2024 I-RIM Conference
October 25-27, Rome, Italy
ISBN: 9788894580556
10.5281/zenodo.14731057

Structureless Bundle Adjustment for Robotics

Lorenzo De Rebotti, Giorgio Grisetti
Dept. of Computer, Control, and Management Engineering "Antonio Ruberti", Sapienza University of Rome, Italy
derebotti@diag.uniroma1.it - grisetti@diag.uniroma1.it

***Abstract*—Global Epipolar Adjustment (GEA) is an alternative formulation for solving the Bundle Adjustment (BA) problem without explicitly considering the points in the map, hence it is *structureless*. Albeit computationally interesting, this formulation has not found extensive use in robotics applications. In this paper we experimentally analyze the advantages and the shortcomings of GEA and BA. The goal of this work is to characterize the situations when the use of one might be convenient over the other. We made available an open source C++ implementation of all approaches at the time of writing.** [1]

I. Introduction

Bundle Adjustment (BA) [1] is a well-known problem in computer vision and robotics, and is used as a building block of many Structure from Motion (SfM) [2] and Simultaneous Localization and Mapping (SLAM) [8] systems. Simply put, the task of BA is to compute the positions of a set of 3D points in the scene *and* the poses of a set of cameras that better explain the *measured* projections of the points on the images captured by the cameras. The common way to solve BA is by minimizing the robustified sum of the reprojection errors of all point observations through an Iterative Reweighted Least-Squares (IRLS) schema. This approach explicitly estimates the pose of all cameras and the position of all points. Given the potentially high number of variables, this procedure is computationally challenging.

Global Epipolar Adjustment (GEA) [6], [9] is an alternative formulation for solving the poses of the cameras that allows to *never* explicitly computing the poses of the points. To achieve this, GEA minimizes the epipolar constraint instead of the reprojection error. Thanks to the peculiar structure of the GEA error terms, the implementation can be significantly accelerated. Still, in the absence of outliers and errors GEA and BA have the same global minimum for the camera poses. BA is used as a building block of Visual Odometry (VO) and SLAM to refine the local/global map estimates and reduce their drift. In this paper, we describe how these two formulations are correlated and how they can be both solved by IRLS. Furthermore, we perform a comparative analysis of the two approaches in a representative set of situations.

II. Bundle Adjustment

Consider a set of cameras located at poses $\mathbf{X}_{1:N}$ observing a set of points $\mathbf{x}_{1:M}$ in the environment. Let $\mathbf{z}_{nm}$ be the measured projection of the point $\mathbf{x}_m$ onto the camera $\mathbf{X}_n$.

This work has been supported by PNRR MUR project PE0000013-FAIR.

[1] https://gitlab.com/srrg-software/srrg2_solver/-/tree/epipolar_ba/srrg2_solver/app/sfm

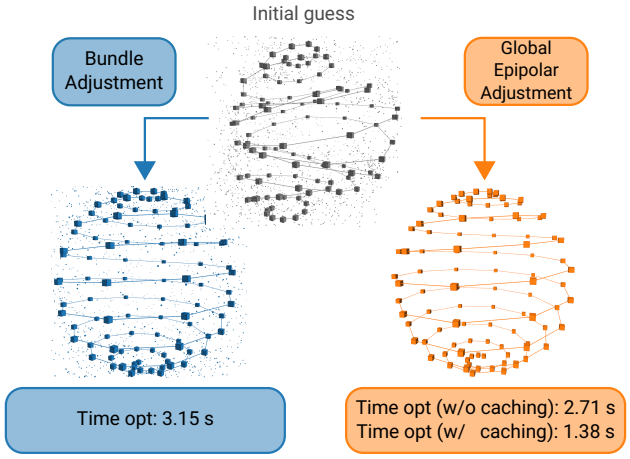

Fig. 1: BA problem consisting of 1814 points, observed by 100 stereo cameras totaling 72271 measurements. Top: initial configuration. Bottom left: BA optimization result, it took 3.15 seconds. Bottom right: GEA optimization result, it took 2.71 seconds in a naive implementation, and 1.38 s with algorithmic enhancements.

BA seeks to estimate both camera poses and point positions based on the known set $\{\mathbf{z}_{nm}\}$. In the remainder of this work, we represent a camera pose $\mathbf{X} = [\mathbf{R}\ \mathbf{t}]$ as a homogeneous transformation matrix consisting of a 3D rotation matrix $\mathbf{R}$, and a translation vector $\mathbf{t}$, where we omit the last constant row $[\mathbf{0}\ 1]$ for compactness. For the same reason, we refer to the transformation of a 3D point $\mathbf{p}$ by $\mathbf{X}$ with product $\mathbf{Xp} \triangleq \mathbf{Rp}+\mathbf{t}$, where we implicitly assume that the point can be converted to homogeneous coordinates and converted back after the operation.

For an ideal pinhole camera, the re-projection error is the difference between the image coordinates that would result from imaging the point $\mathbf{x}_m$ from a camera located at $\mathbf{X}_n$ and the actual measurement $\mathbf{z}_{nm}$, as follows:

$$\mathbf{e}^{\text{ba}}(\mathbf{X}_n, \mathbf{x}_m) \triangleq \pi\left(\mathbf{K}\mathbf{X}_n^{-1}\mathbf{x}_m\right) - \mathbf{z}_{nm}. \tag{1}$$

Here $\mathbf{K}$ is the known camera matrix, and $\pi(\cdot) : \Re^3 \rightarrow \Re^2$ is the homogeneous division so that $\pi([x\ y\ z]^\top) = [\frac{x}{z}, \frac{y}{z}]^\top$.

A common formulation of BA involves the minimization of *all* measurement errors with respect to the camera poses and point positions, as follows:

$$\mathcal{X}^* = \underset{\mathcal{X}}{\operatorname{argmin}} \sum_{k=1}^{K} \rho \|\mathbf{e}_k^{\text{ba}}(\mathbf{X}_{n(k)}, \mathbf{x}_{m(k)})\|_{\mathbf{\Omega}_k}. \tag{2}$$

Here $\mathcal{X} = \langle \mathbf{X}_{1:N}, \mathbf{x}_{1:M} \rangle$ is the set of all variables being estimated, and k is an index enumerating all measurements $\{\mathbf{z}_{nm}\}$. Accordingly, $n(k)$ and $m(k)$ are selector functions denoting respectively the camera pose index and the point index corresponding to the k^{th} measurement, while $\mathbf{\Omega}_k$ is an information matrix expressing the confidence of the measurement. Finally, $\rho(\cdot)$ is a robust cost function used to lessen the effect of outliers, usually characterized by large errors.

A common way to represent the problem expressed in Eq. (2) is through a factor graph [4]. A factor graph for BA would consist of $N+M$ variable nodes, one for each camera pose and one for each point position, and of K factor nodes corresponding to the error terms $\mathbf{e}_k^{\text{ba}}$. Each factor node $\mathbf{e}_k^{\text{ba}}$ is connected to the pair of variables $\mathbf{X}_{n(k)}$ and $\mathbf{x}_{m(k)}$ from which it depends.

Popular factor graph solvers implement IRLS to solve Eq. (2). IRLS seeks to iteratively refine an existing solution by computing a perturbation $\mathbf{\Delta x}$ that minimizes a local quadratic approximation of the problem. In the case of BA, $\mathbf{\Delta x}$ has dimension $6N+3M$, since each camera pose contributes with 6 degrees of freedom, and each point has 3. For completeness, we report in Alg. 1 a schematic version of one iteration of IRLS. According to [5], we used the $\boxplus$ notation since we carry on the linearization of the error functions in Line 9 with respect to a local Euclidean perturbation around the current estimate $\breve{\mathcal{X}}$. Let $\mathbf{X}_i$ be the i^{th} variable in the pool, and let $\mathbf{\Delta x}_i$ be the corresponding block in the perturbation vector, if $\mathbf{X}_i$ is a camera pose $\mathbf{X}_i \boxplus \mathbf{\Delta x}_i = \exp(\mathbf{\Delta x}_i)\mathbf{X}_i$, while if $\mathbf{x}_i$ is a point position $\mathbf{x}_i \boxplus \mathbf{\Delta x}_i = \mathbf{x}_i + \mathbf{\Delta x}_i$. With $\exp(\mathbf{\Delta x}_i)$ we denote the transformation matrix computed from the 6D perturbation $\mathbf{\Delta x}_i$, where the zero perturbation maps the identity transform: $\exp(\mathbf{0}) = \mathbf{I}$.

Remarkably, in BA the error $\mathbf{e}_k$ depends only on a pair of variables: the camera $\mathbf{X}_{n(k)}$ and the point $\mathbf{x}_{m(k)}$. Hence, the Jacobian in line 9 will be mostly empty except in the 2×6 block corresponding to $\mathbf{\Delta x}_{n(k)}$ and in the 2×3 block of $\mathbf{\Delta x}_{m(k)}$. This leads to a sparse structure of the $\mathbf{H}$ matrix computed in line 11, where each measurement contributes to the block diagonal, and to the off-diagonal block at the intersection of the observed point and the observing camera.

In the next section, we investigate an alternative formulation of the problem, firstly presented in [9], that completely neglects estimating the point variables by leveraging on the epipolar constraint between pairs of views.

Algorithm 1 One iteration of IRLS

Require: Initial guess $\breve{\mathcal{X}}$; Measurements $\mathcal{C} = \{\langle \mathbf{Z}_k, \mathbf{\Omega}_k \rangle\}$
Ensure: Improved solution $\mathcal{X}^\star$
1: $\mathbf{b} \leftarrow \mathbf{0}$
2: $\mathbf{H} \leftarrow \mathbf{0}$
3: **for all** $k \in \{1 \dots K\}$ **do**
4: $\quad \mathbf{e}_k \leftarrow \mathbf{e}_k(\mathcal{X})$
5: $\quad \chi_k \leftarrow \mathbf{e}_k^T \mathbf{\Omega}_k \mathbf{e}_k$
6: $\quad u_k \leftarrow \sqrt{\chi_k}$
7: $\quad \gamma_k = \frac{1}{u_k} \left. \frac{\partial \rho_k(u)}{\partial u} \right|_{u=u_k}$.
8: $\quad \tilde{\mathbf{\Omega}}_k = \gamma_k \mathbf{\Omega}_k$
9: $\quad \mathbf{J}_k \leftarrow \left. \frac{\partial \mathbf{e}_k(\mathcal{X} \boxplus \mathbf{\Delta x})}{\partial \mathbf{\Delta x}} \right|_{\mathbf{\Delta x}=\mathbf{0}}$
10: $\quad \mathbf{b} \leftarrow \mathbf{b} + \mathbf{J}_k^\top \tilde{\mathbf{\Omega}}_k \mathbf{e}_k$
11: $\quad \mathbf{H} \leftarrow \mathbf{H} + \mathbf{J}_k^\top \tilde{\mathbf{\Omega}}_k \mathbf{J}_k$
12: $\mathbf{\Delta x} \leftarrow \text{solve}(\mathbf{H \Delta x} = -\mathbf{b})$
13: **return** $\breve{\mathcal{X}} \boxplus \mathbf{\Delta x}$

III. Global Epipolar Adjustment

The essential matrix $\mathbf{E}$ characterizes the relation between two cameras observing the same point in the world from two different poses $\mathbf{X}$ and $\mathbf{X}'$ through the following constraint

$$\bar{\mathbf{z}}^\top \mathbf{E} \bar{\mathbf{z}}' = 0. \tag{3}$$

Here $\bar{\mathbf{z}} = \mathbf{K}^{-1}\mathbf{z}$ is a measurement in camera coordinates, hence $\bar{\mathbf{z}}$ and $\bar{\mathbf{z}}'$ represent the direction of a ray imaging the point in the first and the second cameras respectively.

The essential matrix is correlated to the relative position $\mathbf{\Delta X} = \mathbf{X}^{-1}\mathbf{X}'$ of the two cameras based on the following relation:

$$\mathbf{E} = \mathbf{\Delta R}^\top \lfloor \mathbf{\Delta t} \rfloor_\times, \tag{4}$$

however to prevent degeneration for $\mathbf{t} = \mathbf{0}$, we employ a normalized version of the essential matrix

$$\bar{\mathbf{E}} = \frac{\mathbf{E}}{\|\mathbf{\Delta t}\|}. \tag{5}$$

Since the relative translation depends on the poses $\mathbf{X}$ and $\mathbf{X}'$, we can say that the normalized essential matrix is a function of the two poses, and we turn the constraint of Eq. (4) to a monodimensional error factor:

$$\mathbf{e}^{\text{gea}}(\mathbf{X}, \mathbf{X}') = \bar{\mathbf{z}}^\top \bar{\mathbf{E}}(\mathbf{X}, \mathbf{X}') \bar{\mathbf{z}}'. \tag{6}$$

In GEA we are interested in finding the camera poses that minimize the norm of errors in Eq. (6), as follows:

$$\mathbf{X}^* = \underset{\mathbf{X}}{\text{argmin}} \sum_{n=1}^{N} \sum_{(m,m') \subset \mathcal{C}(n)} \rho(\|\mathbf{e}_k^{\text{gea}}(\mathbf{X}_m, \mathbf{X}_{m'})\|). \tag{7}$$

For each point $\mathbf{x}_n$ we consider the set of all cameras $\mathcal{C}(n)$ that observed the point. From this set we pick pairs of distinct cameras (m, m'), and we evaluate the error $\mathbf{e}^{\text{gea}}$. For each pair of cameras, we will have as many factors as the number of points that are visible from both. The variables in this new estimation problem, however do not include the points.

Simplistic algebraic manipulations of Eq. (6) lead us to the express the epipolar error as:

$$\begin{aligned}\mathbf{e}^{\text{gea}}(\mathbf{X},\mathbf{X}') &= \bar{\mathbf{z}}^\top \bar{\mathbf{E}}(\mathbf{X},\mathbf{X}')\bar{\mathbf{z}}' \\ &= \bar{\mathbf{z}}^\top \underbrace{\begin{bmatrix} \bar{\mathbf{z}}'^\top & & \\ & \bar{\mathbf{z}}'^\top & \\ & & \bar{\mathbf{z}}'^\top \end{bmatrix}}_{\mathbf{u}^\top} \underbrace{\begin{bmatrix} \bar{\mathbf{E}}(\mathbf{X},\mathbf{X}')^\top_{1:*} \\ \bar{\mathbf{E}}(\mathbf{X},\mathbf{X}')^\top_{2:*} \\ \bar{\mathbf{E}}(\mathbf{X},\mathbf{X}')^\top_{3:*} \end{bmatrix}}_{\mathbf{v}(\mathbf{X},\mathbf{X}')} \\ &= \mathbf{u}^\top \mathbf{v}(\mathbf{X},\mathbf{X}').\end{aligned} \tag{8}$$

In Eq. (8) we expressed the same formula of Eq. (6) as the dot product of two 9D vectors: $\mathbf{u}$, and $\mathbf{v}$. The first vector $\mathbf{u}$ contains the components of the outer product between the measurements $\bar{\mathbf{z}}$ and $\bar{\mathbf{z}}'$. The second term $\mathbf{v}$ contains the rows of the essential matrix and is a function of the camera poses $\mathbf{X}$ and $\mathbf{X}'$. This expression paves the road for further optimization in implementing Alg. 1 for this case.

At first, we notice that the generic Jacobian $\mathbf{J}$ can be computed as

$$\begin{aligned}\mathbf{J}^{\text{gea}}(\mathbf{X},\mathbf{X}') &= \begin{bmatrix} \frac{\partial \mathbf{e}^{\text{gea}}(\mathbf{X}\boxplus\boldsymbol{\Delta}\mathbf{x},\mathbf{X}')}{\partial \boldsymbol{\Delta}\mathbf{x}} & \frac{\partial \mathbf{e}^{\text{gea}}(\mathbf{X},\mathbf{X}'\boxplus\boldsymbol{\Delta}\mathbf{x}')}{\partial \boldsymbol{\Delta}\mathbf{x}'} \end{bmatrix} \\ &= \mathbf{u}^\top \begin{bmatrix} \underbrace{\frac{\partial \mathbf{v}\left(\mathbf{X}\boxplus\boldsymbol{\Delta}\mathbf{x},\mathbf{X}'\right)}{\partial \boldsymbol{\Delta}\mathbf{x}}}_{\mathbf{J}_\mathbf{v}} & \underbrace{\frac{\partial \mathbf{v}\left(\mathbf{X},\mathbf{X}'\boxplus\boldsymbol{\Delta}\mathbf{x}'\right)}{\partial \boldsymbol{\Delta}\mathbf{x}'}}_{\mathbf{J}'_\mathbf{v}} \end{bmatrix} \\ &= \mathbf{u}^\top \begin{bmatrix} \mathbf{J}_\mathbf{v} & \mathbf{J}'_\mathbf{v} \end{bmatrix}.\end{aligned} \tag{9}$$

Here $\mathbf{J}_\mathbf{v}$ and $\mathbf{J}'_\mathbf{v}$ are 9×6 Jacobian matrices of the essential vector $\mathbf{v}$, with respect to the perturbation of the first and second camera poses $\boldsymbol{\Delta}\mathbf{x}$ and $\boldsymbol{\Delta}\mathbf{x}'$, and they do not depend on the measurements encoded in the vector $\mathbf{u}$. For the sake of readability in Eq. (8) and Eq. (9) we omitted the indices.

Given two cameras m and m', in Eq. (7) we have an addend for each point imaged by both. This term depends on the measurements $\bar{\mathbf{z}}$ and $\bar{\mathbf{z}}'$. For the generic pair of cameras m and m', we can then define a set of measurement pairs $\mathcal{Z}(m,m') = \{\langle \bar{\mathbf{z}}_i, \bar{\mathbf{z}}'_i \rangle\}$ spanning all mutually observed points.

So, we can compactly express both the Jacobian matrices and the error of the generic measurement between a pair of cameras m and m' as:

$$\mathbf{e}_i^{\text{gea}}(m,m') = \mathbf{u}_i^T \mathbf{v}(m,m') \tag{10}$$

$$\mathbf{J}_i(m,m') = \mathbf{u}_i^T \mathbf{J}_\mathbf{v}(m,m'). \tag{11}$$

Within one iteration of Alg. 1 both $\mathbf{v}(m,m')$ and $\mathbf{J}_\mathbf{v}(m,m')$ do not change for a fixed pair of cameras. Hence, the computation requires just evaluating a dot product for the error $\mathbf{e}_i^{\text{gea}}$ and a product between a row vector and two 9×6 matrices for the Jacobian $\mathbf{J}_i$. If one chooses the $L2$ norm as $\rho(\cdot)$, Alg. 1 degenerates to a classical Gauss-Newton (GN) schema, and in this case one can carry on further optimizations by *precomputing* constant terms that encapsulate *all* pairwise measurements.

Given a pair of cameras m and m', the cumulative errors computed in lines 5, 10 and line 11 become:

$$\begin{aligned}\chi_{m,m'} &= \sum_{\langle \mathbf{z}_i, \mathbf{z}'_i \rangle \in \mathcal{Z}(m,m')} \mathbf{v}^\top(m,m')\mathbf{u}_i\mathbf{u}_i^\top\mathbf{v}(m,m') \\ \mathbf{b}_{m,m'} &= \sum_{\langle \mathbf{z}_i, \mathbf{z}'_i \rangle \in \mathcal{Z}(m,m')} \mathbf{J}_\mathbf{v}^\top(m,m')\mathbf{u}_i\mathbf{u}_i^\top\mathbf{v}(m,m') \\ \mathbf{H}_{m,m'} &= \sum_{\langle \mathbf{z}_i, \mathbf{z}'_i \rangle \in \mathcal{Z}(m,m')} \mathbf{J}_\mathbf{v}^\top(m,m')\mathbf{u}_i\mathbf{u}_i^\top\mathbf{J}_\mathbf{v}^(m,m')\end{aligned} \tag{12}$$

Bringing in the summations, and defining per camera pair 9×9 constant measurement matrix $\mathbf{U}_{m,m'} = \sum_i \mathbf{u}_i\mathbf{u}_i^\top$, allows to simplify the above as follows:

$$\begin{aligned}\chi_{m,m'} &= \mathbf{v}^\top(m,m')\mathbf{U}_{m,m'}\mathbf{v}(m,m') \\ \mathbf{b}_{m,m'} &= \mathbf{J}_\mathbf{v}^\top(m,m')\mathbf{U}_{m,m'}\mathbf{v}(m,m') \\ \mathbf{H}_{m,m'} &= \mathbf{J}_\mathbf{v}^\top(m,m')\mathbf{U}_{m,m'}\mathbf{J}_\mathbf{v}(m,m')\end{aligned} \tag{13}$$

$\mathbf{U}_{m,m'}$ depends only on the measurements of shared observations between the camera pair and can be precomputed, while the Jacobians $\mathbf{J}_\mathbf{v}$ and the vectorized essential matrix $\mathbf{v}$ remain unchanged within one IRLS iteration. By leveraging these aspects, we lead to an algorithm that can construct the linear system in a time proportional to the number of camera pairs that share a common observation, regardless of the number of points in the scene.

IV. Experiments

To experimentally compare BA and GEA we used synthetic datasets representing bundling problems. We employed three different trajectories: Sphere, Torus, and Colosseum. The first two emulate the motion around the corresponding geometric figure, while the third one is extracted from real data [3]. For each trajectory, we emulated two sensor settings: monocular and stereo. The characteristics of the datasets are summarized in Tab. I. To render our data statistically representative, we added five different noise realizations to the same initial guess. We implemented both systems within the `srrg2_solver` [4]. In the results, we report mean and standard deviation of Absolute Trajectory Error (ATE) RMSE, computed with `evo`[2], and runtime. We ran the experiments with an i7-7700k CPU (4 cores @4.50 GHz), using the Levenberg-Marquardt (LM) algorithm on a single-core implementation.

In Tab. II it's possible to see that BA has a lower error in both sphere and torus trajectories. On Colosseum with mono setup GEA is slightly more accurate. This can be explained by the absence of a loop closure that prevents all systems from recovering the drift. The high variance of the results with the stereo setup shows the sensitivity to the initial guess in the alternative formulation.

GEA with caching Eq. (13) is the fastest approach, see Fig. 2, however it prevents the use of robustifiers. This is not an issue if we are highly confident about the data association

[2] https://github.com/MichaelGrupp/evo

TABLE I: Characteristics of the datasets, where N is the number of poses, M the number of landmarks generated, F_{ba} the number of BA factors, and F_{gea} the number of pairwise GEA factors.

		N	M	F_{ba}	F_{gea}
Mono	Coloss.	1000	10480	823773	97849
	Sphere	100	1813	36063	2557
	Torus	200	3869	144522	12804
Stereo	Coloss.	1000	7279	2450792	765445
	Sphere	100	1814	72271	10232
	Torus	200	5400	400009	53158

TABLE II: ATE RMSE[m] results of the two factor formulations with synthetic trajectory - mean and standard deviation.

			Init Guess	**BA**	**GEA**
$\text{ATE}_{pos}\,[m]$	**Mono**	Colosseum	0.970 ± 0.450	0.018 ± 0.008	$\mathbf{0.009 \pm 0.002}$
		Sphere	3.512 ± 0.455	$\mathbf{0.007 \pm 2\text{e}{-4}}$	0.041 ± 0.011
		Torus	8.359 ± 1.653	$\mathbf{0.011 \pm 2\text{e}{-4}}$	$0.027 \pm 9\text{e}-4$
	Stereo	Colosseum	2.201 ± 1.090	0.550 ± 0.783	1.521 ± 1.080
		Sphere	8.516 ± 2.821	$\mathbf{0.013 \pm 0.004}$	0.277 ± 0.127
		Torus	24.918 ± 4.854	$\mathbf{0.064 \pm 0.032}$	0.433 ± 0.499
$\text{ATE}_{rot}\,[rad]$	**Mono**	Colosseum	0.016 ± 0.004	$3\text{e}{-4} \pm 9\text{e}{-5}$	$\mathbf{2\text{e}{-4} \pm 5\text{e}{-5}}$
		Sphere	0.507 ± 0.026	$\mathbf{0.001 \pm 6\text{e}{-5}}$	$0.004 \pm 4\text{e}{-4}$
		Torus	0.621 ± 0.166	$\mathbf{0.001 \pm 2\text{e}{-5}}$	$0.002 \pm 9\text{e}-5$
	Stereo	Colosseum	0.025 ± 0.009	0.005 ± 0.007	0.014 ± 0.008
		Sphere	0.764 ± 0.258	$\mathbf{0.001 \pm 2\text{e}{-4}}$	0.020 ± 0.008
		Torus	0.960 ± 0.303	$\mathbf{0.002 \pm 0.001}$	0.051 ± 0.085

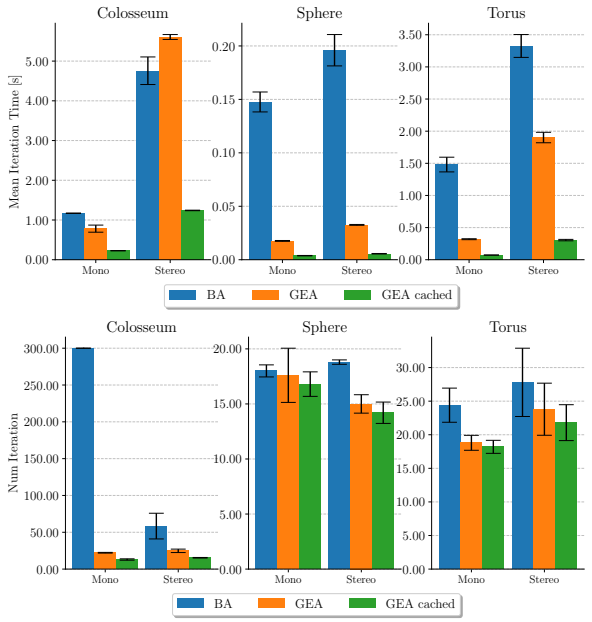

Fig. 2: Runtime analysis of the BA, GEA, and GEA cached factors for the three synthetic trajectories. Above: mean time required by each iteration. Below: mean number of iterations to converge to solution.

or the number of outliers is irrelevant. To analyze the effect of outliers we inject 5% of wrong measurements in a subsequent experiment, by randomly flipping the measurement indices in the optimization problem. We chose the *Cauchy* robust function [7]. The results reported in Tab. III confirm our

TABLE III: ATE RMSE[m] results of the two factor formulations with and without robustifier in presence of outliers - mean and standard deviation.

			Init Guess	**GEA w/o rob.**	**GEA w/ rob.**	**BA w/ rob.**
$\text{ATE}_{pos}\,[m]$	**Mono**	Coloss.	0.970 ± 0.450	0.560 ± 0.112	0.313 ± 0.150	$\mathbf{0.100 \pm 0.051}$
		Sphere	3.512 ± 0.455	0.241 ± 0.076	0.143 ± 0.024	$\mathbf{0.008 \pm 6\text{e}{-4}}$
		Torus	8.359 ± 1.653	0.055 ± 0.003	$0.031 \pm 8\text{e}{-4}$	$\mathbf{0.012 \pm 5\text{e}{-4}}$
	Stereo	Coloss.	2.201 ± 1.090	2.483 ± 0.888	0.772 ± 0.145	$\mathbf{0.525 \pm 0.847}$
		Sphere	8.516 ± 2.821	0.758 ± 0.152	0.285 ± 0.100	$\mathbf{0.012 \pm 0.003}$
		Torus	24.918 ± 4.854	0.312 ± 0.442	0.050 ± 0.016	$\mathbf{0.026 \pm 0.004}$
$\text{ATE}_{rot}\,[rad]$	**Mono**	Coloss.	0.016 ± 0.004	0.010 ± 0.003	0.007 ± 0.004	$\mathbf{0.001 \pm 7\text{e}{-4}}$
		Sphere	0.507 ± 0.026	0.034 ± 0.012	0.019 ± 0.004	$\mathbf{0.001 \pm 1\text{e}{-4}}$
		Torus	0.621 ± 0.166	$0.004 \pm 6\text{e}{-5}$	$0.002 \pm 7\text{e}{-5}$	$\mathbf{0.001 \pm 2\text{e}{-5}}$
	Stereo	Coloss.	0.025 ± 0.009	0.020 ± 0.008	0.011 ± 0.008	$\mathbf{0.005 \pm 0.008}$
		Sphere	0.764 ± 0.258	0.101 ± 0.020	0.024 ± 0.011	$\mathbf{0.001 \pm 2\text{e}{-4}}$
		Torus	0.960 ± 0.303	0.046 ± 0.085	$0.002 \pm 6\text{e}{-4}$	$\mathbf{9\text{e}{-4} \pm 1\text{e}{-4}}$

conjuncture: the accuracy of BA with robustifier is ten times higher than GEA in realistic scenarios, i.e., noisy measurements and wrong data association. We omitted the results of BA without robustifier because their computational cost is negligible, and there is no reason not to use them.

V. Conclusions

In this work, we made an accuracy and runtime comparison between BA and GEA, two different solutions to the problem of estimating the camera trajectory. The results suggest that BA is more accurate and robust, while GEA requires a substantially lower computation and achieves a reasonable accuracy in the absence of data association errors. Based on these considerations we envision GEA as a promising alternative to BA in embedded applications or in settings where the data association is given.

References

[1] S. Agarwal, N. Snavely, S. M. Seitz, and R. Szeliski. Bundle Adjustment in the Large. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, pages 29–42. Springer.

[2] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. Building Rome in a day. In *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*, pages 72–79.

[3] L. Brizi, E. Giacomini, L. D. Giammarino, S. Ferrari, O. Salem, L. D. Rebotti, and G. Grisetti. VBR: A Vision Benchmark in Rome. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pages 15868–15874, May 2024.

[4] G. Grisetti, T. Guadagnino, I. Aloise, M. Colosi, B. D. Corte, and D. Schlegel. Least squares optimization: From theory to practice. *Robotics*, 9(3):51, 2020.

[5] C. Hertzberg, R. Wagner, U. Frese, and L. Schröder. Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds. *Information Fusion*, 14(1):57–77.

[6] V. Indelman, R. Roberts, C. Beall, and F. Dellaert. Incremental Light Bundle Adjustment. In *Proc. of British Machine Vision Conference (BMVC)*, pages 134.1–134.11. British Machine Vision Association.

[7] K. MacTavish and T. D. Barfoot. At all Costs: A Comparison of Robust Cost Functions for Camera Correspondence Outliers. In *Proc. of the Canadian Conf. on Computer and Robot Vision (CRV)*, pages 62–69. IEEE.

[8] R. Mur-Artal and J. D. Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Trans. on Robotics (TRO)*, 33(5):1255–1262, 2017.

[9] A. L. Rodríguez, P. E. López-de Teruel, and A. Ruiz. Reduced epipolar cost for accelerated incremental SfM. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3097–3104.