Fusing Directions and Displacements in Translation Averaging

Lalit Manam and Venu Madhav Govindu
Indian Institute of Science
Bengaluru, India - 560012
{lalitmanam,venug}@iisc.ac.in

Abstract

Translation averaging solves for 3D camera translations given many pairwise relative translation directions. The mismatch between inputs (directions) and output estimates (absolute translations) makes translation averaging a challenging problem, which is often addressed by comparing either directions or displacements using relaxed cost functions that are relatively easy to optimize. However, the distinctly different nature of the cost functions leads to varied behaviour under different baselines and noise conditions. In this paper, we argue that translation averaging can benefit from a fusion of the two approaches. Specifically, we recursively fuse the individual updates suggested by direction and displacement-based methods using their uncertainties. The uncertainty of each estimate is modelled by the inverse of the Hessian of the corresponding optimization problem. As a result, our method utilizes the advantages of both methods in a principled manner. The superiority of our translation averaging scheme is demonstrated via the improved accuracies of camera translations on benchmark datasets compared to the state-of-the-art methods.

1. Introduction

Global methods for Structure-from-Motion (SfM) use a number of pairwise motion estimates to solve for the absolute motions of individual cameras. In this context of global SfM, the absolute motions can be represented as nodes of a viewgraph where the pairwise relative motions form the edges. Since there is an inherent scale ambiguity for pairwise relative translation (epipolar geometry [22]), the problem of translation averaging is one of estimating camera translations given the relative translation directions for individual viewgraph edges. Such schemes fall under the class of motion averaging methods [18, 19], which, unlike incremental methods [34, 38, 43], jointly solve for all the cameras at once. The dissimilarity between the input measurement space (directions) and the output solution space (camera translations) makes translation averaging a challenging problem since this involves the estimation of translation scales. Moreover, the existence of a unique solution is determined by the non-trivial issue of parallel rigidity [3, 33].

Translation Averaging Formulation: Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a viewgraph, where $\mathcal{V}$ and $\mathcal{E}$ denote the set of nodes and edges in $\mathcal{G}$, respectively. Each node i represents an absolute 3D rotation $\mathbf{R}_i \in \mathbb{SO}(3)$ and a translation $\mathbf{T}_i \in \mathbb{R}^3$ denoting its motion with respect to a global frame of reference. Each edge $(i, j) \in \mathcal{E}$ denotes the relative rotation and translation direction $(\mathbf{R}_{ij}, \mathbf{t}_{ij})$ between camera nodes i and j, where $\mathbf{t}_{ij} \in \mathbb{S}^2$. These result in the following relationships:

$$\mathbf{R}_{ij} = \mathbf{R}_j \mathbf{R}_i^{-1} \tag{1}$$

$$\mathbf{t}_{ij} = \frac{\mathbf{R}_j(\mathbf{T}_i - \mathbf{T}_j)}{\|\mathbf{R}_j(\mathbf{T}_i - \mathbf{T}_j)\|} \tag{2}$$

$$\mathbf{v}_{ij} = -\mathbf{R}_j^{-1}\mathbf{t}_{ij} = \frac{\mathbf{T}_j - \mathbf{T}_i}{\|\mathbf{T}_j - \mathbf{T}_i\|}. \tag{3}$$

The unit vector $\mathbf{v}_{ij}$ is the translation direction represented in the global frame of reference. In this paper, we assume that the rotations $\mathbf{R}_i$ for $i \in \mathcal{V}$ are either known or estimated using rotation averaging [5, 19, 21]. Given a set of directions $\{\mathbf{v}_{ij}\}$, the translation averaging problem is defined as the estimation of absolute translations of N cameras, i.e. $\mathbb{T} = \{\mathbf{T}_1, \cdots, \mathbf{T}_N\}$.

Given that the inputs are relative directions, the objective function for translation averaging can compare either (1) relative directions, obtained from estimated absolute translations, to that of the observed input directions or (2) relative displacements, obtained from estimated absolute translations, to that of the input directions with appropriately scaled magnitudes as a substitute for observed displacements.

Direction and Displacement Costs: A displacement-based cost compares the relative displacements between the cameras. It can be written as

$$e_{dis}(\mathbb{T}) = \sum_{(i,j)\in\mathcal{E}} \rho_{dis}\left(\|\mathbf{T}_j - \mathbf{T}_i - \|\mathbf{T}_j - \mathbf{T}_i\|\mathbf{v}_{ij}\|\right), \tag{4}$$

where $\rho(.)$ denotes a robust loss function, and its subscript represents a choice of loss function for the specific cost. Alternately, one can choose a direction-based cost, which compares the observed heading direction to that of the equivalent computed from absolute translations [42]. It can be written as

$$e_{dir}(\mathbb{T}) = \sum_{(i,j)\in\mathcal{E}} \rho_{dir}\left(\left\|\frac{\mathbf{T}_j - \mathbf{T}_i}{\|\mathbf{T}_j - \mathbf{T}_i\|} - \mathbf{v}_{ij}\right\|\right). \tag{5}$$

These costs are still non-convex, and converging to a good local minimum is still not straightforward.

Relaxed Costs: Both the costs in Eqns. 4 and 5 can be relaxed by introducing slack variables for each edge $(i,j) \in \mathcal{E}$. For the displacement-based cost (Eqn. 4), one can have non-negative slack variables, $\lambda_{ij} \geq 0$, for the magnitude of the input relative displacement [32, 41]. The *relaxed displacement cost* can be written as

$$e_{rdis}(\mathbb{T},\Lambda) = \sum_{(i,j)\in\mathcal{E}} \rho_{rdis}\left(\|\mathbf{T}_j - \mathbf{T}_i - \lambda_{ij}\mathbf{v}_{ij}\|\right), \tag{6}$$

where Λ is the set of slack variables $\{\lambda_{ij}|(i,j)\in\mathcal{E}\}$. This makes the residual terms linear in the relaxed displacement-based cost. For the direction-based cost (Eqn. 5), the normalization factor for converting relative displacements (from estimated absolute translations) to relative directions can be relaxed with non-negative slack variables, $\gamma_{ij} \geq 0$ [45]. The *relaxed direction-based cost* can be written as

$$e_{rdir}(\mathbb{T},\Gamma) = \sum_{(i,j)\in\mathcal{E}} \rho_{rdir}\left(\|(\mathbf{T}_j - \mathbf{T}_i)\,\gamma_{ij} - \mathbf{v}_{ij}\|\right), \tag{7}$$

where Γ is the set of slack variables $\{\gamma_{ij}|(i,j)\in\mathcal{E}\}$. This leads to bilinear residual terms for the relaxed direction-based cost. Such relaxations to the costs lead to efficient optimization routines. We note that λ_{ij} and γ_{ij} should ideally be equal to the baseline and inverse baseline between camera pairs, respectively.

Although the relaxed costs make them relatively easy to optimize, this does not necessarily mean that the performance of relaxed costs is similar in the presence of noise. The optimal solutions of the relaxed costs can be significantly different depending on the spread of the absolute translations and the noise in the input translation directions. In SfM, baselines between the camera pairs vary a lot and have different noise levels [15], resulting in varied behaviour of the relaxed cost functions.

Since the direction and displacement-based costs capture different attributes of the translation averaging problem, we seek to utilize both of them. To this end, in this paper, we propose a principled approach by recursively fusing the estimates of both costs based on uncertainty. Such an approach allows us to improve the quality of translation estimates, which is demonstrated for synthetic and real datasets.

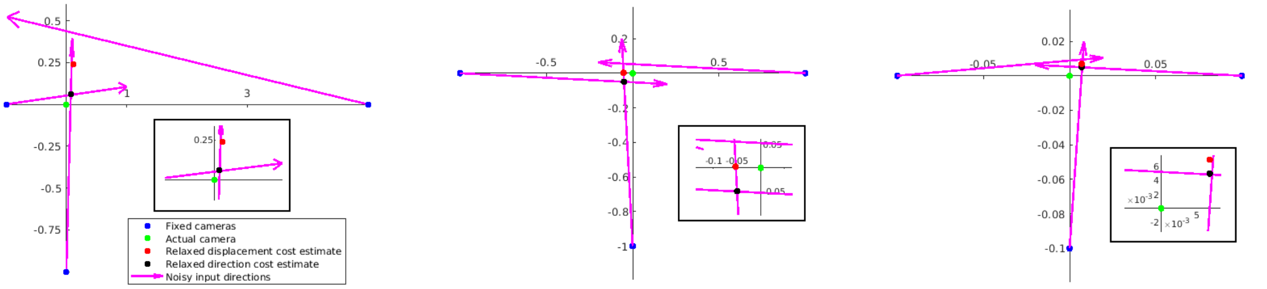
2. Literature Review

2.1. Rotation Averaging

Translation averaging requires input translation directions in the global frame of reference, which can be obtained using known absolute rotations. This is generally obtained using rotation averaging, which can be classified into intrinsic and extrinsic methods. Intrinsic methods, like [5, 6, 19, 21], solve the problem while optimizing directly on the rotation group $\mathbb{SO}(3)$. Extrinsic methods, like [7, 16, 30] solve a relaxed problem. Some recent developments can be found in [12, 36, 37] and the references therein.

2.2. Translation Averaging

Govindu [18] solved the translation averaging problem by minimizing the cross-product between the observed directions and the estimated relative camera translations. Arie-Nachimson *et al.* [2] sets up a linear system of cross-product constraints using epipolar geometry. Moulon *et al.* [31] converted the problem from aligning pairs to triplets by formulating a trifocal tensor with known rotations. In a similar spirit, Jiang *et al.* [24] used camera triplets and formulated the problem based on the constraints on a triangle. 1DSfM [42], proposed by Wilson *et al.*, used the direction-based cost and added camera-to-point constraints to stabilize the problem. A pre-processing step was incorporated to remove outliers. The relaxed displacement-based cost was used by Tron *et al.* [41] and was solved in a distributed manner. Least Unsquared Deviations (LUD) [32], proposed by Ozyesil *et al.*, used L_1 loss for robustness on the relaxed displacement-based cost. Arrigoni *et al.* [4] minimized the squared error of the orthogonal projection of the estimated relative translations onto observed directions. ShapeFit/ShapeKick [17], proposed by Goldstein *et al.*, minimized the orthogonal projection using ADMM but with an L_1 loss for robustness. Methods, like Cui *et al.* [10] and Cui *et al.* [8], used feature tracks to estimate the baseline scales and then solved a linear system. To avoid the influence of outliers, multiple estimates of baseline scales from different tracks were handled carefully. Bilinear Angle-based Translation Averaging (BATA) [45], proposed by Zhuang *et al.*, used the relaxed direction-based cost. Manam *et al.* [28] proposed to improve the input trans-



(a) Disparate baselines $\eta = \{3^\circ, 4^\circ, 5^\circ\}$, $bd = \{1, 1, 5\}$ units

(b) Similar baselines $\eta = \{3^\circ, 3^\circ, 3^\circ\}$, $bd = \{1, 1, 1\}$ units

(c) Similar baselines $\eta = \{5^\circ, 4^\circ, 3^\circ\}$, $bd = \{0.1, 0.1, 0.1\}$ units

Figure 1. Toy examples showing the behaviour of relaxed costs under noise. η denotes the perturbations to get the noisy input directions, and bd denotes baseline distances. The zoomed part of the region containing the solutions is shown in the inset box. No cost performs the best in all the scenarios.

lation directions by guiding the weights on the point correspondences in the edges based on the translation estimates and iteratively reestimating the directions. Some methods removed severely corrupted directions [35] or sensitive directions [29] to obtain reliable translation estimates. Other approaches which solved absolute translations include similarity averaging [9], averaging essential and fundamental matrices [25, 26], and utilizing the properties of the matrix generated from pairwise camera displacements [13].

3. Motivation

Firstly, we look at the non-relaxed costs in a least squares sense, i.e. $\rho_{dis}(x) = x^2$ and $\rho_{dir}(x) = x^2$. Then the displacement-based cost (Eqn. 4) can be written as

$$\begin{aligned} e_{dis,ls}(\mathbb{T}) &= \sum_{(i,j)\in\mathcal{E}} (\|\mathbf{T}_j - \mathbf{T}_i - \|\mathbf{T}_j - \mathbf{T}_i\|\mathbf{v}_{ij}\|)^2 \\ &= \sum_{(i,j)\in\mathcal{E}} \|\mathbf{T}_j - \mathbf{T}_i\|^2 \left\| \frac{\mathbf{T}_j - \mathbf{T}_i}{\|\mathbf{T}_j - \mathbf{T}_i\|} - \mathbf{v}_{ij} \right\|^2, \end{aligned} \tag{8}$$

and the direction-based cost (Eqn. 5) as

$$e_{dir,ls}(\mathbb{T}) = \sum_{(i,j)\in\mathcal{E}} \left\| \frac{\mathbf{T}_j - \mathbf{T}_i}{\|\mathbf{T}_j - \mathbf{T}_i\|} - \mathbf{v}_{ij} \right\|^2. \tag{9}$$

It can be clearly seen that each residual term in Eqn. 8 is a weighted residual term in Eqn. 9, where the weights are the squared baseline distance of the edge. This makes displacement-based cost sensitive to baselines, whereas direction-based cost does not take into account the baselines. This observation is demonstrated empirically in [45]. Hence, the direction-based cost is preferred instead of the displacement-based cost.

Now let us consider the relaxed versions of these costs (Eqns. 6 and 7). For a given $\mathbb{T}$, the optimal values of the non-negative slack variables [45] are given as

$$\lambda_{ij} = \max\left(\frac{\langle \mathbf{T}_j - \mathbf{T}_i, \mathbf{v}_{ij} \rangle}{\|\mathbf{v}_{ij}\|^2}, 0 \right), \forall (i,j) \in \mathcal{E}, \tag{10}$$

$$\gamma_{ij} = \max\left(\frac{\langle \mathbf{T}_j - \mathbf{T}_i, \mathbf{v}_{ij} \rangle}{\|\mathbf{T}_j - \mathbf{T}_i\|^2}, 0 \right), \forall (i,j) \in \mathcal{E}. \tag{11}$$

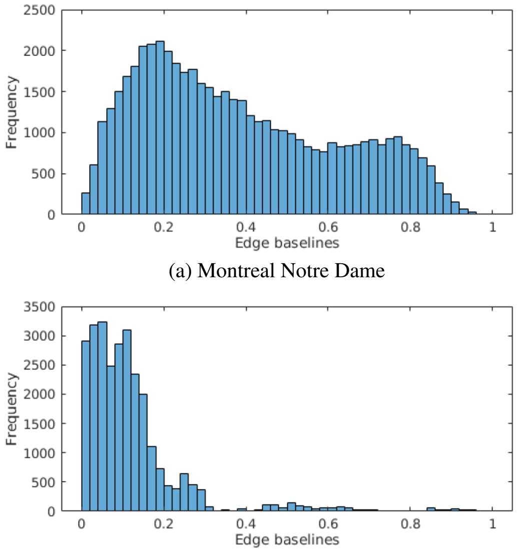
Let $\mathbf{T}_{ij} = \mathbf{T}_j - \mathbf{T}_i$. Since $\mathbf{v}_{ij}$ denotes the direction, $\|\mathbf{v}_{ij}\| = 1$. Then from Eqns. 10 and 11, for positive values of slack variables, we get

$$\frac{\lambda_{ij}}{\|\mathbf{T}_{ij}\|} = \langle \hat{\mathbf{T}}_{ij}, \mathbf{v}_{ij} \rangle, \tag{12}$$

$$\gamma_{ij} \cdot \|\mathbf{T}_{ij}\| = \langle \hat{\mathbf{T}}_{ij}, \mathbf{v}_{ij} \rangle, \tag{13}$$

where $\hat{\mathbf{a}}$ denotes the direction along the vector $\mathbf{a}$. The left-hand side of Eqns. 12 and 13 (ideally equal to 1) show the closeness of slack variables, λ_{ij} and γ_{ij}, to the baselines and inverse baselines, respectively. It is dependent on how closely the estimated relative directions, obtained from absolute translations, align to the observed input directions.

To understand the behaviour of the relaxed costs, we consider toy examples which have disparate and similar baselines with different levels of small perturbation to the input directions. Again, for a similar analysis, we choose $\rho_{rdis}(x) = x^2$ and $\rho_{rdir}(x) = x^2$. In these examples, three cameras are fixed, and the translation of the fourth camera is estimated, given directions from the other three. We use Eqns. 10 and 11 in the relaxed costs (Eqns. 6 and 7 respectively) and perform a grid search to find their optimal solutions.



(a) Montreal Notre Dame

(b) Yorkminster

Figure 2. Ground truth baseline distances of two datasets. Baselines are normalized to have a maximum value of 1 since its relative spread is of interest.

Fig. 1 shows the difference in the solutions obtained with the relaxed cost under different baseline conditions. It can be seen that, in Fig. 1a, when the baselines are disparate, relaxed direction-based cost performs better. We observed this trend for all perturbations of input, which are not shown here due to space constraints. However, when the baselines are similar, relaxed displacement-based cost performs better for the example in Fig. 1b and vice versa in Fig. 1c. This shows that the performance of both the cost functions is dependent on the distribution of baselines and noise conditions. In SfM, the camera translations are spread nonuniformly, where the baselines are similar for a subset of edges and disparate for another subset. This can be seen in Fig. 2, which shows the histogram of ground truth baselines on two 1DSfM datasets [42]. Also, the noise levels are not known apriori. The resulting different behaviour of the relaxed costs suggests that we should seek to combine them in a principled fashion.

4. Proposed Method

4.1. Constraint Set

The solution to the translation averaging problem is defined up to a *global scale* and a *choice of origin* due to the mismatch in input space and output space. To fix the origin ambiguity, we constrain the problem such that the centroid of the estimated absolute translations is zero. The choice of constraint for fixing the global scale has a huge impact on the quality of the solution, which is discussed very well in [45]. It shows that inequality constraint on scales can have a shrinkage effect, and equality constraint mitigates such unintended effects. So, we use the equality-based dot product constraint. Thus, the constraint set, consisting of linear equality constraints, is defined as

$$\mathcal{C} = \left\{ \mathbb{T} \;\middle|\; \frac{1}{|\mathcal{E}|} \sum_{(i,j)\in\mathcal{E}} \langle \mathbf{T}_j - \mathbf{T}_i, \mathbf{v}_{ij} \rangle = 1, \sum_{i\in\mathcal{V}} \mathbf{T}_i = \mathbf{0} \right\}. \tag{14}$$

LUD [32] uses relaxed displacement-based cost with inequality constraints. Please refer [32] for details. BATA [45] uses relaxed direction-based cost with the constraint set $\mathcal{C}$. [45] also defines Revised LUD or RLUD, which uses relaxed displacement-based cost with the constraint set $\mathcal{C}$. For the rest of the paper, LUD refers to the original formulation defined in [32], and RLUD refers to the modification given in [45]. In our method, we use the constraint set $\mathcal{C}$.

4.2. Fused Translation Averaging

Our aim is to encapsulate the benefits of both the cost functions to solve the problem. One way to capture the properties of the cost functions is through the uncertainties of the estimates. It can be obtained through the Hessians of the costs without knowing the ground truth. A simple way to go forward is to consider the uncertainty in the estimates from the two cost functions to be Gaussian distributed. Firstly, due to the dissimilarity of input and output space, the input noise cannot be directly related to the output uncertainty. Moreover, input translation directions contain outliers, which makes the Gaussian assumption unfit for practical purposes. To handle the problem of outliers, we use the IRLS [20, 23] framework. The IRLS approach weights each residual term in the cost function and then solves the resulting cost as a weighted least squares problem iteratively. For our specific case of the relaxed costs, the weighted least squares can be written as

$$e_{rdis,w}(\mathbb{T}) = \sum_{(i,j)\in\mathcal{E}} w_{ij}^{rdis} \|\mathbf{T}_j - \mathbf{T}_i - \lambda_{ij}\mathbf{v}_{ij}\|^2, \tag{15}$$

$$e_{rdir,w}(\mathbb{T}) = \sum_{(i,j)\in\mathcal{E}} w_{ij}^{rdir} \left\|(\mathbf{T}_j - \mathbf{T}_i)\,\gamma_{ij} - \mathbf{v}_{ij}\right\|^2, \tag{16}$$

where w_{ij} denotes the weights obtained in the IRLS step corresponding to a robust loss. We follow [32, 45] by recomputing the slack variables in every iteration using Eqns. 10 and 11 and fix them for the weighted least squares problem. For reasonable weights w_{ij}, the influence of outliers in $e_{rdis,w}$ and $e_{rdir,w}$ reduces. Given that the weights w_{ij} are non-negative, the Hessians of Eqns. 15 and 16 are weighted Graph Laplacians with w_{ij}'s as edge weights, making Eqns. 15 and 16 convex quadratic. This leads us to a reasonable assumption that the translations $\mathbb{T}$ obtained from $e_{rdis,w}$ and $e_{rdir,w}$ are distributed as Gaussian $\mathcal{N}(\mathbb{T}_{rdis,w}, \Sigma_{rdis,w}(\mathbb{T}_{rdis,w}))$ and $\mathcal{N}(\mathbb{T}_{rdir,w}, \Sigma_{rdir,w}(\mathbb{T}_{rdir,w}))$, respectively, where the subscript of $\mathbb{T}$ signifies the solution from the respective cost and the subscript of $\Sigma(\cdot)$ denotes their individual covariances computed at the specific value of translations. It is known that with Gaussian assumption, the Hessian of the cost is the same as the inverse covariance (Σ^{-1}) [44]. For the specific case of translation averaging with Eqns. 15 and 16, the Hessian is the Laplacian of a weighted graph, which is singular. However, we can resolve the gauge ambiguity by fixing the node with the highest degree as the origin for our coordinate system and translating the solutions accordingly. Also, since the translations are solved with the equality-based constraint in Eqn. 14, the Hessians, which are dependent on the translations, also are in the same global scale, thus taking care of the global scale ambiguity.

In each iteration of IRLS, we consider the previous estimate as a prior. The solutions suggested by $e_{rdis,w}$ and $e_{rdir,w}$ are the maximum-likelihood estimate based on the observations (input directions). We compute the Maximum A Posteriori (MAP) estimate using the previous estimate and the suggested solutions. This is similar in spirit to the Bayes recursive filter [40], where the uncertainties are updated with every time step. Since translations obtained from $e_{rdis,w}$ and $e_{rdir,w}$ are distributed as Gaussians, we can easily compute the MAP estimate. Given two estimates $\mathbb{T}_1$ and $\mathbb{T}_2$, distributed as Gaussians, with covariances Σ_1 and Σ_2, the MAP estimate, $\mathbb{T}_{MAP}$, is given as

$$\begin{aligned} &\mathbb{T}_{MAP}\left(\Sigma_1, \mathbb{T}_1, \Sigma_2, \mathbb{T}_2\right) \\ &= \left(\Sigma_1^{-1} + \Sigma_2^{-1}\right)^{-1} \left(\Sigma_1^{-1}\mathbb{T}_1 + \Sigma_2^{-1}\mathbb{T}_2\right). \end{aligned} \tag{17}$$

To leverage the uncertainty information of each cost function, we compute the MAP estimate using $e_{rdis,w}$ and then using $e_{rdir,w}$ in each IRLS iteration. This is somewhat reminiscent of Multi-Objective Optimization (MOO) [11, 14], where multiple cost functions are simultaneously optimized. However, our aim here is to find a single solution instead of multiple solutions, as in MOO. This allows us to use the advantages of both the costs in the IRLS iteration itself. We empirically found that using uncertainty only from one cost function degrades the solution, which is discussed in the next section. The complete scheme is summarized in Algo. 1. We provide implementation details of Algo. 1 in the supplementary material.

5. 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 provided by [6][1]. For all experiments, the maximal parallel rigid component of the viewgraph is extracted based on [27]. We use LUD [32] implemented in Theia [39]. 1DSfM [42] and BATA's [45] code provided by the respective authors[2,3]. The implementation of ShapeFit [17] is not publicly available, and hence we implement it using ADMM as suggested in the paper. 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 [42]. 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**.

Algorithm 1: Fused Translation Averaging (Fused-TA)

```
Initialize global translations $\mathbb{T}$
while not converged do
    Update scales $\Lambda$ and $\Gamma$ using Eqns. 10 and 11.
    Update weights for robustness, $w_{ij}$, for $e_{rdis}$.
    Compute translations, $\mathbb{T}_{rdis,w}$, by solving
      Eqn.15 ($e_{rdis,w}$) with fixed $\Lambda$.
    Get the covariances $\Sigma_{rdis,w}(\mathbb{T}_{rdis,w})$ and
      $\Sigma_{rdis,w}(\mathbb{T})$.
    Update translations $\mathbb{T} \leftarrow$
      $\mathbb{T}_{MAP}\left(\Sigma_{rdis,w}(\mathbb{T}_{rdis,w}), \mathbb{T}_{rdis,w}, \Sigma_{rdis,w}(\mathbb{T}), \mathbb{T}\right)$
      using Eqn. 17.
    Update weights for robustness, $w_{ij}$, for $e_{rdir}$.
    Compute translations, $\mathbb{T}_{rdir,w}$, by solving
      Eqn.16 ($e_{rdir,w}$) with fixed $\Gamma$.
    Get the covariances $\Sigma_{rdir,w}(\mathbb{T}_{rdir,w})$ and
      $\Sigma_{rdir,wls}(\mathbb{T})$.
    Update translations $\mathbb{T} \leftarrow$
      $\mathbb{T}_{MAP}\left(\Sigma_{rdir,w}(\mathbb{T}_{rdir,w}), \mathbb{T}_{rdir,w}, \Sigma_{rdir,w}(\mathbb{T}), \mathbb{T}\right)$
      using Eqn. 17.
end
```

5.1. Synthetic Data

We carry out experiments with synthetic data to study the comparative behaviour of different methods in the presence of noise. We check the behaviour of RLUD, BATA, and our method, denoted as Fused-TA. We first create a synthetic dataset which has disparate baselines. We sample 50 points uniformly from two unit spheres, which are 5 units apart. We consider these points as the ground truth camera translations. Then, in each sphere, we check

[1] https://ee.iisc.ac.in/cvlab/research/rotaveraging/

[2] https://github.com/wilsonkl/SfM_Init

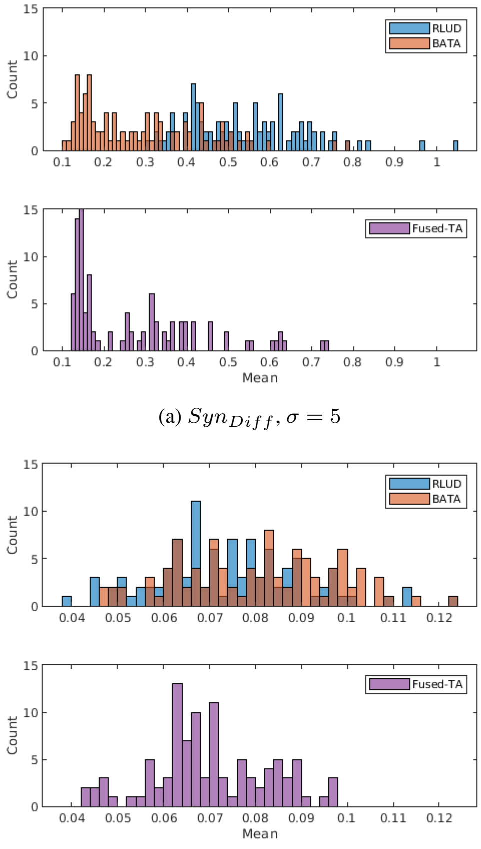
[3] https://bbzh.github.io/document/BATA.zip

Dataset	RLUD	BATA	Fused-TA (Ours)
		Mean Errors	
$Syn_{Diff}, \sigma = 2$	3.96	1.34	**0.81**
$Syn_{Diff}, \sigma = 5$	5.41	2.87	**2.70**
$Syn_{Sim}, \sigma = 2$	0.61	0.42	**0.36**
$Syn_{Sim}, \sigma = 5$	0.74	0.80	**0.70**
		Median Errors	
$Syn_{Diff}, \sigma = 2$	2.81	0.91	**0.72**
$Syn_{Diff}, \sigma = 5$	4.42	**2.08**	**2.08**
$Syn_{Sim}, \sigma = 2$	0.54	0.36	**0.32**
$Syn_{Sim}, \sigma = 5$	0.67	0.73	**0.64**

Table 1. Camera translation errors ($\times 10^{-1}$) averaged over 100 trials for each synthetic dataset. Units are specified by the ground truth synthetic data.

for the four nearest neighbours of every point and then add edges between them. This creates edges with similar baselines. Then, we add 100 edges across the two spheres, which have much larger baselines than those created within the spheres. We call this dataset Syn_{Diff}. Now, we add noise to the edges, similar to [45], as follows: we perturb the ground truth relative translations $\mathbf{v}_{ij}^{gt}$ by multiplying them with a rotation matrix $\mathbf{R}(\mathbf{n}_{ij}^{u}, \sigma\theta_{ij})$. $\mathbf{n}_{ij}^{u}$ is the axis uniformly sampled from the orthogonal complement of $\mathbf{v}_{ij}^{gt}$ and $\sigma\theta_{ij}$ is the rotation angle. θ_{ij} is drawn from Gaussian $\mathcal{N}(0, 1)$ and σ is the standard deviation of noise. We take $\sigma \in \{2, 5\}$ degrees. Next, we generate another dataset where the baselines are similar. We sample 100 points uniformly from a unit sphere and perform a similar procedure as done for Syn_{Diff} to create noisy datasets. We call this dataset Syn_{Sim}. We generate ten instances for each noise level for both datasets. We evaluate the performance by checking the mean and median errors for the translation estimates. For each instance of the graphs, we run 10 trials of all the methods.

Table 1 shows the absolute translation errors for the synthetic datasets. It can be seen that for Syn_{Diff}, which has disparate baselines, BATA performs better than RLUD. For Syn_{Sim}, which has similar baselines, BATA performs better than RLUD for $\sigma = 2$, but RLUD performs better than BATA for $\sigma = 5$. This shows that neither e_{rdis} nor e_{rdir} perform consistently under different baseline and noise conditions. For both Syn_{Diff} and Syn_{Sim}, Fused-TA performs the best. This reveals that the fusion leads to better translation estimates under different baseline and noise conditions. Fig. 3 shows the histogram of mean errors obtained from 100 runs for each noise level on both the datasets. For Syn_{Diff} with $\sigma = 5$, mean errors for most instances for Fused-TA are lower than BATA, followed by RLUD. For Syn_{Sim} with $\sigma = 5$, RLUD has low mean errors for most of the instances compared to BATA, and Fused-TA performs better than RLUD. This shows that using uncertainty information from both costs improves translation estimates.



(a) $Syn_{Diff}, \sigma = 5$

(b) $Syn_{Sim}, \sigma = 5$

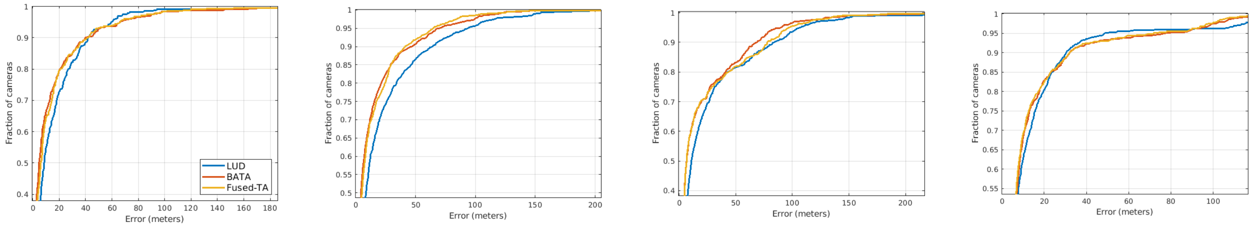
Figure 3. Histogram of mean errors for the two synthetic datasets with $\sigma = 5$ noise. The leftward shift indicates the superior performance of our method.

5.2. Real Data

We present results on real unordered image datasets provided by the authors of 1DSfM [42]. We use COLMAP [34] to generate pairwise relative rotations and translations in 1DSfM datasets and align COLMAP's solution to 1DSfM provided ground truth to get absolute translations in meters. Then, the data is pre-processed in a manner similar to that suggested in [32]: 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 in a RANSAC loop, where the relative rotations obtained from the rotation-averaged solution are used. The results for our method Fused-TA are shown in Table 2 along with other state-of-the-art methods. We use COLMAP's solution as the reference for performance comparison in this setting. It can be seen from Table 2 that translations improve using our method (Fused-TA), with Fused-TA performing the overall best. Specifically, the median errors of the camera

Dataset	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	Mean Errors					Median Errors				
			1DSfM [42]	LUD [32]	ShapeFit [17]	BATA [45]	Fused-TA (Ours)	1DSfM [42]	LUD [32]	ShapeFit [17]	BATA [45]	Fused-TA (Ours)
Alamo (ALM)	1016	16831	8e3	7.6	35.3	7.5	**7.0**	**2.4**	3.0	2.6	**2.4**	**2.4**
Ellis Island (ELS)	908	10383	107.3	45.1	39.7	42.6	**27.3**	32.3	26.0	**15.6**	25.3	17.5
Gendarmenkmart (GMM)	1026	14175	3e4	**42.3**	51.3	43.6	44.4	27.5	**25.9**	32.3	26.3	27.7
Madrid Metropolis (MDR)	627	4941	2e4	17.5	170.8	**15.3**	15.7	5.4	8.7	5.5	**4.8**	6.1
Montreal Notre Dame (MND)	599	18390	1e3	**4.0**	4.3	4.5	4.3	1.9	2.4	2.3	1.9	**1.8**
Notre Dame (ND)	1421	70771	4e4	4.0	3.7	**3.3**	**3.3**	**1.2**	1.8	1.3	**1.2**	**1.2**
NYC Library (NYC)	858	8173	3e4	**6.4**	3e3	8.2	7.6	3.1	3.0	3.6	2.5	**2.3**
Piazza del Popolo (PDP)	1038	15182	6e4	8.0	19.1	8.1	**7.6**	7.2	5.5	15.8	5.0	**4.5**
Piccadilly (PIC)	3124	46754	3e4	**6.5**	25.2	7.0	6.7	2.2	2.6	2.4	2.3	**2.1**
Roman Forum (ROF)	1575	19593	5e5	22.4	33.5	15.5	**15.3**	4.3	8.6	8.6	**4.4**	5.3
Tower of London (TOL)	824	9457	8e4	28.2	58.0	**22.9**	24.5	8.0	10.7	8.6	6.9	**6.5**
Trafalgar (TFG)	7483	115052	4e4	21.4	33.3	22.4	**21.3**	11.7	7.6	8.3	6.9	**6.2**
Union Square (USQ)	1166	13460	2e4	**14.5**	23.0	17.1	31.7	9.2	8.5	14.3	8.1	**7.9**
Vienna Cathedral (VNC)	1647	27386	7e4	14.8	172.0	14.9	**14.3**	12.8	6.5	7.1	6.5	**6.1**
Yorkminster (YKM)	1834	19177	7e4	21.1	72.6	21.6	**20.7**	22.9	12.9	15.8	13.2	**13.0**

Table 2. Camera translation errors (in meters) on 1DSfM [42] datasets ($\lvert\mathcal{V}\rvert$: number of nodes, $\lvert\mathcal{E}\rvert$: number of edges).



(a) Madrid Metropolis (MDR)

(b) Roman Forum (ROF)

(c) Tower of London (TOL)

(d) Vienna Cathedral (VNC)

Figure 4. Zoomed part of empirical cumulative error distribution (in meters) for camera translations obtained on 1DSfM datasets.

translations are reduced for most of the datasets.

In Fig. 4, we present the zoomed part of empirical cumulative error distribution for camera translations, where we analyze the behaviour of LUD, BATA and our method, Fused-TA (please refer to the supplementary material for full figures). For MDR, 90% of the cameras are better with BATA and our method (Fused-TA), but LUD performs better for the remaining 10% of the cameras. In the case of ROF, BATA performs slightly better than our method, up to 85% of the cameras, but the trend reverses for the remaining cameras. For TOL, our method performs better than BATA for up to 70% of the cameras, but BATA estimates the remaining cameras in a better fashion. For VNC, our method performs best up to 85% of the cameras, and LUD's performance is better for most of the remaining ones. In all the cases, it can be seen that our method tries to incorporate the benefits from both the costs and, in many datasets, the high errors in translations for our method lie between LUD and BATA.

Now, we study the importance of taking MAP estimate using both the cost functions in Algo. 1. We check for two cases, one where we consider only relaxed displacement-based cost (e_{rdis}) by removing steps 8-11 in Algo. 1 and in the other, we consider only relaxed direction-based cost (e_{rdir}) by removing steps 4-7 in Algo. 1. We compare the performance of the two cases with the complete procedure in Algo. 1 in Table 3. It can be seen that for most of the datasets, using only either of the cost functions for MAP estimate leads to degradation in the performance compared to the full Algo. 1.

Next, we use the camera motions obtained in Table 2 and carry out triangulation using Theia [39] and check the reconstructions. At this stage, no form of bundle adjustment is employed. In Fig. 5, we visualize the 3D reconstructions obtained using our scheme. We also show the bundle-adjusted solution (obtained using the Ceres-Solver [1]) as a reference. In all these datasets, we see that triangulation based on our solution accurately recovers most of the reconstruction when compared with the bundle-adjusted solution as a reference.

Finally, we compare the computation time of different methods in Table 4. Owing to the differences between displacement and direction-based costs, the behaviour of RLUD and BATA are also different in each iteration. Our approach of uncertainty-based fusion attempts to reconcile these differences during each iteration. As a result, apart

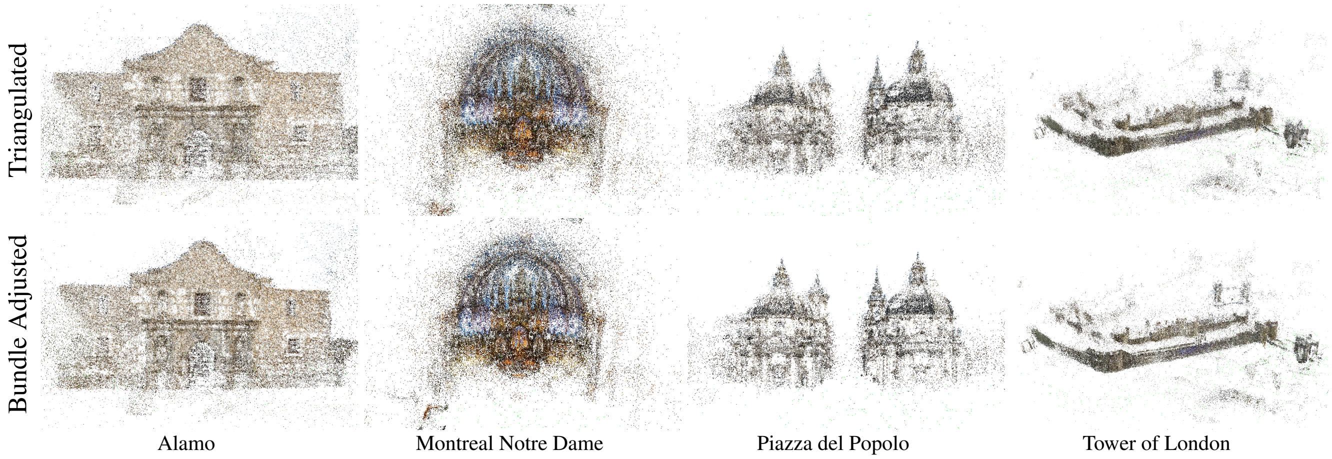


Figure 5. Reconstructions obtained with triangulation using our Fused-TA translation estimate (first row) compared to bundle adjustment (second row).

Dataset	LUD-only step (w/o steps 8-11)	Fused-TA (Full Algo. 1)	BATA-only step (w/o steps 4-7)	Fused-TA (Full Algo. 1)
	Mean Errors			
ALM	7.6	**7.0**	**7.0**	**7.0**
ELS	70.7	**27.3**	43.8	**27.3**
GMM	45.7	**44.4**	**42.2**	44.4
MDR	25.8	**15.7**	**15.1**	15.7
MND	**3.5**	4.3	**4.2**	4.3
NYC	**7.6**	**7.6**	**7.2**	7.6
ND	4.1	**3.3**	**3.3**	**3.3**
PDP	**7.5**	7.6	7.8	**7.6**
PIC	10.5	**6.7**	**6.7**	**6.7**
ROF	8e5	**15.3**	17.1	**15.3**
TOL	26.5	**24.5**	25.2	**24.5**
TFG	22.8	**21.3**	22.0	**21.3**
USQ	39.8	**31.7**	**14.8**	31.7
VNC	**14.0**	14.3	15.1	**14.3**
YKM	**20.3**	20.7	25.5	**20.7**
	Median Errors			
ALM	2.8	**2.4**	**2.4**	**2.4**
ELS	**12.1**	17.5	24.9	**17.5**
GMM	29.9	**27.7**	**25.1**	27.7
MDR	13.3	**6.1**	**5.6**	6.1
MND	1.9	**1.8**	**1.8**	**1.8**
NYC	2.8	**2.3**	2.6	**2.3**
ND	1.7	**1.2**	**1.2**	**1.2**
PDP	5.1	**4.5**	5.1	**4.5**
PIC	6.1	**2.1**	2.3	**2.1**
ROF	29.5	**5.3**	**5.3**	**5.3**
TOL	7.8	**6.5**	6.7	**6.5**
TFG	9.9	**6.2**	7.0	**6.2**
USQ	**4.9**	7.9	**7.8**	7.9
VNC	6.8	**6.1**	6.9	**6.1**
YKM	16.8	**13.0**	19.9	**13.0**

Table 3. Camera translation errors (in meters) only considering particular costs in Algo. 1 vs full Algo. 1. Entries marked in bold show improvement using a single cost vs full Algo. 1 and not comparing all variants.

Dataset	RLUD	BATA	Fused-TA
ALM	10	21	39
ELS	6	14	25
GMM	8	18	31
MDR	3	8	12
MND	8	21	34
ND	39	75	125
NYC	5	12	21
PDP	8	21	33
PIC	45	83	152
ROF	13	31	52
TOL	6	14	23
TFG	167	357	610
USQ	8	17	30
VNC	15	35	56
YKM	8	19	32

Table 4. Computation time (in seconds) for different schemes on 1DSfM datasets.

from the cost of computing both RLUD and BATA estimates in every iteration, our approach needs more iterations to converge than RLUD or BATA. The result is an increase in the overall computation time of Fused-TA.

6. Conclusion

In this paper, we discuss the relative behaviour of the relaxed direction and displacement-based cost functions in translation averaging. We argue that translation estimation can benefit from a careful consideration of the two approaches in a principled manner by recursively fusing the estimates based on their uncertainties. The merits of our proposed approach are detailed via experiments using both synthetic and real datasets.

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 the Science and Engineering Research Board, Department of Science and Technology, Government of India.

References

[1] Sameer Agarwal, Keir Mierle, and Others. Ceres solver. http://ceres-solver.org. 7

[2] Mica Arie-Nachimson, Shahar Z Kovalsky, Ira Kemelmacher-Shlizerman, Amit Singer, and Ronen Basri. Global motion estimation from point matches. In 2012 Second international conference on 3D imaging, modeling, processing, visualization & transmission, pages 81–88. IEEE, 2012. 2

[3] Federica Arrigoni and Andrea Fusiello. Bearing-based network localizability: a unifying view. IEEE transactions on pattern analysis and machine intelligence, 41(9):2049–2069, 2018. 1

[4] Federica Arrigoni, Beatrice Rossi, and Andrea Fusiello. Robust and efficient camera motion synchronization via matrix decomposition. In International Conference on Image Analysis and Processing, pages 444–455. Springer, 2015. 2

[5] Avishek Chatterjee and Venu Madhav Govindu. Efficient and robust large-scale rotation averaging. In Proceedings of the IEEE International Conference on Computer Vision, pages 521–528, 2013. 1, 2

[6] Avishek Chatterjee and Venu Madhav Govindu. Robust relative rotation averaging. IEEE transactions on pattern analysis and machine intelligence, 40(4):958–972, 2017. 2, 5

[7] David Crandall, Andrew Owens, Noah Snavely, and Dan Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. In CVPR 2011, pages 3001–3008. IEEE, 2011. 2

[8] Hainan Cui, Shuhan Shen, and Zhanyi Hu. Robust global translation averaging with feature tracks. In 2016 23rd International Conference on Pattern Recognition (ICPR), pages 3727–3732. IEEE, 2016. 2

[9] Zhaopeng Cui and Ping Tan. Global structure-from-motion by similarity averaging. In Proceedings of the IEEE International Conference on Computer Vision, pages 864–872, 2015. 3

[10] Zhaopeng Cui, Nianjuan Jiang, Chengzhou Tang, and Ping Tan. Linear global translation estimation with feature tracks. In Proc. ECCV, pages 61–75, 2014. 2

[11] Kalyanmoy Deb. Multi-objective optimisation using evolutionary algorithms: an introduction. In Multi-objective evolutionary optimisation for product design and manufacturing, pages 3–34. Springer, 2011. 5

[12] Frank Dellaert, David M. Rosen, Jing Wu, Robert E. Mahony, and Luca Carlone. Shonan rotation averaging: Global optimality by surfing so(p)$^{\mathrm{n}}$. In ECCV (6), pages 292–308. Springer, 2020. 2

[13] Qiulei Dong, Xiang Gao, Hainan Cui, and Zhanyi Hu. Robust camera translation estimation via rank enforcement. IEEE transactions on cybernetics, 2020. 3

[14] Michael Emmerich and André H Deutz. A tutorial on multiobjective optimization: fundamentals and evolutionary methods. Natural computing, 17(3):585–609, 2018. 5

[15] Olof Enqvist, Fredrik Kahl, and Carl Olsson. Non-sequential structure from motion. In 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), pages 264–271. IEEE, 2011. 2

[16] A. Eriksson, C. Olsson, F. Kahl, and T. Chin. Rotation averaging and strong duality. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 127–135, 2018. 2

[17] Thomas Goldstein, Paul Hand, Choongbum Lee, Vladislav Voroninski, and Stefano Soatto. Shapefit and shapekick for robust, scalable structure from motion. In European Conference on Computer Vision, pages 289–304. Springer, 2016. 2, 5, 7

[18] Venu Madhav Govindu. Combining two-view constraints for motion estimation. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, pages II–II. IEEE, 2001. 1, 2

[19] Venu Madhav Govindu. 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., pages I–I. IEEE, 2004. 1, 2

[20] Peter J Green. Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives. Journal of the Royal Statistical Society: Series B (Methodological), 46(2):149–170, 1984. 4

[21] Richard Hartley, Khurrum Aftab, and Jochen Trumpf. L1 rotation averaging using the weiszfeld algorithm. In CVPR 2011, pages 3041–3048. IEEE, 2011. 1, 2

[22] R. I. Hartley and A. Zisserman. Multiple View Geometry in Computer Vision. Cambridge University Press, ISBN: 0521540518, second edition, 2004. 1

[23] Peter J Huber. Robust statistics. John Wiley & Sons, 2004. 4

[24] Nianjuan Jiang, Zhaopeng Cui, and Ping Tan. A global linear method for camera pose registration. In Proceedings of the IEEE international conference on computer vision, pages 481–488, 2013. 2

[25] Yoni Kasten, Amnon Geifman, Meirav Galun, and Ronen Basri. Algebraic characterization of essential matrices and their averaging in multiview settings. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 5895–5903, 2019. 3

[26] Yoni Kasten, Amnon Geifman, Meirav Galun, and Ronen Basri. 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, pages 3264–3272, 2019. 3

[27] Ryan Kennedy, Kostas Daniilidis, Oleg Naroditsky, and Camillo J Taylor. Identifying maximal rigid components in bearing-based localization. In 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 194–201. IEEE, 2012. 5

[28] Lalit Manam and Venu Madhav Govindu. Correspondence reweighted translation averaging. In European Conference on Computer Vision, Proceedings, Part XXXIII, pages 56–72. Springer, 2022. 2

[29] Lalit Manam and Venu Madhav Govindu. Sensitivity in translation averaging. In Thirty-seventh Conference on Neural Information Processing Systems, 2023. 3

[30] Daniel Martinec and Tomas Pajdla. Robust rotation and translation estimation in multiview reconstruction. In 2007 IEEE Conference on Computer Vision and Pattern Recognition, pages 1–8. IEEE, 2007. 2

[31] Pierre Moulon, Pascal Monasse, and Renaud Marlet. Global fusion of relative motions for robust, accurate and scalable structure from motion. In Proceedings of the IEEE International Conference on Computer Vision, pages 3248–3255, 2013. 2

[32] Onur Ozyesil and Amit Singer. Robust camera location estimation by convex programming. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2674–2683, 2015. 2, 4, 5, 6, 7

[33] Onur Ozyesil, Amit Singer, and Ronen Basri. Stable camera motion estimation using convex programming. SIAM Journal on Imaging Sciences, 8(2):1220–1262, 2015. 1

[34] Johannes Lutz Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4104–4113, 2016. 1, 6

[35] Yunpeng Shi and Gilad Lerman. Estimation of camera locations in highly corrupted scenarios: All about that base, no shape trouble. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2868–2876, 2018. 3

[36] Yunpeng Shi and Gilad Lerman. Message passing least squares framework and its application to rotation synchronization. In International Conference on Machine Learning, pages 8796–8806. PMLR, 2020. 2

[37] Chitturi Sidhartha and Venu Madhav Govindu. It is all in the weights: Robust rotation averaging revisited. In 2021 International Conference on 3D Vision (3DV), pages 1134–1143. IEEE, 2021. 2

[38] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In ACM siggraph 2006 papers, pages 835–846. 2006. 1

[39] Christopher Sweeney, Tobias Hollerer, and Matthew Turk. Theia: A fast and scalable structure-from-motion library. In Proceedings of the 23rd ACM international conference on Multimedia, pages 693–696, 2015. 5, 7

[40] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. Probabilistic Robotics (Intelligent Robotics and Autonomous Agents). The MIT Press, 2005. 5

[41] Roberto Tron and René Vidal. 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, pages 901–908. IEEE, 2009. 2

[42] Kyle Wilson and Noah Snavely. Robust global translations with 1dsfm. In European Conference on Computer Vision, pages 61–75. Springer, 2014. 2, 4, 5, 6, 7

[43] Changchang Wu et al. Visualsfm: A visual structure from motion system. 2011. 1

[44] Ka-Veng Yuen. Bayesian methods for structural dynamics and civil engineering. John Wiley & Sons, 2010. 5

[45] Bingbing Zhuang, Loong-Fah Cheong, and Gim Hee Lee. Baseline desensitizing in translation averaging. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4539–4547, 2018. 2, 3, 4, 5, 6, 7