Correspondence Reweighted Translation Averaging

Lalit Manam and Venu Madhav Govindu

Indian Institute of Science, Bengaluru, India - 560012
{lalitmanam,venug}@iisc.ac.in

Abstract. Translation averaging methods use the consistency of input translation directions to solve for camera translations. However, translation directions obtained using epipolar geometry are error-prone. This paper argues that the improved accuracy of translation averaging should be leveraged to mitigate the errors in the input translation direction estimates. To this end, we introduce weights for individual correspondences which are iteratively refined to yield improved translation directions. In turn, these refined translation directions are averaged to obtain camera translations. This results in an alternating approach to translation averaging. The modularity of our framework allows us to use existing translation averaging methods and improve their results. The efficacy of the scheme is demonstrated by comparing performance with state-of-the-art methods on a number of real-world datasets. We also show that our approach yields reasonably good 3D reconstructions with straightforward triangulation, i.e. without any bundle adjustment iterations.

Keywords: Structure from Motion, Translation Averaging, Reweighting Correspondences

1 Introduction

In Structure-from-Motion (henceforth SfM) [16], given point correspondences across many images, we solve for the corresponding camera motions and 3D scene structure. Many SfM pipelines incrementally grow the solution by adding one camera at a time [27,30,35]. While they work well, incremental methods suffer from drift and have a significant computational load owing to the repeated use of Bundle Adjustment (henceforth BA) [32]. In contrast, batch or global methods [31] determine the absolute poses of the cameras simultaneously (also known as motion averaging [14,15]). Typically one solves for rotations, followed by translations using rotation and translation averaging respectively. Subsequently, we estimate 3D structure given the camera motions, often with a final BA refinement. In this paper, we address the problem of translation averaging.

Using epipolar geometry, we can only recover the translation direction owing to an inherent scale ambiguity which has a number of serious implications. Firstly, it makes translation averaging a challenging problem since we need to

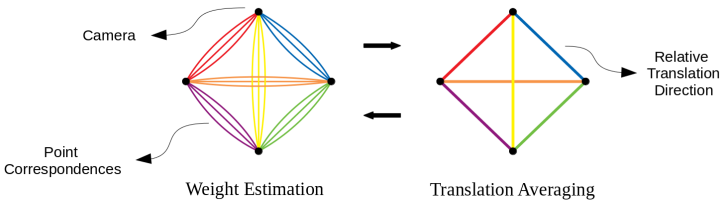


Fig. 1: Schematic diagram of our framework for a viewgraph of camera-camera relationships. Each vertex represents a camera. The multiedges on the left indicate correspondences between camera-camera pairs whereas the edges on the right indicate a relative translation estimate between camera pairs. Weight estimation for point correspondences (resulting in refined translation directions) and translation averaging are alternately carried out in our framework

use relative translation directions to solve for absolute camera translations. Existing methods [13,25,34,36] adopt a variety of approaches to tackle the scale estimation problem inherent to translation averaging. Secondly, in contrast to rotation averaging, determining the feasibility of solutions for a given translation averaging problem is related to the non-trivial issue of parallel rigidity [2,26]. Finally, translation directions recovered from point correspondences using epipolar geometry can be of poor quality owing to the presence of noise and outliers or when the baseline is narrow.

All translation averaging methods solve for camera translations by exploiting consistency relationships the input translation directions should satisfy. In this paper, we argue that while this is useful, the poor quality of input translation directions obtained from epipolar geometry imposes limitations on the accuracy of translation averaging solutions. A key observation of our paper is that instead of using a single estimate of translation directions, we can refine them by introducing weights for individual correspondences. We schematically illustrate our approach in Fig. 1.

Given an initial set of translation directions, we use averaging to obtain camera translations. The translation directions obtained from translation averaging estimates are superior to the original pairwise epipolar geometry based estimates as the averaged solution is based on global consistency. Hence we can use the estimates obtained from translation averaging to refine the weights ascribed to individual point correspondences between a camera pair (shown as multiedges in Fig. 1). Subsequently, the multiedges give us an estimate of translation directions (i.e. edges in a graph) which are averaged to obtain an improved translation estimate.

We emphasize that our reweighting of individual correspondences is based on a global view of geometric consistency and not the standard approach of robustness based on a limited view of two camera epipolar geometry. Our approach is modular in nature and can use different translation averaging and weighting schemes. Thus, our method can take a translation averaging approach and improve it by refining the weighting of individual point correspondences.

The paper is organized as follows: Sec. 2 reviews existing approaches. Sec. 3 presents the formulation and details of our proposed approach. Sec. 4 presents an extensive set of experiments to demonstrate the superiority of our approach compared to the state-of-the-art methods. We provide a discussion of some aspects of our method in Sec. 5 and a brief conclusion in Sec. 6.

2 Related Work

2.1 Rotation Averaging

Translation averaging presumes the availability of rotation estimates, often obtained using rotation averaging which is a well studied problem. Intrinsic methods like [15] exploit the Lie group structure of $\mathbb{SO}_3$. Robustness was incorporated in [4,5,17]. Extrinsic methods like [6,12,22] solve for a relaxed version of the problem. Readers are referred to [10,28,29] and the references therein for recent developments.

2.2 Translation Averaging

Most translation averaging approaches are based on the structure of the essential matrix and trifocal tensor [16]. Govindu [14] proposed to minimize the cross-product between the observed directions and the estimated relative camera translations. Arie-Nachimson *et al.* [1] set up a linear system of cross product constraints based on epipolar geometry. Moulon *et al.* [24] formulated a trifocal tensor with known rotations which converted the problem to aligning triplets instead of pairs.

Jiang *et al.* [18] used camera triplets which converted pairwise constraints into constraints on a triangle. Wilson *et al.* [34] compared the observed and estimated heading directions. They added camera-to-point constraints to make the problem stable and relied on a pre-processing step to remove outliers. Tron *et al.* [33] compared squared relative displacements and used it in a distributed fashion. The Least Unsquared Deviations (LUD) method [25] proposed by Ozyesil *et al.* extended [33] by using L_1 loss for robustness and posed the problem as a convex program. Arrigoni *et al.* [3] proposed to minimize the squared error of the orthogonal projection of the estimated relative translations onto observed directions. In a similar spirit, Goldstein *et al.* [13] proposed ShapeFit/ShapeKick that minimized the orthogonal projection using ADMM but with an L_1 loss for

robustness. Cui *et al.* [8] used feature tracks to construct a linear system and also solved the problem using ADMM. Cui *et al.* [7] used sin-length ratio constraints between cameras and points to estimate camera-to-camera scales and then solved a linear system. Although [7,8] used point correspondences in their method, all correspondences were treated equally. Moreover, multiple estimates of scales with different feature tracks were handled carefully to avoid the influence of outliers. Zhuang *et al.* [36] proposed a Bilinear Angle-based Translation Averaging (BATA) scheme comparing the estimated heading directions from camera translations to that of the observed directions relaxing the cost in [34].

Other related approaches include similarity averaging [9], averaging of essential and fundamental matrices [19,20] and exploiting the structure of the matrix generated from pairwise camera displacements [11].

3 Proposed Method

In this section, we define some preliminaries and develop our proposed algorithm for translation averaging. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a viewgraph, where $\mathcal{V}$ and $\mathcal{E}$ denotes the set of vertices and edges in $\mathcal{G}$ respectively. To each vertex i, we assign a 3D rotation $\mathbf{R}_i \in \mathbb{SO}(3)$ and translation $\mathbf{T}_i \in \mathbb{R}^3$ that denotes its motion with respect to a global frame of reference. Each edge $(i, j) \in \mathcal{E}$ denotes the relative rotation and translation $(\mathbf{R}_{ij}, \mathbf{T}_{ij})$ between camera vertices i and j. We note that owing to the scale ambiguity of epipolar geometry we can only recover the relative translation upto an unknown scale factor, i.e. the unit norm translation direction vectors $\mathbf{t}_{ij} \in \mathbb{S}^2$, resulting in the relationships:

$$\begin{aligned} \mathbf{R}_{ij} =& \mathbf{R}_j \mathbf{R}_i^{-1}, \\ \mathbf{t}_{ij} =& \frac{\mathbf{R}_j(\mathbf{T}_i - \mathbf{T}_j)}{\|\mathbf{R}_j(\mathbf{T}_i - \mathbf{T}_j)\|_2} \qquad (1) \\ \rightarrow \mathbf{v}_{ij} =& -\mathbf{R}_j^{-1}\mathbf{t}_{ij} = \frac{\mathbf{T}_j - \mathbf{T}_i}{\|\mathbf{T}_j - \mathbf{T}_i\|_2} \qquad (2) \end{aligned}$$

where the unit vector $\mathbf{v}_{ij}$ is the translation direction represented in the global frame of reference and is defined for simplicity of notation. We assume that the rotations $\mathbf{R}_i$ for $i \in \mathcal{V}$ are either known or estimated using rotation averaging. Thus, the translation averaging problem of interest is one of recovering the absolute translations $\mathbf{T}_i$ for $i \in \mathcal{V}$ given the relative translation directions $\mathbf{t}_{ij}$ for $(i, j) \in \mathcal{E}$. For each edge $(i, j) \in \mathcal{E}$, we associate a number of point correspondences between cameras i and j. We denote these correspondences as $\{(\mathbf{p}_i^k, \mathbf{q}_j^k) | k \in \mathcal{I}_{ij}\}$ where $\mathbf{p}$ and $\mathbf{q}$ denote the homogeneous representation of correspondences in camera i and j respectively which are normalized to unit vectors, $\mathcal{I}_{ij}$ is the set of point indexes for the edge $(i, j) \in \mathcal{E}$, and k is the point index. We can now write the epipolar constraint between cameras i and j for the k-th point correspondence as

$$\left(\mathbf{q}_j^k\right)^T \left(\mathbf{t}_{ij} \times \mathbf{R}_{ij}\mathbf{p}_i^k\right) = 0 \qquad (3)$$

Denoting $\tilde{\mathbf{p}}_i^k = \mathbf{R}_i^{-1}\mathbf{p}_i^k$ and $\tilde{\mathbf{q}}_j^k = \mathbf{R}_j^{-1}\mathbf{q}_j^k$, the epipolar constraint of Eqn. 3 can be rewritten as

$$\begin{aligned} &\left(\mathbf{q}_j^k\right)^T \left(\mathbf{t}_{ij} \times \mathbf{R}_j\mathbf{R}_i^{-1}\mathbf{p}_i^k\right) = 0 \\ \Rightarrow &\left(\mathbf{q}_j^k\right)^T \mathbf{R}_j \left(\mathbf{R}_j^{-1}\mathbf{t}_{ij} \times \mathbf{R}_i^{-1}\mathbf{p}_i^k\right) = 0 \qquad (4) \\ \Rightarrow &\left(\mathbf{R}_j^{-1}\mathbf{q}_j^k\right)^T \left(-\mathbf{v}_{ij} \times \left(\mathbf{R}_i^{-1}\mathbf{p}_i^k\right)\right) = 0 \\ \Rightarrow &\left(\mathbf{m}_{ij}^k\right)^T \mathbf{v}_{ij} = 0 \qquad (5) \end{aligned}$$

where $\mathbf{m}_{ij}^k = \tilde{\mathbf{q}}_j^k \times \tilde{\mathbf{p}}_i^k$. Eqn. 4 is obtained using the relationship $\mathbf{a} \times \mathbf{S}\mathbf{b} = \mathbf{S}^{-T}(\mathbf{S}^{-1}\mathbf{a} \times \mathbf{b})$ (upto scale) for any invertible matrix $\mathbf{S} \in \mathbb{R}^{3\times3}$ and $\mathbf{a}, \mathbf{b} \in \mathbb{R}^3$ (Appendix A4.2 in [16]). Collecting the relationships in Eqn. 5 for all $k \in \mathcal{I}_{ij}$ we get $\mathbf{M}_{ij}\mathbf{v}_{ij} = \mathbf{0}$ where the k-th row of $\mathbf{M}_{ij}$ is $\mathbf{m}_{ij}^{kT}$. Further, to account for our confidence in each observation, we assign a scalar weight w_{ij}^k to every point correspondence $(\mathbf{p}_i^k, \mathbf{q}_j^k)$. We normalize these weights for each edge $(i,j) \in \mathcal{E}$ such that $\sum_k w_{ij}^k = 1$. We define a diagonal matrix $\mathbf{W}_{ij}$ where the k-th entry on the diagonal is w_{ij}^k. Finally, we denote the set of all the translations as $\mathbb{T} = \{\mathbf{T}_1, \cdots, \mathbf{T}_N\}$ where $N = |\mathcal{V}|$ is the number of cameras.

3.1 Our Framework for Translation Averaging

When Eqn. 5 is not exactly satisfied due to noise, the least squares solution for $\mathbf{v}_{ij}$ is the smallest right singular vector of the matrix $\mathbf{M}_{ij}$. Analogously, the weighted least squares solution is the smallest right singular vector of $\mathbf{W}_{ij}\mathbf{M}_{ij}$. While this holds for a single camera pair $(i,j) \in \mathcal{E}$, in translation averaging, we seek a global solution that is most consistent with the observations on each individual edge. Thus we formulate our optimization problem as

$$\min_{\mathbb{T}} \sum_{(i,j)\in\mathcal{E}} \|\mathbf{W}_{ij}\mathbf{M}_{ij}\mathbf{v}_{ij}(\mathbb{T})\|_2^2 \qquad (6)$$

where we denote $\mathbf{v}_{ij}(\mathbb{T})$ to emphasize that it is a function of the global camera translations $\mathbb{T}$. $\mathcal{C}$ corresponds to the constraint set to fix origin and scale ambiguity in the problem. It will be immediately noted (from Eqn. 2) that Eqn. 6 is a highly non-linear problem and is challenging to solve for large-scale SfM datasets. When robust estimators are used, such as IRLS, the weights w_{ij}^k are iteratively updated based on a robust loss minimization. In contrast, as stated earlier, we wish to refine our weights based on the global consistency of the translation directions. In our approach, after initialization, we solve for translation directions (given translations) and then solve translation averaging (given translation directions). This is repeated till convergence. We note that our alternating approach is akin to Expectation Maximization (EM) [23] where the weights for point correspondences play the role of latent or unobserved variables. Specifically, we have the following alternating steps:

Algorithm 1: Correspondence Reweighted Translation Averaging (CReTA)

```
Initialize global translations 𝕋
while not converged do
    Update weights ∀(p_i^k, q_j^k)
    Estimate {v_ij | ∀(i,j) ∈ ℰ}
    Solve Translation Averaging using {v_ij}
end
```

- **Weights Update:** Given the current estimate of global translations $\mathbb{T}$, for each point correspondence $(\mathbf{p}_i^k, \mathbf{q}_j^k)$, we compute the residual error using Eqn. 5, i.e. $e_{ij}^k = \mathbf{m}_{ij}^{kT}\mathbf{v}_{ij}(\mathbb{T})$. We map these errors e_{ij}^k into weights w_{ij}^k using a suitable function to denote our current confidence in that correspondence.
- **Translations Update:** Given the weights defined above, the minimization problem in Eqn. 6 is completely defined. Directly minimizing Eqn. 6 is infeasible for large-scale datasets owing to the very large number of correspondences involved and the non-linear nature of the problem. Instead, we use a two-step procedure as follows: For each edge $(i,j) \in \mathcal{E}$, we solve for the translation direction $\mathbf{v}_{ij}$ as the null space of $\mathbf{W}_{ij}\mathbf{M}_{ij}$. As described below, we use these estimated $\mathbf{v}_{ij}$ as inputs for a translation averaging method to solve for the global translations $\mathbb{T}$. As noted in [36], this is akin to functional lifting since we solve for $\mathbf{v}_{ij}$ for all edges and then average them in terms of the smaller number of camera translations. This approach makes our optimization using point correspondence weights tractable with lower computation requirements than directly solving Eqn. 6. We also note that the modular nature of our formulation allows us to use any translation averaging scheme.

Choice of Translation Averaging Method: In an excellent discussion, [36] shows that the weakness of the LUD formulation of [25] is its use of an inequality constraint to remove scale ambiguity and prevent the collapse of camera translations to a point. They further show that this problem can be mitigated by revising the scale constraint, resulting in a Revised-LUD or RLUD method, which is shown to have a minimum identical to that of Shapefit/ShapeKick [13]. For the remainder of this paper, LUD refers to the original formulation in [25] and RLUD refers to the Revised-LUD modification given in [36]. We provide below the formulations for RLUD and BATA which we use in our experiments.

RLUD:

$$\min_{\mathbf{T}_{i,i\in\mathcal{V}},\lambda_{ij,(i,j)\in\mathcal{E}}} \|\mathbf{T}_j - \mathbf{T}_i - \lambda_{ij}\mathbf{v}_{ij}\|_2 \tag{7}$$
$$\text{s.t. } \sum_{i\in\mathcal{V}} \mathbf{T}_i = \mathbf{0},\ \sum_{(i,j)\in\mathcal{E}} \langle \mathbf{T}_j - \mathbf{T}_i, \mathbf{v}_{ij}\rangle = 1, \lambda_{ij} \geq 0,\ \forall (i,j) \in \mathcal{E}$$

BATA:

$$\min_{\mathbf{T}_{i,i\in\mathcal{V}},\gamma_{ij,(i,j)\in\mathcal{E}}} \rho\left(\|\left(\mathbf{T}_j-\mathbf{T}_i\right)\gamma_{ij}-\mathbf{v}_{ij}\|_2\right) \tag{8}$$
$$\text{s.t. } \sum_{i\in\mathcal{V}}\mathbf{T}_i=\mathbf{0},\ \sum_{(i,j)\in\mathcal{E}}\langle\mathbf{T}_j-\mathbf{T}_i,\mathbf{v}_{ij}\rangle=1,\gamma_{ij}\geq 0,\ \forall(i,j)\in\mathcal{E}$$

The zero centroid and dot product constraints in Eqns. 7 and 8 remove the inherent origin and scale ambiguity. ρ denotes a robust loss function. λ_{ij} and γ_{ij} are non-negative variables that are ideally equal to baseline and inverse baseline for the edge (i,j) respectively. It can be seen that RLUD compares the relative displacements (by also computing the translation scales λ_{ij}) while BATA compares the heading directions. In other words, RLUD and BATA are representative of the two approaches feasible for translation averaging, i.e. comparing directions or comparing translation vectors.

Implementation details: We provide a high level description of our approach in Algorithm 1. In order to remove outlier or low quality correspondences, after the initial estimate of translations, for each edge (i,j), we compute the weights for each correspondence pair and remove the bottom 25% of such correspondences. In every iteration, edges are pruned when the estimated $\mathbf{v}_{ij}$ deviates by more than 40° from the equivalent derived from the global translations estimated in the previous iteration. In the Weights Update step, we use the function $w_{ij}^k=\frac{\alpha^2}{\alpha^2+e_{ij}^{k2}}$, with $\alpha=0.01$. Other weighting functions are also feasible here. Estimation of $\mathbf{v}_{ij}$ involves resolving its sign ambiguity which can either be done using chirality constraint or comparing with the directions from the current estimate of global translations. The latter one is used in our implementation. In our experiments, we use RLUD and BATA for translation averaging and denote our corresponding methods as CReTA-RLUD and CReTA-BATA respectively. For CReTA-RLUD, we remove edges with negative scale factors λ_{ij} in each iteration. Algorithm 1 is run till the absolute fractional change of the translation averaging cost function is less than 10^{-5} or mean change in translations is less than 10^{-6} or the maximum number of iterations N_{max} is reached. For CReTA-RLUD and CReTA-BATA, N_{max} equals 50 and 10 respectively[1]. In addition, both RLUD and BATA are themselves iterative methods run for 20 iterations each. Finally, for BATA, we use a Cauchy loss with scale $\beta=0.1$, as used by the authors in [36].

4 Experiments

In this section, we provide experimental comparisons of our method with state-of-the-art methods for translation averaging on synthetic and real datasets. For camera rotations, we use the rotation averaging solution obtained using the code

[1] For large datasets with number of cameras greater than 2000, N_{max} equals 30 and 5 for CReTA-RLUD and CReTA-BATA respectively.

provided by [5][2]. For all experiments, the maximal parallel rigid component of the viewgraph is extracted based on [21]. We note that although the translation directions change for each iteration in our approach, we do not recompute parallel rigidity as the maximal component does not change significantly over the iterations. We use LUD implemented in Theia [31] and BATA's code provided by the authors[3]. Our method is implemented in MATLAB. To quantitatively evaluate the performance of different schemes, the estimated camera translations are robustly aligned to the ground truth using the code provided by [34][4]. All experiments are performed on a PC with Intel Xeon Silver 4210 processor with 128 GB RAM. Finally, in each table, the best performing method for each dataset is highlighted in **bold**, $\boldsymbol{\mu}$ and $\hat{\boldsymbol{\mu}}$ imply mean and median errors respectively.

4.1 Synthetic Data

We carry out experiments with synthetic data to study the comparative behaviour of different methods in the presence of noise and outliers. To validate the usefulness of weighting every point correspondence based on global consistency, RLUD and BATA are compared to our methods, CReTA-RLUD and CReTA-BATA. For creating synthetic data as close to the real data as possible, we use the ground truth from two 1DSfM datasets [34], i.e. *Montreal Notre Dame* (450 nodes, 52340 edges) and *Tower of London* (467 nodes, 23777 edges). We chose these datasets as they have a similar number of cameras but a different number of edges. We refer to these synthetic datasets as MND_{syn} and TOL_{syn}. For these synthetic datasets, we create point correspondences by projecting the ground truth 3D points onto the cameras using the ground truth poses and then applying camera calibration as provided in the dataset. Only the point features within the image dimensions and with a positive depth in the camera coordinate frame are retained. For every edge, a maximum of 1000 correspondences generated in this fashion are retained. This creates a perfectly clean dataset with real-world camera motions and 3D structure. Now we perturb the image points with Gaussian noise $\mathcal{N}(0, \sigma^2)$ with $\sigma \in \{1, 3\}$ pixels. 10 instances of both datasets are generated for the two noise levels. To generate outliers, 30% of the correspondences in each edge are perturbed with Gaussian noise $\mathcal{N}(0, \sigma_o^2)$ with $\sigma_o = 10$ pixels. For every dataset, relative translations are computed in a RANSAC loop using epipolar geometry with rotations set to ground truth. This removes the effect of rotation errors from the problem. For these realizations, we extract the corresponding maximal parallel rigid graph, which are used as inputs in our experiments.

To evaluate the accuracy of camera translations, we use the normalized root mean square error (NRMSE) as in [25] and [36]. Let $\mathbf{T}_i^{gt}$ be the set of ground truth camera translations and $\mathbf{T}_i^{avg}$ estimated from different methods, then

[2] `https://ee.iisc.ac.in/cvlab/research/rotaveraging/`
[3] `https://bbzh.github.io/document/BATA.zip`
[4] `https://github.com/wilsonkl/SfM_Init`

Table 1: Camera translation errors (in meters) for synthetic datasets. The reported values are averaged over 10 instances each (wo: with outliers)

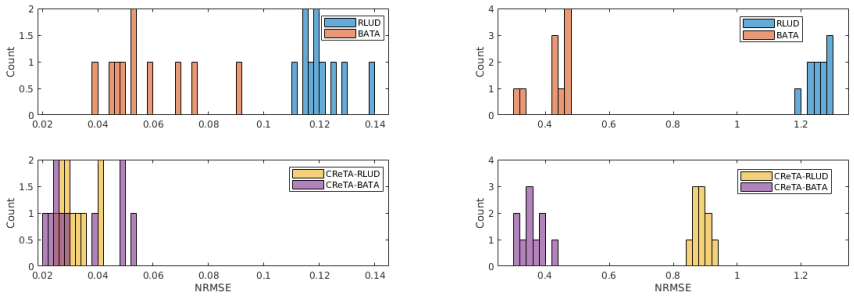
Dataset	RLUD		BATA		CReTA-RLUD		CReTA-BATA	
	$\boldsymbol{\mu}$	$\hat{\boldsymbol{\mu}}$	$\boldsymbol{\mu}$	$\hat{\boldsymbol{\mu}}$	$\boldsymbol{\mu}$	$\hat{\boldsymbol{\mu}}$	$\boldsymbol{\mu}$	$\hat{\boldsymbol{\mu}}$
MND_{syn}, $\sigma = 1$	1.64	1.10	0.17	0.06	**0.12**	0.07	0.13	**0.05**
MND_{syn}, $\sigma = 1$, wo	1.61	1.09	0.15	**0.06**	0.14	0.08	**0.11**	**0.06**
MND_{syn}, $\sigma = 3$	1.55	1.04	0.25	0.12	**0.18**	**0.11**	0.22	0.14
MND_{syn}, $\sigma = 3$, wo	1.49	0.97	0.29	0.14	**0.19**	**0.11**	0.25	0.16
TOL_{syn}, $\sigma = 1$	12.78	2.56	2.88	1.75	7.38	**0.46**	**1.19**	0.52
TOL_{syn}, $\sigma = 1$, wo	12.36	2.13	2.56	1.56	7.51	**0.63**	**1.64**	0.92
TOL_{syn}, $\sigma = 3$	12.44	2.64	2.50	1.46	7.76	**1.00**	**2.06**	1.11
TOL_{syn}, $\sigma = 3$, wo	12.47	2.73	2.72	1.40	7.89	**0.99**	**2.52**	1.20

Table 2: Comparison of input and output relative translation directions (in degrees) on synthetic data. The reported values are averaged over 10 instances each (RMS: root mean square error; wo: with outliers)

Dataset	Input		CReTA-RLUD		CReTA-BATA	
	$\boldsymbol{\mu}$	RMS	$\boldsymbol{\mu}$	RMS	$\boldsymbol{\mu}$	RMS
MND_{syn}, $\sigma = 1$	0.33	3.80	**0.17**	**1.45**	0.23	**1.45**
MND_{syn}, $\sigma = 1$, wo	0.52	5.35	**0.18**	**1.20**	0.22	**1.20**
MND_{syn}, $\sigma = 3$	1.50	8.17	**0.72**	3.03	0.89	**3.00**
MND_{syn}, $\sigma = 3$, wo	2.00	10.34	**0.77**	3.12	0.94	**3.09**
TOL_{syn}, $\sigma = 1$	0.79	7.13	0.37	1.55	**0.30**	**1.53**
TOL_{syn}, $\sigma = 1$, wo	1.27	9.55	0.44	2.48	**0.34**	**2.46**
TOL_{syn}, $\sigma = 3$	2.74	13.56	**1.00**	3.95	**1.00**	**3.89**
TOL_{syn}, $\sigma = 3$, wo	3.60	16.15	**1.05**	3.92	1.07	**3.88**

$NRMSE = \sqrt{\sum_{i \in \mathcal{V}} \|\mathbf{T}_i^{gt} - \mathbf{T}_i^{avg}\|_2^2}$. Both $\mathbf{T}_i^{avg}$ and $\mathbf{T}_i^{gt}$ are normalized such that $\sum_{i \in \mathcal{V}} \mathbf{T}_i = 0$ and $\sum_{i \in \mathcal{V}} \|\mathbf{T}_i\|_2^2 = 1$. The evaluation is also done in terms of mean and median errors of camera translations (after aligning the solutions to ground truth) and comparing the input and output relative translations.

In Table 1, we show the translation errors for each datasets with differing scenarios ($\sigma = \{1, 3\}$, with and without outliers) averaged over 10 instances. As can be clearly seen, our CReTA methods are significantly better in accuracy compared to LUD and BATA. In Fig. 2, we show the distribution for NRMSE for different methods for the 10 instances when $\sigma = 3$ and with outliers. We show the results for only one setting of noise and outliers for visual clarity (other settings are shown in the supplementary material). The large leftward shift of the distributions for our methods clearly demonstrates a significant improvement in performance over the corresponding translation averaging methods used. Further, in Table 2, we compare the quality of input to that of output relative translations for different methods. It is seen that relative translation directions are substantially improved in our framework.



(a) MND_{syn}, $\sigma = 3$ with outliers (b) TOL_{syn}, $\sigma = 3$ with outliers

Fig. 2: Comparison of the histograms of NRMSE in 10 instances for two synthetic datasets. The leftward shift for our methods clearly indicates a significant improvement in performance

Table 3: Camera translation errors (in meters) on 1DSfM [34] datasets ($|\boldsymbol{\mathcal{V}}|$: number of nodes, $|\boldsymbol{\mathcal{E}}|$: number of edges)

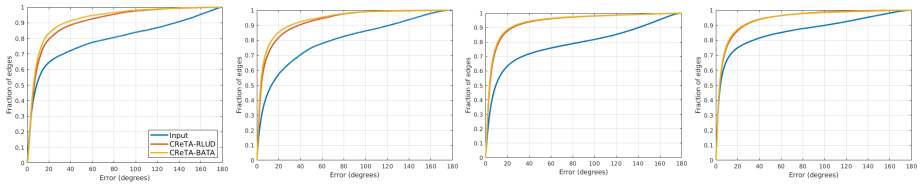
Dataset	$\|\boldsymbol{\mathcal{V}}\|$	$\|\boldsymbol{\mathcal{E}}\|$	**LUD** [25]		**ShapeFit** [13]		**BATA** [36]		**CReTA-RLUD (Ours)**		**CReTA-BATA (Ours)**	
			$\boldsymbol{\mu}$	$\hat{\boldsymbol{\mu}}$	$\boldsymbol{\mu}$	$\hat{\boldsymbol{\mu}}$	$\boldsymbol{\mu}$	$\hat{\boldsymbol{\mu}}$	$\boldsymbol{\mu}$	$\hat{\boldsymbol{\mu}}$	$\boldsymbol{\mu}$	$\hat{\boldsymbol{\mu}}$
Alamo (ALM)	586	81437	2.7	**0.5**	**0.9**	**0.5**	2.0	0.6	2.0	**0.5**	2.0	**0.5**
Ellis Island (ELS)	229	14728	6.9	3.6	12	**1.9**	6.7	3.2	**6.0**	2.9	6.2	3.3
Gendarmenkmart (GMM)	686	27145	31.2	11.3	-	-	31.3	11.4	**31.0**	11.2	31.5	**11.1**
Madrid Metropolis (MDR)	325	11995	8.4	1.9	81	6.0	6.9	**1.6**	7.6	1.7	**6.1**	**1.6**
Montreal Notre Dame (MND)	461	45737	0.9	0.5	1.7	0.8	0.8	0.5	0.9	0.5	**0.7**	**0.4**
Notre Dame (ND)	552	80647	1.2	0.3	1.5	**0.2**	**1.0**	**0.2**	2.1	1.1	**1.0**	**0.2**
NYC Library (NYC)	337	14365	2.2	0.8	162	1.4	2.1	1.7	**2.0**	**0.7**	**2.0**	**0.7**
Piazza del Popolo (PDP)	334	20974	3.8	2.8	5.9	3.6	**3.4**	**2.0**	4.5	3.4	3.8	2.5
Piccadilly (PIC)	2362	201600	**2.8**	1.3	15	1.2	3.2	**1.1**	**2.8**	**1.1**	3.0	**1.1**
Roman Forum (ROF)	1069	54207	11.9	3.3	25	4.3	8.3	2.0	13.3	3.3	**7.7**	**1.7**
Tower of London (TOL)	474	19252	14.9	3.2	164	**2.3**	9.3	3.0	13.3	3.1	**9.0**	3.0
Trafalgar (TFG)	4900	542480	8.4	5.3	-	-	7.9	4.2	8.0	4.4	**7.5**	**4.1**
Union Square (USQ)	825	19899	10.6	6.1	47	8.9	**10.2**	5.6	10.6	5.4	10.5	**4.9**
Vienna Cathedral (VNC)	826	82793	**5.1**	2.1	11	**1.9**	12.0	2.1	6.5	2.2	6.7	**1.9**
Yorkminster (YKM)	430	22692	7.6	1.8	-	-	**5.1**	**1.3**	7.0	1.6	6.0	2.0

4.2 Real World Data

In this subsection, we present results on real unordered image datasets provided by the authors of 1DSfM [34]. These datasets are pre-processed in a manner similar to that suggested in [25]: Rotation averaging is performed and inconsistent edges with an error greater than 10° are removed. Subsequently, the initial translation directions are estimated using the epipolar geometric relationship (Eqn. 5) in a RANSAC loop. The results for our methods CReTA-RLUD and CReTA-BATA are shown in Table 3 along with other state-of-the-art methods. Since ShapeFit provides multiple results, the overall best results are cited. It can be seen that CReTA has the best performance or is similar in quality to the best solution for most of the datasets. In particular, the improvement in mean errors is significant for many datasets when compared to the respective transla-

Table 4: Camera translation errors (in meters) on 1DSfM datasets using the initialization provided

Dataset	LUD		BATA		CReTA-RLUD		CReTA-BATA	
	μ	$\hat{\mu}$	μ	$\hat{\mu}$	μ	$\hat{\mu}$	μ	$\hat{\mu}$
ALM	5.0	2.8	3.4	0.6	2.4	0.6	**2.1**	**0.5**
ELS	9.7	3.7	11.6	**1.4**	**7.1**	2.9	8.3	3.0
GMM	46.6	**20.5**	45.6	23.4	44.5	22.8	**41.8**	20.9
MDR	19.0	9.2	23.2	2.8	12.7	2.2	**11.9**	**1.7**
MND	1.6	0.9	**1.3**	**0.6**	2.6	1.4	1.5	0.7
ND	4.1	1.6	2.1	0.3	1.7	0.5	**1.1**	**0.2**
NYC	4.5	2.2	3.5	**0.7**	2.6	**0.7**	**2.3**	**0.7**
PDP	6.3	1.9	6.7	1.6	5.6	**1.2**	**5.4**	**1.2**
PIC	6.2	3.8	5.1	**1.3**	4.4	2.3	**3.4**	**1.3**
ROF	25.1	14.6	11.3	**1.6**	18.2	9.4	**9.3**	1.8
TOL	24.9	8.7	17.6	**2.0**	19.5	2.6	**16.8**	2.9
TFG	16.7	13.3	11.7	**4.4**	9.9	6.2	**9.1**	5.7
USQ	12.2	7.6	13.3	**4.8**	11.6	6.2	**10.5**	5.0
VNC	14.2	7.0	10.5	2.1	8.5	2.4	**5.8**	**1.8**
YKM	15.6	6.8	9.4	**1.3**	7.3	1.5	**5.2**	1.5



(a) ELS (b) MDR (c) PIC (d) VNC

Fig. 3: Cumulative error distributions (in degrees) for relative translation directions for inputs and CReTA estimates

tion averaging scheme used, i.e. CReTA-RLUD vs. LUD and CReTA-BATA vs. BATA.

The 1DSfM datasets [34] also provide an estimate of the translation directions $\mathbf{t}_{ij}$ which are of inferior quality compared to that estimated and used in the above experiment. To assess the performance of the methods with low quality inputs, we supply the translation directions given in the 1DSfM datasets [34] to different methods and compare the camera translation accuracies in Table 4. It can be seen that even when the initial input translation directions are of inferior quality, our CReTA approach is able to obtain better accuracies with considerable improvement in the mean errors when compared to the respective translation averaging scheme used. To further illustrate the improvement obtained, in Fig. 3, we compare the cumulative error distributions of the relative translation directions obtained using CReTA and the input translation directions. As can be seen, our approaches significantly improve upon the initial directions. These results suggest that weighting point correspondences with global consistency information can improve the performance of the translation averaging methods even with low quality input translation directions.

Table 5: Reprojection errors (in pixels) and number of 3D points ($N_p \times 10^3$) reconstructed after triangulation

Dataset	LUD			BATA			CReTA-RLUD			CReTA-BATA		
	μ	$\hat{\mu}$	N_p	μ	$\hat{\mu}$	N_p	μ	$\hat{\mu}$	N_p	μ	$\hat{\mu}$	N_p
ALM	6.6	5.4	83	7.7	6.7	120	**3.0**	**1.7**	**223**	3.5	2.3	221
ELS	8.3	7.3	37	7.6	6.3	56	**4.5**	**3.2**	62	4.7	3.6	**69**
GMM	7.1	6.5	54	6.8	5.7	99	**5.1**	**3.6**	146	5.5	4.2	**159**
MDR	7.1	7.0	11	7.7	6.8	38	**4.9**	**3.5**	60	5.1	3.8	**74**
MND	8.2	7.1	87	7.3	5.9	123	**3.6**	**2.3**	178	3.8	2.5	**184**
ND	7.9	6.9	53	8.2	7.0	117	**4.8**	**3.2**	232	5.4	3.8	**240**
NYC	8.0	7.6	32	7.5	6.5	65	**3.9**	**2.5**	107	4.5	3.1	**113**
PDP	7.6	6.5	34	7.2	5.9	40	**3.8**	**2.4**	64	4.5	3.1	**65**
PIC	7.3	6.5	121	7.2	6.1	187	**5.3**	**3.7**	298	5.6	4.1	**318**
ROF	7.2	6.4	87	7.4	6.3	204	**4.5**	**3.0**	347	5.0	3.6	**373**
TOL	5.9	4.8	63	6.7	5.4	95	**5.1**	**3.4**	180	5.5	3.8	**189**
TFG	7.1	6.1	308	7.0	5.9	416	**4.9**	**3.3**	649	5.5	3.9	**654**
USQ	7.8	6.8	34	7.5	6.5	38	**6.3**	**4.9**	62	6.5	5.1	**66**
VNC	7.8	6.9	124	7.6	6.4	194	**4.8**	**3.3**	341	5.0	3.6	**354**
YKM	7.6	6.9	56	7.4	6.3	96	**5.1**	**3.6**	153	5.7	4.2	**171**

Impact on 3D Reconstruction: In Tables 3 and 4, we compare the quality of translation estimates based on the available pseudo ground truth. But we believe that comparing the mean and median errors for translation estimates are inadequate since, in SfM, we are also interested in estimating 3D scene structure. In order to understand this aspect of camera motion estimation, we carry out the following experiment. For all datasets used, we extract a large number of correspondences using COLMAP [27]. We use the camera motions obtained in the experiment in Table 4 and carry out triangulation using Theia [31]. This allows us to assess the quality of the different translation solutions in terms of their impact on 3D reconstruction. The reprojection errors, shown after triangulation using the different translation solutions in Table 5, are independent of the pseudo ground truth and indicate the quality of reconstruction. As can be seen, our methods yield significantly lower reprojection errors. Specifically, CReTA-RLUD has the least mean and median reprojection errors followed by CReTA-BATA. The Theia package removes triangulated points with a reprojection error larger than 15 pixels. We note that our methods yield many more triangulated points with CReTA-BATA producing the highest number of 3D points for all datasets except ALM.

In Fig. 4 we illustrate the quality of translation estimation using our reweighting scheme by visualizing the 3D reconstructions obtained. We show the reconstructions obtained by triangulation using our CReTA-RLUD solution, with additional reconstructions provided in the supplementary material. The corresponding point clouds obtained using full BA are shown for reference. As can be seen, we obtain reasonably good quality reconstructions by straightforward triangulation without having to carry out any bundle adjustment refinements. The high quality of our triangulations indicate that our translation estimates are consistent with the point correspondences.

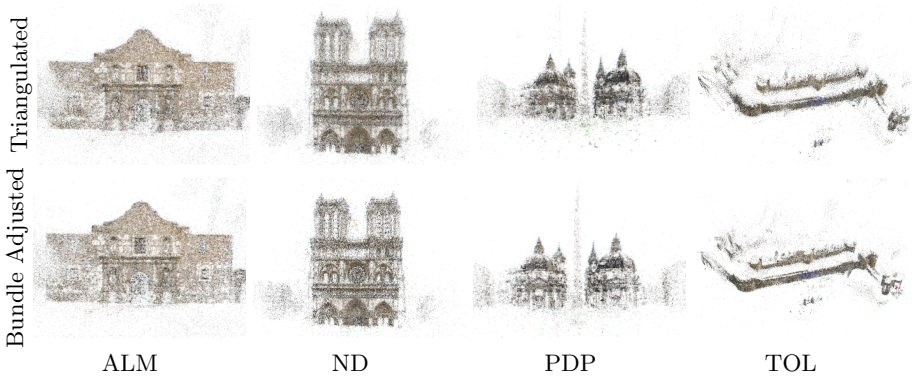


Fig. 4: Reconstructions obtained with triangulation using our CReTA-RLUD translation estimate (first row) compared to bundle adjustment (second row)

5 Discussion

As we have shown in Sec. 4, using our refined weighting of point correspondences improves the quality of the translation estimates. In this section, we briefly consider some other issues of significance.

Ablation Study: As indicated earlier, in our implementation, we remove some of the point correspondences from each edge $(i, j) \in \mathcal{E}$ after the first translation averaging estimate. This is especially useful when the input relative translations are inferior as the translation averaging steps improve the overall solution, and correspondences with low weights indicate that they are low in quality with respect to the translation averaging solution. For this study, we do not prune edges to understand the effect of removing point correspondences exclusively. Table 6 shows that removing point correspondences in the first iteration results in an improvement in our CReTA results for almost all datasets, with significant improvements for ELS, MDR, USQ and VNC datasets.

Computation Time: While RLUD and BATA are based on a single optimization, CReTA methods carry out repeated optimizations with refined input translation directions in each iteration increasing the computation load of CReTA as shown in Table 7. We believe that the additional computation time for our approach can be significantly reduced with a C++ implementation.

Limitations: Finally, we note that our improvement on translation averaging leverages the image correspondences in the dataset. We may not get as significant an improvement in performance over the translation averaging method used if there are very few point correspondences or if they have high noise and outlier ratios. Further, our method cannot be used if only the translation direction estimates are available without access to the correspondences.

Table 6: Impact of removing low quality correspondences in 1DSfM datasets after first iteration. Entries marked in **bold** shows improvement of a given method and not comparing all variants (woRC: without Removing Correspondences)

Dataset	CReTA-RLUD woRC		CReTA-RLUD		CReTA-BATA woRC		CReTA-BATA	
	μ	$\hat{\mu}$	μ	$\hat{\mu}$	μ	$\hat{\mu}$	μ	$\hat{\mu}$
ALM	**2.2**	**0.6**	**2.2**	**0.6**	**2.3**	0.7	2.4	**0.5**
ELS	28.3	3.3	**9.5**	**3.0**	21.0	3.9	**12.2**	**3.0**
GMM	45.1	21.3	**43.9**	**20.8**	**44.1**	**20.7**	**44.1**	21.3
MDR	16.4	2.7	**14.9**	**2.3**	19.8	2.8	**13.9**	**1.9**
MND	2.3	1.2	**1.9**	**1.0**	1.2	0.6	**0.8**	**0.4**
ND	1.2	0.3	**1.1**	**0.3**	1.7	0.6	**1.3**	**0.2**
NYC	4.3	1.0	**2.8**	**0.9**	3.1	0.9	**2.9**	**0.7**
PDP	6.4	**1.2**	**6.2**	1.3	6.8	1.8	**6.2**	**1.3**
PIC	**4.0**	**2.0**	**4.0**	2.3	4.2	1.5	**3.3**	**1.2**
ROF	12.1	4.0	**11.7**	**3.6**	8.6	2.0	**8.2**	**1.8**
TOL	19.6	2.8	**20.0**	**2.6**	**13.4**	3.2	14.8	**3.0**
TFG	13.6	10.0	**10.2**	**7.1**	13.7	7.8	**8.9**	**5.1**
USQ	26.3	9.1	**15.9**	**5.5**	17.9	7.3	**13.0**	**5.7**
VNC	11.4	4.0	**7.3**	**2.0**	14.0	2.7	**8.2**	**1.8**
YKM	6.7	**1.5**	**6.3**	**1.5**	12.9	**1.5**	**8.7**	**1.5**

Table 7: Computation time (in seconds) for different schemes

Dataset	RLUD	BATA	CReTA-RLUD	CReTA-BATA
ALM	42	54	225	145
ELS	6	10	33	28
GMM	14	25	84	71
MDR	7	14	45	32
MND	9	29	128	83
ND	46	56	285	188
NYC	6	14	44	31
PDP	8	16	42	43
PIC	211	249	747	521
ROF	24	52	172	128
TOL	9	19	60	43
TFG	543	862	1996	1271
USQ	8	17	63	50
VNC	54	74	283	176
YKM	11	16	74	53

6 Conclusion

This paper presents CReTA, a modular framework that iteratively refines the input translation directions by weighting individual point correspondences. The modularity of our approach allows us to use a translation averaging method and improve upon its performance. This improvement is reflected in the quality metrics for comparing translation estimates. Also of significance is the fact that our approach yields reasonably good reconstructions with triangulation when compared with BA results.

Acknowledgments: Lalit Manam is supported by a Prime Minister's Research Fellowship, Government of India. This research was supported in part by a Core Research Grant from Science and Engineering Research Board, Department of Science and Technology, Government of India.

References

1. Arie-Nachimson, M., Kovalsky, S.Z., Kemelmacher-Shlizerman, I., Singer, A., Basri, R.: Global motion estimation from point matches. In: 2012 Second international conference on 3D imaging, modeling, processing, visualization & transmission. pp. 81–88. IEEE (2012)
2. Arrigoni, F., Fusiello, A.: Bearing-based network localizability: a unifying view. IEEE transactions on pattern analysis and machine intelligence **41**(9), 2049–2069 (2018)
3. Arrigoni, F., Rossi, B., Fusiello, A.: Robust and efficient camera motion synchronization via matrix decomposition. In: International Conference on Image Analysis and Processing. pp. 444–455. Springer (2015)
4. Chatterjee, A., Govindu, V.M.: Efficient and robust large-scale rotation averaging. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 521–528 (2013)
5. Chatterjee, A., Govindu, V.M.: Robust relative rotation averaging. IEEE transactions on pattern analysis and machine intelligence **40**(4), 958–972 (2017)
6. Crandall, D., Owens, A., Snavely, N., Huttenlocher, D.: Discrete-continuous optimization for large-scale structure from motion. In: CVPR 2011. pp. 3001–3008. IEEE (2011)
7. Cui, H., Shen, S., Hu, Z.: Robust global translation averaging with feature tracks. In: 2016 23rd International Conference on Pattern Recognition (ICPR). pp. 3727–3732. IEEE (2016)
8. Cui, Z., Jiang, N., Tang, C., Tan, P.: Linear global translation estimation with feature tracks. In: Proc. ECCV. vol. 3, pp. 61–75 (2014)
9. Cui, Z., Tan, P.: Global structure-from-motion by similarity averaging. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 864–872 (2015)
10. Dellaert, F., Rosen, D.M., Wu, J., Mahony, R.E., Carlone, L.: Shonan rotation averaging: Global optimality by surfing $so(p)^n$. In: ECCV (6). Lecture Notes in Computer Science, vol. 12351, pp. 292–308. Springer (2020)
11. Dong, Q., Gao, X., Cui, H., Hu, Z.: Robust camera translation estimation via rank enforcement. IEEE transactions on cybernetics (2020)
12. Eriksson, A., Olsson, C., Kahl, F., Chin, T.: Rotation averaging and strong duality. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 127–135 (2018)
13. Goldstein, T., Hand, P., Lee, C., Voroninski, V., Soatto, S.: Shapefit and shapekick for robust, scalable structure from motion. In: European Conference on Computer Vision. pp. 289–304. Springer (2016)
14. Govindu, V.M.: Combining two-view constraints for motion estimation. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001. vol. 2, pp. II–II. IEEE (2001)
15. Govindu, V.M.: Lie-algebraic averaging for globally consistent motion estimation. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004. vol. 1, pp. I–I. IEEE (2004)
16. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, ISBN: 0521540518, second edn. (2004)
17. Hartley, R., Aftab, K., Trumpf, J.: L1 rotation averaging using the weiszfeld algorithm. In: CVPR 2011. pp. 3041–3048. IEEE (2011)

18. Jiang, N., Cui, Z., Tan, P.: A global linear method for camera pose registration. In: Proceedings of the IEEE international conference on computer vision. pp. 481–488 (2013)
19. Kasten, Y., Geifman, A., Galun, M., Basri, R.: Algebraic characterization of essential matrices and their averaging in multiview settings. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5895–5903 (2019)
20. Kasten, Y., Geifman, A., Galun, M., Basri, R.: Gpsfm: Global projective sfm using algebraic constraints on multi-view fundamental matrices. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3264–3272 (2019)
21. Kennedy, R., Daniilidis, K., Naroditsky, O., Taylor, C.J.: Identifying maximal rigid components in bearing-based localization. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 194–201. IEEE (2012)
22. Martinec, D., Pajdla, T.: Robust rotation and translation estimation in multiview reconstruction. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8. IEEE (2007)
23. McLachlan, G., Krishnan, T.: The EM algorithm and extensions. Wiley, 2. ed edn. (2008)
24. Moulon, P., Monasse, P., Marlet, R.: Global fusion of relative motions for robust, accurate and scalable structure from motion. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3248–3255 (2013)
25. Ozyesil, O., Singer, A.: Robust camera location estimation by convex programming. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2674–2683 (2015)
26. Ozyesil, O., Singer, A., Basri, R.: Stable camera motion estimation using convex programming. SIAM Journal on Imaging Sciences **8**(2), 1220–1262 (2015)
27. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4104–4113 (2016)
28. Shi, Y., Lerman, G.: Message passing least squares framework and its application to rotation synchronization. In: International Conference on Machine Learning. pp. 8796–8806. PMLR (2020)
29. Sidhartha, C., Govindu, V.M.: It is all in the weights: Robust rotation averaging revisited. In: 2021 International Conference on 3D Vision (3DV). pp. 1134–1143. IEEE (2021)
30. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: exploring photo collections in 3d. In: ACM siggraph 2006 papers, pp. 835–846 (2006)
31. Sweeney, C., Hollerer, T., Turk, M.: Theia: A fast and scalable structure-from-motion library. In: Proceedings of the 23rd ACM international conference on Multimedia. pp. 693–696 (2015)
32. Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustment: A modern synthesis. In: International workshop on vision algorithms. pp. 298–372. Springer (1999)
33. Tron, R., Vidal, R.: Distributed image-based 3-d localization of camera sensor networks. In: Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference. pp. 901–908. IEEE (2009)
34. Wilson, K., Snavely, N.: Robust global translations with 1dsfm. In: European Conference on Computer Vision. pp. 61–75. Springer (2014)
35. Wu, C., et al.: Visualsfm: A visual structure from motion system (2011)

36. Zhuang, B., Cheong, L.F., Lee, G.H.: Baseline desensitizing in translation averaging. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4539–4547 (2018)