest in Table II (9 datasets) where DABA is more accurate than Ceres [12] and DeepLM [19].

Datasets	Loss	4 Devices	8 Devices	16 Devices	32 Devices
All	Trivial	18/20	18/20	15/20	14/20
	Huber	19/20	19/20	16/20	17/20
Largest	Trivial	9/9	9/9	9/9	8/9
	Huber	9/9	9/9	9/9	8/9

ized methods Ceres and DeepLM on most of the datasets as summarized in Table III. Unsurprisingly, decentralized methods converge to lower accuracy as the number of devices increases, since more devices lead to more relaxed upper bounds for DABA and greater consensus errors for DR and ADMM. However, we also find that larger problems are less impacted by the increasing number of devices. This validates that decentralized methods like DABA are suitable for large-scale bundle adjustment.

DR, ADMM and DABA use Levenberg–Marquardt (LM) algorithm to solve subproblems, e.g., Eqs. (12) and (26). We found DR and ADMM need up to 20 inner LM steps to converge to acceptable accuracy, whereas DABA achieves better accuracy with only one successful inner LM step for all setups. This suggests that DABA is more robust to approximated solutions of subproblems and more time efficient due to fewer inner LM steps, which is evaluated in the next section.

### B. Efficiency

First, we evaluate the time speedup of DABA (ours) with 4 and 8 devices against centralized methods Ceres [12] and DeepLM [19] on the largest datasets of more than 700 cameras in Table II. Each device uses one GPU and we consider together both the computation and communication time. The time speedup of DABA for a problem  $p$  is  $\frac{T_{\Delta}(p)}{T_{\text{DABA}}(p)}$ , where  $T_{\text{DABA}}(p)$  is the time that DABA takes to reach the reference objective value  $F_{\text{ref}}$  while  $T_{\Delta}(p)$  is the time that Ceres or DeepLM takes to reach a target objective value  $F_{\Delta}(p) > F_{\text{ref}}$ :

$$F_{\Delta}(p) \triangleq F_{\text{ref}} + \Delta \cdot (F_{\text{init}} - F_{\text{ref}}) \quad (32)$$

where  $F_{\text{init}}$  is the initial objective value and  $\Delta \in (0, 1)$  is the suboptimality tolerance. Here, we choose  $F_{\text{ref}}$  to be the smallest objective value separately achieved by Ceres and DeepLM for 40 iterations and  $\Delta = 2.5 \times 10^{-4}$  since Ceres and DeepLM have a slow convergence around  $F_{\text{ref}}$ . In contrast, DABA has to exactly attain  $F_{\text{ref}} < F_{\Delta}(p)$ . This means when measuring time, our method is subjected to a stricter convergence criteria. Nevertheless, Table IV indicates that DABA is significantly faster in all setups, where DABA is 23.9~588.3x faster than Ceres and 2.6~174.6x faster than DeepLM for the trivial loss, and 69.7~953.7x faster than Ceres for the Huber loss. In summary, while DABA might take several hundred iterations to reach accuracies comparable to Ceres and DeepLM, our multi-GPU implementation makes DABA empirically far more time-efficient for large-scale bundle adjustment problems.

Next, we analyze the efficiency of decentralized methods in terms of numbers of iterations. In addition to DR [24], ADMM

TABLE IV: **Time speedup** of DABA with 4 and 8 devices over Ceres [12] and DeepLM [19] on datasets of more than 700 cameras. DABA **achieves speedups on all ranging from 23.9~953.7x over Ceres and 2.6~174.6x over DeepLM.**

Dataset		Time Speedup					
		Trivial Loss				Huber Loss	
		4 Devices		8 Devices		4 Devices	8 Devices
		Ceres	DeepLM	Ceres	DeepLM	Ceres	
BAL [1]	Ladybug	575.1	174.6	588.3	165.1	811.3	953.7
	Venice	23.9	2.6	43.9	4.8	89.2	148.4
	Final	185.0	22.7	258.2	32.8	173.2	250.6
1DSfM [2]	Gen. Markt	99.7	24.6	65.4	16.2	102.0	65.0
	Piccadilly	117.2	18.1	170.5	23.8	139.3	155.2
	R. Forum	138.3	11.0	181.8	11.2	69.7	97.1
	Trafalgar	148.2	29.7	239.6	41.7	174.1	228.0
	U. Square	59.2	9.7	49.9	8.6	162.3	117.1
	V. Cathedral	308.8	32.6	477.9	50.9	176.6	419.8

[25], DABA (ours), we also implement DUBA (Decentralized and Unaccelerated Bundle Adjustment) to ablate our method without acceleration and adaptive restart. The only difference between DUBA and DABA is that  $\mathbf{x}^{\alpha(k+1)}$  in DUBA is always yielded by Eq. (12), whereas  $\mathbf{x}^{\alpha(k+1)}$  in DABA results from Eq. (26) unless adaptive restart is triggered. Here, we compute the performance profiles [43] of all decentralized methods with respect to the number of iterations on all datasets. Given an optimizer and a problem set  $\mathcal{P}$ , the performance profile at iteration  $k$  refers to the percentage of problems  $p \in \mathcal{P}$  solved, i.e. the optimizer attains the target objective value  $F_{\Delta}(p)$  in Eq. (32) with the reference objective value  $F_{\text{ref}}$  from Ceres and suboptimality tolerance  $\Delta = 1 \times 10^{-4}$ . The performance profiles with the trivial loss and the Huber loss on 4, 8, 16, 32 devices are shown in Fig. 3. DABA solves more problems than other methods and takes fewer iterations to reach more accurate results. DABA is also much faster than DUBA, demonstrating that Nesterov’s acceleration and adaptive restart significantly improve performance.

### C. Memory & Communication

We evaluate the maximum memory usage per device and the total communication load per iteration for decentralized methods with 1, 2, 4, 8, 16, 32 devices on the four largest datasets in Table II, i.e., Venice, Final in BAL and Piccadilly, Trafalgar in 1DSfM. In Fig. 4(a) to 4(d), we observe that, as expected, the maximum memory usage per device decreases as the number of devices increases, enabling scaling to large bundle adjustment problems. Furthermore, DABA takes almost the same memory as the others while being more efficient and accurate. In Fig. 4(e) to 4(h), we observe that the total communication load generally increases with the number of devices, also expected, since more devices require more communication. Since ADMM communicates merely cameras while DABA exchanges points as well as cameras, DABA has more communication load than ADMM. Despite the larger communication load, the superior efficiency and accuracy of DABA compared to the other decentralized methods (see



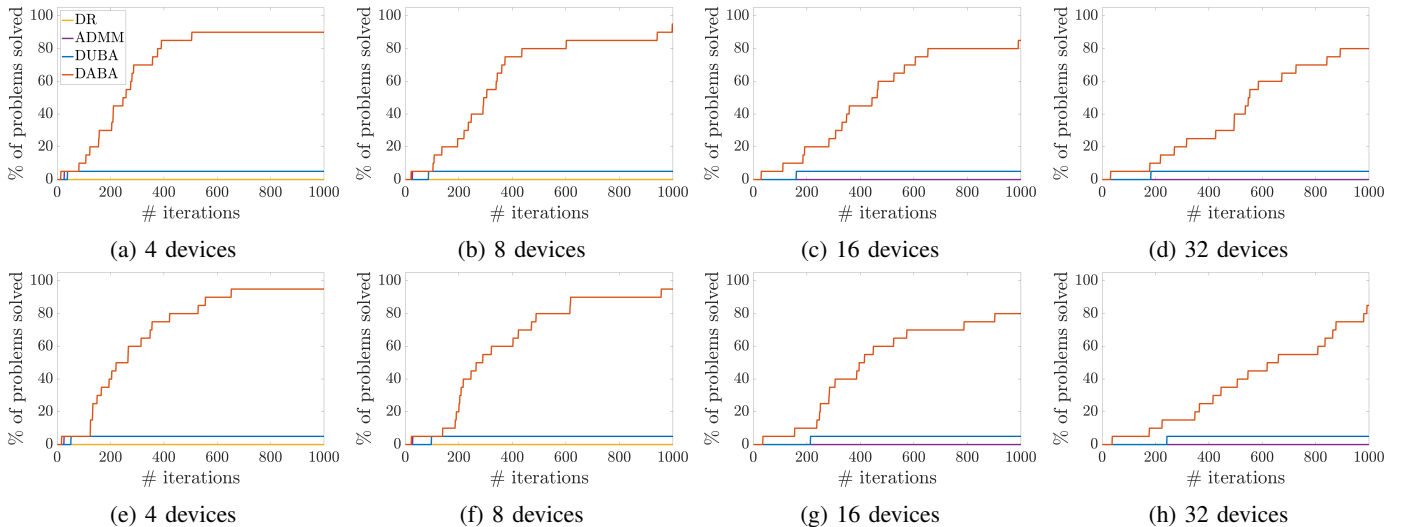


Fig. 3: **Performance profiles** with respect to the number of iterations for DABA (ours) compared to other decentralized methods on the 20 datasets in BAL and 1DSfM with (a)-(d) the trivial loss and (e)-(h) the Huber loss. Reference objective values  $F_{\text{ref}}$  is from Ceres [12] and suboptimality tolerance  $\Delta = 1 \times 10^{-4}$ .

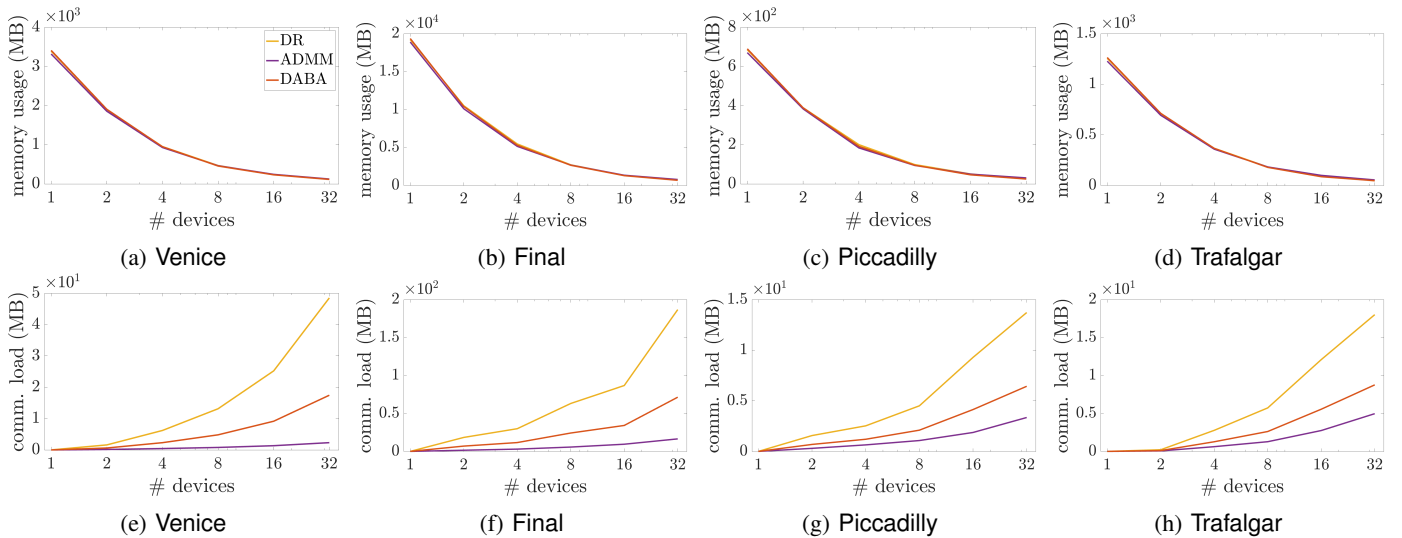


Fig. 4: **Max memory usage per device** (a)-(d) and **total communication load per iteration** (e)-(h) for DABA (ours) compared to decentralized methods on four largest datasets, Venice and Final in BAL, and Piccadilly and Trafalgar in 1DSfM.

Secs. VII-A and VII-B) makes it desirable for arbitrarily large-scale bundle adjustment.

### VIII. CONCLUSION

We have presented DABA, a decentralized and accelerated method for large-scale bundle adjustment with provable convergence to first-order critical points. The key insight of our method is to decouple optimization variables and reduce bundle adjustment to independent subproblems with majorization minimization. This is achieved through the complete analysis of a novel reprojection error. We also implement Nesterov’s acceleration for empirical speedup and adaptive restart for theoretical guarantees while maintaining decentralization. Compared to decentralized baselines [23]–[26], our method has less strict assumptions for provable convergence, no need for specific parameter tuning, and more robustness to approxi-

mate solutions of subproblems. On extensive benchmarks with public datasets, our method has similar memory and communication overhead but outperforms decentralized baselines [24], [25] with a large margin in terms of accuracy and efficiency. As a result of multi-GPU implementation, our method, albeit decentralized, is more accurate on large-scale datasets with a speedup of up to 953.7x and 174.6x over centralized baselines Ceres [12] and DeepLM [19], respectively. In the future, we are planning to theoretically relax the local minimum conditions of  $\mathbf{x}^{\alpha(k+1)}$  in Eqs. (12) and (26); extend our method for bundle adjustment with lines and planes; and implement our method on multi-robot large-scale 3D reconstruction.

### ACKNOWLEDGMENTS

TM was supported by the National Science Foundation under award 1837515.



## REFERENCES

- [1] S. Agarwal, N. Snavely, S. M. Seitz, and R. Szeliski, "Bundle adjustment in the large," in *European conference on computer vision*. Springer, 2010, pp. 29–42.
- [2] K. Wilson and N. Snavely, "Robust global translations with ldsfm," in *European conference on computer vision*. Springer, 2014, pp. 61–75.
- [3] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.
- [4] D. M. Rosen, K. J. Doherty, A. Terán Espinoza, and J. J. Leonard, "Advances in inference and representation for simultaneous localization and mapping," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, pp. 215–242, 2021.
- [5] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT press, 2005.
- [6] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski, "Building Rome in a Day," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2009.
- [7] Y. Jeong, D. Nister, D. Steedly, R. Szeliski, and I. Kweon, "Pushing the envelope of modern methods for bundle adjustment," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.
- [8] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The kitti vision benchmark suite," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3354–3361.
- [9] S. Song, M. Chandraker, and C. C. Guest, "High accuracy monocular sfm and scale correction for autonomous driving," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 4, pp. 730–743, 2015.
- [10] H. Liu, G. Zhang, and H. Bao, "Robust keyframe-based monocular slam for augmented reality," in *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2016, pp. 1–10.
- [11] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—a modern synthesis," in *International workshop on vision algorithms*. Springer, 1999, pp. 298–372.
- [12] S. Agarwal, K. Mierle, and Others, "Ceres solver," <http://ceres-solver.org>.
- [13] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g<sup>2</sup>o: A general framework for graph optimization," in *2011 IEEE International Conference on Robotics and Automation*.
- [14] F. Dellaert, "Factor graphs and GTSAM," Georgia Institute of Technology, Tech. Rep. GT-RIM-CP&R-2012-002, 2012.
- [15] L. Pineda, T. Fan, M. Monge, S. Venkataraman, P. Sodhi, R. T. Chen, J. Ortiz, D. DeTone, A. Wang, S. Anderson, J. Dong, B. Amos, and M. Mukadam, "Theseus: A Library for Differentiable Nonlinear Optimization," *Advances in Neural Information Processing Systems*, 2022.
- [16] H. Martiros, A. Miller, N. Bucki, B. Solliday, R. Kennedy, J. Zhu, T. Dang, D. Pattison, H. Zheng, T. Tomic *et al.*, "Symforce: Symbolic computation and code generation for robotics," in *Robotics: Science and Systems (RSS)*, 2022.
- [17] K. Ni, D. Steedly, and F. Dellaert, "Out-of-core bundle adjustment for large-scale 3d reconstruction," in *2007 IEEE 11th International Conference on Computer Vision*. IEEE, 2007, pp. 1–8.
- [18] L. Zhou, Z. Luo, M. Zhen, T. Shen, S. Li, Z. Huang, T. Fang, and L. Quan, "Stochastic bundle adjustment for efficient and scalable 3d reconstruction," in *European Conference on Computer Vision*. Springer, 2020, pp. 364–379.
- [19] J. Huang, S. Huang, and M. Sun, "Deeplm: Large-scale nonlinear least squares on deep learning frameworks using stochastic domain decomposition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 308–10 317.
- [20] C. Wu, S. Agarwal, B. Curless, and S. M. Seitz, "Multicore Bundle Adjustment," in *CVPR*, 2011.
- [21] J. Ren, W. Liang, R. Yan, L. Mai, S. Liu, and X. Liu, "Megba: A gpu-based distributed library for large-scale bundle adjustment," in *European Conference on Computer Vision*, 2022.
- [22] Y. Tian, A. S. Bedi, A. Koppel, M. Calvo-Fullana, D. M. Rosen, and J. P. How, "Distributed riemannian optimization with lazy communication for collaborative geometric estimation," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 4391–4398.
- [23] J. Ortiz, M. Pupilli, S. Leutenegger, and A. J. Davison, "Bundle adjustment on a graph processor," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2416–2425.
- [24] A. Eriksson, J. Bastian, T.-J. Chin, and M. Isaksson, "A consensus-based framework for distributed bundle adjustment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [25] R. Zhang, S. Zhu, T. Fang, and L. Quan, "Distributed very large scale bundle adjustment by global camera consensus," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct 2017.
- [26] N. Demmel, M. Gao, E. Laude, T. Wu, and D. Cremers, "Distributed photometric bundle adjustment," in *2020 International Conference on 3D Vision (3DV)*. IEEE, 2020, pp. 140–149.
- [27] Y. Nesterov, "A method for unconstrained convex minimization problem with the rate of convergence  $O(1/k^2)$ ," in *Doklady AN USSR*, vol. 269, 1983, pp. 543–547.
- [28] —, *Introductory lectures on convex optimization: A basic course*. Springer Science & Business Media, 2013, vol. 87.
- [29] B. O'donoghue and E. Candes, "Adaptive restart for accelerated gradient schemes," *Foundations of computational mathematics*, vol. 15, no. 3, pp. 715–732, 2015.
- [30] T. Fan and T. Murphey, "Majorization minimization methods for distributed pose graph optimization," *arXiv preprint arXiv:2108.00083*, 2021.
- [31] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [32] M. Bujnak, Z. Kukelova, and T. Pajdla, "New efficient solution to the absolute pose problem for camera with unknown focal length and radial distortion," in *Asian Conference on Computer Vision (ACCV)*, 2010, pp. 11–24.
- [33] K. Josephson and M. Byrod, "Pose estimation with radial distortion and unknown focal length," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2009, pp. 2419–2426.
- [34] Z. Kukelova, M. Bujnak, and T. Pajdla, "Real-time solution to the absolute pose problem with unknown radial distortion and focal length," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2013, pp. 2816–2823.
- [35] D. R. Hunter and K. Lange, "A tutorial on MM algorithms," *The American Statistician*, vol. 58, no. 1, pp. 30–37, 2004.
- [36] Y. Sun, P. Babu, and D. P. Palomar, "Majorization-minimization algorithms in signal processing, communications, and machine learning," *IEEE Transactions on Signal Processing*, vol. 65, no. 3, pp. 794–816, 2016.
- [37] T. Fan, J. Ortiz, M. Hsiao, M. Monge, J. Dong, T. Murphey, and M. Mukadam, "Decentralization and acceleration enables large-scale bundle adjustment," *arXiv:2305.07026*, 2023.
- [38] H. Li and Z. Lin, "Accelerated proximal gradient methods for nonconvex programming," in *Advances in neural information processing systems*, 2015, pp. 379–387.
- [39] T. Fan and T. Murphey, "Generalized proximal methods for pose graph optimization," in *International Symposium on Robotics Research (ISRR)*, 2019.
- [40] —, "Majorization minimization methods for distributed pose graph optimization with convergence guarantees," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020.
- [41] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 4, pp. 376–380, 1991.
- [42] J. Levinson, C. Esteves, K. Chen, N. Snavely, A. Kanazawa, A. Ros-tamizadeh, and A. Makadia, "An analysis of svd for deep rotation estimation," *Advances in Neural Information Processing Systems*, vol. 33, pp. 22 554–22 565, 2020.
- [43] E. D. Dolan and J. J. Moré, "Benchmarking optimization software with performance profiles," *Mathematical Programming*, vol. 91, no. 2, pp. 201–213, 2002.