Baseline Desensitizing In Translation Averaging

Bingbing Zhuang, Loong-Fah Cheong, Gim Hee Lee
National University of Singapore
zhuang.bingbing@u.nus.edu, {eleclf, gimhee.lee}@nus.edu.sg

Abstract

Many existing translation averaging algorithms are either sensitive to disparate camera baselines and have to rely on extensive preprocessing to improve the observed Epipolar Geometry graph, or if they are robust against disparate camera baselines, require complicated optimization to minimize the highly nonlinear angular error objective. In this paper, we carefully design a simple yet effective bilinear objective function, introducing a variable to perform the requisite normalization. The objective function enjoys the baseline-insensitive property of the angular error and yet is amenable to simple and efficient optimization by block coordinate descent, with good empirical performance. A rotation-assisted Iterative Reweighted Least Squares scheme is further put forth to help deal with outliers. We also contribute towards a better understanding of the behavior of two recent convex algorithms, LUD [20] and Shapefit/kick [9], clarifying the underlying subtle difference that leads to the performance gap. Finally, we demonstrate that our algorithm achieves overall superior accuracies in benchmark dataset compared to state-of-the-art methods, and is also several times faster.

1. Introduction

Modern large-scale Structure-from-Motion (SfM) systems have enjoyed widespread success in many applications [26, 33, 27, 21, 17]. Earlier methods often adopt an incremental method by adding the cameras one by one sequentially as the size of the model grows up. As a consequence, the quality of the result heavily depends on the order in which the cameras are added, and the accumulated error often leads to significant drift as the size of the model increases. Therefore, frequent intermediate bundle adjustments (BA) [29] have to be applied to obtain stable result, which unfortunately increases the computational cost substantially. Given such disadvantages, the global SfM method emerges as a serious alternative. Unlike the incremental method, the global method attempts to determine the absolute poses for all the cameras simultaneously from all the observed pairwise Epipolar Geometry (EG) [12]. Such a holistic approach spreads the error as uniformly as possible to the whole model, avoiding the problem of error accumulation and drift. Thus, BA needs to be run only once as a final refinement, leading to a more efficient system.

Formally, a global SfM algorithm takes as input a view graph $G=(V, E)$, where each node $V_i$ in $V$ and edge $E_{ij}$ in $E$ represent respectively a camera and relative pose $(R_{ij}, t_{ij})$ between the camera pair $i$ and $j$ whose fields of view overlap. It aims to find the absolute rotation $R_i$ (a.k.a. rotation averaging) and location $t_i$ (a.k.a. translation averaging) for each camera (up to a gauge freedom), such that the observed pairwise relative poses are best explained. In the noiseless case, the following two equations hold:

$$R_i^T R_j = R_{ij}, \quad \frac{t_j - t_i}{||t_j - t_i||_2} = R_i t_{ij}. \quad (1)$$

Typically, rotation averaging is performed before translation averaging. In this paper, we follow this practice and shall focus on the second equation to perform translation averaging. Note that rotation is assumed to have been solved, and henceforth, for brevity, we shall denote $R_i t_{ij}$ as $v_{ij}$.

Translation averaging is recognized as a hard task. One of the reasons is that the input relative translation estimate is sensitive to small camera baselines [7]. More importantly, EG only encodes the relative direction between cameras without any magnitude information. This causes a remove of the measurement space (directions between pairwise cameras) from the solution space (camera locations); this gap complicates the task much more, posing a significant challenge for the objective function design. The geometrically more meaningful objective would be to minimize the angular error between unit direction vectors [24, 32], but this leads to highly nonlinear functions that require complicated optimization. Instead, many recent methods simply ignore the normalization terms required for obtaining unit vectors, thereby yielding various forms of quasi-Euclidean distance terms in the objectives [16, 20, 9]. Often, such expediency allows the problem to be formulated as a convex optimization problem. However, the serious qualification of such magnitude-based objective functions is that they suffer from unbalanced weighting on each individual term when
the camera baselines are disparate in lengths, which may lead to biased solutions. These methods usually employ extensive preprocessing (e.g., outlier filtering) of the view graph to obtain better relative translations as input. This relieves but does not resolve the issue fundamentally and the accuracy is still limited in practice.

The contributions in this paper are threefold. (1) We show that by carefully designing the objective function, the numerical sensitivity of the solution to different camera baseline lengths can be readily removed. Specifically, we propose a return to the geometrically more meaningful angular-error based objective function, putting forth a simple yet accurate Bilinear Angle-based Translation Averaging (BATA) framework. The key idea is to introduce a variable that performs the requisite normalization for a baseline-insensitive angular error term. This splits the original problem into easier subproblems, which can be easily optimized by block coordinate descent; empirically, the algorithm converges fast and yields superior performance. (2) To deal with outlier EG, we put forth a rotation-assisted iterative Reweighted Least Squares (IRLS) scheme that leverages on the stable solution from rotation averaging as an extra source of information to determine the reliability of each observation in the view graph. (3) Our objective formulation also lends perspective to the behavior of various algorithms with a magnitude-based objective function [16, 20, 9]. We reveal that the subtle difference in the scale ambiguity removal strategy can nevertheless lead to rather different performance in such algorithms. Specifically, we build the equivalence between Shapefit/kick [9] and a slightly revised version of LUD [20], which allows us to trace the difference between Shapefit/kick and LUD to the scale ambiguity removal constraint. We then demonstrate that a weaker lower-bound constraint can cause a squashing effect on the overall shape of the recovered camera locations, especially under the presence of disparate baselines; conversely, a stronger constraint would help desensitize the effect of unbalanced baselines.

We demonstrate the utility of the proposed framework by extensive experiments on both synthetic and real data. In particular, we obtain superior performance on the benchmark 1DSfM dataset [32] both in terms of accuracy and efficiency compared to state-of-the-art methods. The code will be made publicly available.

2. Related Work

Rotation Averaging. Many methods exist for this task [11, 4, 15, 5, 8, 10]; we refer readers to [31] for a survey. View Graph Preprocessing. Some methods [35, 32] utilize loop consistency to remove outlier EG in the view graph. Some other works attempt to refine the whole view graph using loop consistency [23] or low-rank constraint [22]. A robust re-estimation of the pairwise translation after the recovery of absolute rotation is proposed in [20].

Translation Averaging. The pioneering work by Govindu [10] proposes to minimize the cross product between the relative camera location \( t_j - t_i \) and the observed direction \( v_{ij} \). An ad-hoc iterative reweighting scheme is adopted to reduce the bias from different baseline lengths. As reported in [32], this generates poor accuracy in challenging dataset. Some methods aim to minimize the relaxation of the endpoint distance \( \| t_j - t_i - t_i \|_2^2 v_{ij} \|_2 \) or its variants. For example, Moulon et al. [16] propose to minimize a relaxed version using the \( L_\infty \) norm. A similar penalty is utilized in [20] but with a least unsquared deviations (LUD) form to be more robust. Goldstein et al. [9] propose a Shapefit/kick scheme based on the alternating direction method of multipliers (ADMM) to minimize the magnitude of the projection of \( t_j - t_i \) on the orthogonal complement of \( v_{ij} \). Despite its convex formulation, works such as [16, 20, 9] suffer from bias due to the unnormalized camera baseline magnitude in their objectives, and often have to resort to extensive preprocessing strategies reviewed in the preceding paragraph to take more accurate EG view graph as input.

Works that minimize the angular residual between \( t_j - t_i \) and \( v_{ij} \), denoted as \( \theta_{ij} \), are relatively rare in the literature. One of the representative works is that of Sim and Hartley [24]. They show that minimizing the maximal absolute value of \( \tan \theta_{ij} \) from all observations, i.e. \( L_\infty \) norm, can be reformulated into a quasi-convex problem and a globally optimal solution can be found by solving a sequence of Second Order Cone Programming (SOCP) feasibility problems. However, it is well known that \( L_\infty \) is sensitive to outliers, and solving multiple SOCP problems restricts their method to medium-size problems. Wilson et al. [32] present another attempt by minimizing the residual of \( \sin \theta_{ij} / 2 \). The trust-region method Levensen-Marquard is applied to optimize the resultant highly nonlinear function. In our work, we present another objective function along this line of approach; it minimizes the residual of \( \sin \theta_{ij} \) in essence, is easily optimizable and yet achieves superior performance.

Other heuristics have been proposed. These include coplanar constraint on triple cameras [13], reprojection error [11, 15], reducing the problem into similarity averaging by local depth estimation [6], and others [23, 5, 30, 19, 2]. We note that existing methods often include scene points to assist translation averaging and/or involve careful outlier filtering step. In this paper, we demonstrate the possibility of achieving good accuracy in practice even if we directly process the raw view graph. Such a concise framework is more amenable to efficient processing.

3. Method

3.1. Bilinear Angle-based Translation Averaging

Instead of penalizing the angular deviation \( \theta_{ij} \) between
\( t_j - t_i \) and the observed \( v_{ij} \) (illustrated in Fig. 1(a)) directly, many existing algorithms adopt an objective function of the form \( \sum_{i,j \in E} \sqrt{\left( t_j - t_i \right)^2 - \left( v_{ij} \right)^2} \) (Fig. 1(d)) or its variants. Note that this objective function is not normalized by the vector magnitude \( \| t_j - t_i \|_{2} \), and is thus plagued by numerical difficulties when there is large variation in the magnitudes of \( t_j - t_i \). In particular, though this objective will yield the true solution in the absence of noise, it is not necessarily statistically optimal under noisy condition. Let us illustrate this point via a toy example in the 2D plane. Referring to Fig. 2 suppose we know the ground-truth locations of three neighboring cameras to be at (1, 0), (0, -1) and (5, 0) and would like to localize the fourth camera, with ground truth at (0, 0), according to its observed pairwise directions with respect to the three neighboring cameras. Due to noise, suppose all these observed directions deviate from their true direction by 3°. We use the red dot to denote the best location (found by exhaustive search on 2D grid) that minimizes the preceding magnitude-based residual. We also use the black star to denote the best location that minimizes the squared angular deviation \( \sum_{i,j \in E} \theta_{ij}^2 \). Clearly, the solution in the former case is much worse off compared to that in the latter. In the former case, the objective function is essentially trying to determine the intersection point of the three direction vectors (in some least squares sense). This process is highly susceptible to errors when one or more cameras are far away. It follows that using such a magnitude-based objective function for the translation averaging problem would also experience similar sensitivity issue when there are disparate camera baseline distances.

In view of the foregoing discussion, we propose the following angle-based objective function instead:

\[
\min_{\hat{t}_j, i,j \in E} \sum_{i,j \in E} \rho\left(\{\hat{t}_j - t_i\}d_{ij} - v_{ij}\right),
\]

subject to

\[
\sum_{i \in V} t_i = 0, \sum_{i,j \in E} (\hat{t}_j - t_i, v_{ij}) = 1,
\]

\[
d_{ij} \geq 0, \forall i,j \in E,
\]

where \( \rho(\cdot) \) stands for a robust M-estimator function to be discussed in the next section. \( d_{ij} \) is a non-negative variable. The first two constraints on \( t \) are to remove the inherent positional and scale ambiguity. We now show that the optimal solution of this problem essentially minimizes an angular error \( \sum_{i,j \in E} \rho(h(\theta_{ij})) \), where

\[
h(\theta_{ij}) = \begin{cases} 
\sin \theta_{ij}, & \theta_{ij} \leq 90^\circ; \\
1, & \theta_{ij} > 90^\circ.
\end{cases}
\]

First, note that for any given candidate solution \( \hat{t} \), each \( d_{ij} \) serves to scale the relative location vector \( \hat{t}_j - \hat{t}_i \) such that the Euclidean distance between the endpoint of \((\hat{t}_j - \hat{t}_i)d_{ij}\) and the unit vector \( v_{ij} \) is minimized. It follows that if the angle between \( \hat{t}_j - \hat{t}_i \) and \( v_{ij} \) is less than \( 90^\circ \), the optimal value \( d_{ij} \) would be such that \( (\hat{t}_j - \hat{t}_i)d_{ij} \) equals to the projection of \( v_{ij} \) onto the direction along \( \hat{t}_j - \hat{t}_i \) and the penalty amounts to \( \sin \theta_{ij} \) (Fig. 1(b)). The constraint \( d_{ij} > 0 \) prevents \( d_{ij} \) from overcompensating when \( \theta_{ij} > 90^\circ \), otherwise the objective would decrease when \( \theta_{ij} \) increases from \( 90^\circ \) to \( 180^\circ \). With this constraint, the optimal \( d_{ij} \) when \( \theta_{ij} > 90^\circ \) would be 0 and the penalty would be 1.

As an alternative viewpoint, we could have regarded our objective function as a functional lifting and relaxation of the formulation in 1DSfM [32] which is an unconstrained minimization of the objective function \( \sum_{i,j \in E} \rho\left(\{\hat{t}_j - t_i\}d_{ij} - v_{ij}\right)^2 \). The scale factor \( 1/\|\hat{t}_j - t_i\|_{2} \) is replaced with the variable \( d_{ij} \) together with a relaxation of the constraint from \( d_{ij} = 1/\|\hat{t}_j - t_i\|_{2} \) to \( d_{ij} > 0 \). We note that since the scale factor in 1DSfM always normalizes the vector \( \hat{t}_j - t_i \) to a unit vector, its objective amounts to the penalty term \( 2 \sin(\theta_{ij}/2) \) (Fig. 1(c)), which bears a close resemblance to the \( \sin \theta_{ij} \) established in (3). It is clear that without any prior knowledge on the noise distribution, there is no reason for one to claim superiority over the other. Thus, while one can regard our objective function as a relaxation of that of 1DSfM, one should not see the relaxed version as a poorer cousin of the two, since there is nothing sacrosanct about \( 2 \sin(\theta_{ij}/2) \) over \( \sin \theta_{ij} \) in terms of its geometrical meaning. We also note that the relaxation reduces the original highly nonlinear term into a bilinear one that permits simple alternating optimization; compared to the solution of 1DSfM from Ceres [1], we empirically observe that BATA can generally recover the camera locations more reliably in real Internet photo collections, especially for those challenging sparsely connected cameras. Also note that our penalty term in (3) levels off after \( \theta_{ij} > 90^\circ \), and this might bestow greater robustness to our formulation.
3.2. Robust Rotation-Assisted IRLS

As the estimated EG view graph often contains gross outliers, we thus embed the least squares objective into a M-estimator $\rho(\cdot)$. Iterative Reweighted Least Squares scheme is often used to optimize such objective, whereby a weighted least squares problem is solved in each iteration. The weight function, denoted as $\phi(\cdot)$ here, returns a value proportional to the goodness of fit of an observation $ij$, evaluated at the last iteration. Note that the specific form of $\phi(\cdot)$ depends on the M-estimator function $\rho(\cdot)$ being used, e.g. for Cauchy $\rho(\varepsilon)=\log(1+\varepsilon^2/\alpha^2)$ and $\phi(\varepsilon)=\alpha^2/(\alpha^2+\varepsilon^2)$, where $\varepsilon$ denotes the residual for each observation and $\alpha$ is the loss width. Since it is well known that rotation averaging can often be computed more reliably \[ [11,4], a natural idea is to leverage its result to assist the reliability assessment or weighting for each observed EG. We thus use for this purpose the following residual $\varepsilon = (\parallel(t_i-t_j)\parallel_2^2 + \beta \parallel R_{ij}^T R_{ij} - R_{ij}^2 \parallel_2^2)^{1/2}$, whereby the goodness of fit of the rotation estimate also contributes to the weighting process. Here, $\beta$ is a predefined weighting factor (set as 1 for all the experiments). It turns out that this strategy can generally improve the accuracy and speed up the convergence of BATA. For each IRLS iteration, we use Block Coordinate Descent (BCD) to optimize $t$ and $d$, as summarized in Algo. [1]

Algorithm 1 IRLS-BCD solver

Input: View Graph $G=(V, E)$, Rotation Averaging Result.
Output: Camera Locations $t_i, \forall i \in V$.
1: Initialize $t_i, \forall i \in V, W_{ij}, \forall ij \in E$. Set $n = 0$;
2: while $n < \text{IRLSIter AND not converged}$ do
3: \hspace{1cm} $m = m + 1$;
4: \hspace{1cm} while $m < \text{BCDIter}$ do
5: \hspace{2cm} Update $d_{ij}$: $d_{ij} = \max ((t_j - t_i) - v_{ij}, 0)$;
6: \hspace{2cm} Update $t_i$: Solve a sparse, weighted, constrained linear least squares system of equations collected from \[ by Cholesky decomposition (see supp. material for more details);
7: \hspace{1cm} end while
8: end while
9: Update $W_{ij}$: $W_{ij} = \phi(\varepsilon)$, where $\varepsilon = (\parallel(t_i-t_j)d_{ij} - v_{ij}\parallel_2^2 + \beta \parallel R_{ij}^T R_{ij} - R_{ij}^2 \parallel_2^2)^{1/2}$;
10: \hspace{1cm} $n = n + 1$;
11: end while

3.3. Why does Shapefit/kick outperform LUD?

Using the same geometric analysis, we are now ready to clarify why Shapefit/kick \[ outperforms LUD \[ (if run on the same problem instances), as reported in \[ and verified by our experiments. For ease of discussion, we present their respective formulations below.

LUD:

\[
\min_{t_i, i \in V, d_{ij}, ij \in E} \sum_{ij \in E} \parallel t_j - t_i - d_{ij}v_{ij}\parallel_2^2, \tag{4}
\]

\[ s.t. \sum_{i \in V} t_i = 0; d_{ij} \geq c, \forall ij \in E, \]

where $d_{ij}$ is deemed as a relaxation of $\parallel t_j - t_i\parallel_2$.

Shapefit/kick:

\[
\min_{t_i, i \in V} \sum_{ij \in E} \parallel P_{v_{ij}}(t_j - t_i)\parallel_2^2, \tag{5}
\]

\[ s.t. \sum_{i \in V} t_i = 0, \sum_{ij \in E} \langle t_j - t_i, v_{ij} \rangle = 1, \]

where $P_{v_{ij}}$ denotes the projection onto the orthogonal complement of the span of $v_{ij}$.

To tease out the connection between these formulations, we replace the LUD’s constraint $d_{ij} \geq c$, which is for removing scale ambiguity and preventing all cameras from collapsing to a single point (under such case the penalty cost vanishes), with the one used in Shapefit/kick, i.e. $\sum_{ij \in E} \langle t_j - t_i, v_{ij} \rangle = 1$. We claim that the resultant optimization problem, denoted as RevisedLUD, has exactly the same optimal solution as that of Shapefit/kick. To verify this, we note that removing the constraint $d_{ij} \geq c$ reduces $d_{ij}$ to a completely free variable. Similar to the analysis for our formulation \[, for a set of estimates $t_i, \forall i \in V$, the optimal $d_{ij}$ would be such that $d_{ij}v_{ij}$ equals to the projection of $t_j - t_i$ onto the direction along $v_{ij}$. It is immediately clear that the residual being minimized in RevisedLUD is $\parallel t_j - t_i \parallel_2 \sin \theta_{ij}$ (Fig. \[e]) for any $\theta_{ij}$. It is also clear that $\parallel P_{v_{ij}}(t_j - t_i)\parallel_2$ coincides with $\parallel t_j - t_i \parallel_2 \sin \theta_{ij}$ and this establishes our claim. We note here that RevisedLUD can also be assisted by rotation when optimized with IRLS in Algo. \[ with small changes (e.g. step 4 becomes $d_{ij} = (t_j - t_i, v_{ij})$). We will have occasion to use this later when a convex initialization for BATA is called for.

Given the above equivalence, we can restrict our attention to the sole difference between the two algorithms, namely in the constraints discussed above, with a view to elucidating the impact that different formulations of these constraints have on performance. First observe that while the constraint used in RevisedLUD fixes the overall scale to a constant value, the one in LUD, i.e. $d_{ij} \geq c$, only imposes a lower bound on the scale. We note that the latter is a weaker constraint, in the sense that while it prevents the collapsing of cameras all the way to a single point, it still allows a partial shrinking to occur, i.e. shrinking without respecting the overall global shape. To see the difference more explicitly, suppose we feed the optimal solution $t^S$ obtained from the more strongly constrained \[ into \[. The solution space in \[ can be conceptually distinguished into two optimization regimes 1 and 2. Regime 1 admits solution of the form $\gamma t^S$ where $\gamma$ is a scale to be optimized together with $d_{ij}$’s. Clearly, the optimal solution in regime 1 is essentially identical to that of \[. However, it is generally not the optimal solution if we permit regime 2, which solves the original \[ without the $\gamma t^S$ restriction. As a consequence, the total residual may be further reduced by adjusting the scale of each residual term individually without respecting the overall global shape. In particular, those $i$–$j$ terms representing large baselines often have larger residu-

4542
als as any slight deviation of the solution from the observation is scaled by the baseline magnitude. As the corresponding \( d_{ij} \)'s are less likely to have reached the lower bound \( c \) (since the optimal \( d_{ij} = \min\langle t_j - t_i, v_{ij} \rangle, c \) for the solution \( t \)), it thus pays to scale those larger baselines \( \| t_j - t_i \|_2 \) and \( d_{ij} \)'s down by shrinking camera \( i \) and \( j \) closer together as long as the reduction in residuals at these \( i-j \) terms can more than make up for the increase in residuals in other terms. The best solution would therefore exhibit a partial shrinking effect, with the overall shape of the camera configuration squashed. We give a concrete toy example to illustrate how disparate baselines would exacerbate this tendency. For simplicity, let us look at a four-camera 2D case where the true camera locations are at \((0, 1), (-1, 0), (0, -1)\) and \((10, 0)\), and all their pairwise relative directions are observed with a 3\(^\circ\) noise. We first visualize the solutions obtained from (4) and from (5) together with the ground truth in Fig. 3(a). As can be seen, the distant camera 4 gravitates significantly towards cameras 1-3 in the solution of (4). We also plot the pairwise camera distances and the residuals of the best solutions (with \( c=1 \)) from regime 1 (optimized by linearly searching \( \gamma \)) and 2 in Fig. 3(b) & (c), respectively. Fig. 3(b) corroborates what we said above: compared to the solution of regime 1, regime 2 tends to pull those well-separated camera pairs closer together. Note that for those camera pairs that have short baselines, their corresponding \( d_{ij} \) has already reached the lower bound, and thus the \( t_i \) and \( t_j \) cannot shrink in tandem with those of the well-separated cameras without \( \| t_j - t_i \|_2 \) deviating too far from the \( d_{ij} \), leading to squashing effect in the camera configuration. Referring to Fig. 3(c), first note that in this toy example the solutions of both regime 1 and 2 agree with most of the observations exactly; we attribute this to the fact that the objectives of (4) and (5) are actually based on a group-sparsity term \((l_{2,1})\), i.e. \( l_{2,1} \) norm, which favors sparse residuals. Note also that regime 2 achieves a lower total residual (0.727) compared to regime 1 (0.747) by suppressing the large residual (the lone yellow peak) at the expense of the small residuals, validating our analysis above.

This observation is generally useful and applicable to other methods where such lower bound exists (e.g. [16]), or latter works based on LUD framework (e.g. [22, 27]).

4. Experiments

4.1. Synthetic Data Experiments

We first study the performance of different methods on synthetic data. To synthesize the view graph, we first generate the ground-truth camera locations \( \hat{t}_i, \forall i \in V \), by drawing i.i.d. samples from \( N(0, I_{3 \times 3}) \). Denoting the number of cameras as \( n \) (set as 200 here), the pairwise edges \( E \) are then drawn randomly from the Erdős-Rényi model \( G(n, p) \), meaning each edge is observed with probability \( p \), independently of all other edges. We then perturb the observed pairwise directions to mimic the effect of noises and outliers. As opposed to [20, 9], where Gaussian noises are added to the endpoint of the direction vector followed by a normalization to be of unit norm, we directly add noise to the orientation of the pairwise direction; we believe this to be more reflective of the actual perturbation. Specifically, we obtain each corrupted pairwise direction \( v_{ij} \) as follows,

\[
v_{ij} = \begin{cases} v_{ij}'^u, & \text{with probability } q, \\ R(\theta_{ij}^u, h_{ij}^u)(\bar{t}_j - \bar{t}_i) / \| \bar{t}_j - \bar{t}_i \|_2, & \text{otherwise}, \end{cases}
\]

where \( v_{ij}'^u \) and \( h_{ij}^u \) are i.i.d. unit random vectors drawn from uniform distribution on the unit sphere and the orthogonal complement of the span of \( \bar{t}_j - \bar{t}_i / \| \bar{t}_j - \bar{t}_i \|_2 \), respectively. \( \theta_{ij}^u \) is drawn from i.i.d. \( N(0, 1) \) and \( \sigma \) is a scale controlling the noise level. \( R(\theta_{ij}^u, h_{ij}^u) \) is a rotation matrix around the aixs \( h_{ij}^u \) for an angle \( \theta_{ij}^u \) (counter-clockwise). Like [20], we use the normalized root mean square error (NRMSE) to evaluate the accuracy: \( NRMSE = \sqrt{\sum_{i \in V} \| \hat{t}_i - \bar{t}_i \|_2^2} / \| \bar{t}_i \|_2^2 = 1 \) and the same is true of \( \bar{\hat{t}} \).

We compare the performance of different objectives including LUD [20], Shapefit/kick [9], the nonlinear objective from 1DSfM [32], and BATA. We follow 1DSfM to use Huber as the robust scheme for BATA. For 1DSfM and BATA, we evaluate both random initialization and initialization from RevisedLUD. In the latter, we run a few iterations (IRLSIter=10 & BCDIter=1) of naïve IRLS for RevisedLUD to bootstrap the 1DSfM and BATA (the results are denoted as “Con.Init.+1DSfM” and “Con.Init.+BATA”). We use Ceres [1] for the optimization of 1DSfM. LUD and Shapefit/kick are optimized by IRLS and ADMM. All methods are run until they are well converged (we fix the number of iterations for BATA as IRLSIter=20 & BCDIter=5).

We investigate the performance under six combinations of observation ratio \( p \) and outlier ratio \( q \), each with increasing noise level \( \sigma \), as shown in Fig. 4. The results are aver-
ill-conditioned with lower accuracies when the camera configuration becomes increasingly strapped with a good initialization, achieves higher accuracy compared to the magnitude-based counterparts. In particular, both 1DSfM and BATA can achieve lower NRMSE compared to the magnitude-based counterparts.

We fix \((p, q) = (0.3, 0.2)\) and \(\sigma = 10\), and increase the value of \(L\) from 0 to 10. For each solution, we compute NRMSE\(^1\) and the ratio \(r_3 = l_{12}/(l_1 + l_2)\). \(l_{12}\) denotes the distance of the two cluster centers. \(l_1\) and \(l_2\) denote the median value of the set of distances from each point to their centers in the two clusters respectively. Note that \(r_3\) explicitly measures the squashing effect caused by the different shrinking rates experienced by the longer inter-cluster baselines versus that of the shorter intra-cluster baselines. As shown in Fig. 6(b)-(c), under increasing \(L\), the NRMSE's from the two magnitude-based methods, especially LUD, increase more significantly, meaning that they are more susceptible to the disparate baselines; it is also clear that \(r_3\) from LUD decreases substantially compared to the ground truth, indicating a significant squashing effect.

### 4.2. Real Data Experiments

We now present the results on real unordered photo collections provided by the 1DSfM dataset \([32]\) (see Fig. 7(a) for examples). The raw largest connected view graph released along with the dataset is used as our input. Similar to \([32, 9, 6, 20]\), we apply the method of \([4]\) to perform rotation averaging. To quantitatively evaluate the quality of a translation averaging estimate, it is compared with the gold standard output by Bundler \([26]\); the two sets of camera positions are robustly registered using the codes of \([32]\).

We evaluate the performance of a few different setups under BATA to understand its behavior. The first case of interest is to simply run BATA from random initialization in two settings, without or with rotation involved in the IRLS re-weighting (denoted as “R.I. w/o R.” and “R.I. w R.”). Next, we use as initialization the moderately accurate output of a convex algorithm: to this end, we run a few rotation-

\(^1\)Here, we centralize and normalize two clusters separately to avoid the inherent decreasing of NRMSE while increasing \(L\).
Table 1: Comparison of the accuracy of different methods in real data. \( N_c \) is the number of cameras in the view graph. \( \bar{\epsilon} \) and \( \epsilon \) respectively denote the median and mean distance error in meter unit. \#iter denotes the number of outer iterations (the value of \( n \) in Algo. 1) required for convergence, bounded by 100. '-' indicates that the result is not available from the corresponding paper.
Table 2: Comparison of running time in seconds. \( T_p \), \( T_{ini} \), \( T_i \), and \( T_{m} \) respectively denote the preprocessing time, initialization time, translation averaging time, and total time.

Table 3: Comparison of running time in seconds. \( T_p \), \( T_{ini} \), \( T_i \), and \( T_{m} \) respectively denote the preprocessing time, initialization time, translation averaging time, and total time.

Figure 9: Different error quantities plotted against the ratio of the observed edges removed.

fails to recover both of them, and Shapefit/kick is able to recover only the left one while the point cloud of the right one is somewhat blurred and shifted. BATA can recover both of them successfully and yields the most similar results to those of the Bundler.

Finally, we compare the running time of different methods. Ours are obtained on a normal PC with a 3.4 GHz Intel Core i7 CPU and 16GB memory. The results are given in Tab. 3. \( T_p \) denotes the general preprocessing time, which may include outlier filtering and pairwise translation re-estimation to improve the input quality \([32, 20, 9] \), or local depth estimation and local BA \([6]\). \( T_{ini} \) denotes the time for convex initialization in our method, \( T_i \) the time for solving the translation averaging optimization, and \( T_{m} \) the total time. As can be seen, since BATA directly processes the raw EG view graph and the sequence of sparse linear system of equations involved can be solved by highly efficient libraries, it is generally faster by several times.

5. Conclusion

In this paper, we advocate a return to angle-based objectives for translation averaging, proposing a simple yet effective bilinear formulation with a rotation-assisted IRLS scheme, achieving good empirical performance. This formulation also contributes to a better understanding of the behavior of the existing convex methods.

Acknowledgements. This work was partially supported by the Singapore PSF grant 1521200082 and the Singapore MOE Tier 1 grant R-252-000-637-112.
References

[1] S. Agarwal, K. Mierle, and Others. Ceres solver. [http://ceres-solver.org](http://ceres-solver.org)

[2] M. Arie-Nachimson, S. Z. Kovalsky, I. Kemelmacher-Shlizerman, A. Singer, and R. Basri. Global motion estimation from point matches. In 3DMPVT, 2012.

[3] F. Arrigoni, L. Magri, B. Rossi, P. Fragneto, and A. Fusiello. Robust absolute rotation estimation via low-rank and sparse matrix decomposition. In 3DV, 2014.

[4] A. Chatterjee and V. Madhav Govindu. Efficient and robust large-scale rotation averaging. In ICCV, 2013.

[5] D. Crandall, A. Owens, N. Snavely, and D. Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. In CVPR, 2011.

[6] Z. Cui and P. Tan. Global structure-from-motion by similarity averaging. In ICCV, 2015.

[7] O. Enqvist, F. Kahl, and C. Olsson. Non-sequential structure from motion. In ICCV Workshops, 2011.

[8] J. Fredriksson and C. Olsson. Simultaneous multiple rotation averaging using lagrangian duality. In ACCV, 2012.

[9] T. Goldstein, P. Hand, C. Lee, V. Voroninski, and S. Soatto. Shapefit and shapekick for robust, scalable structure from motion. In ECCV, 2016.

[10] V. M. Govindu. Combining two-view constraints for motion estimation. In CVPR, 2001.

[11] R. Hartley, J. Trumpf, Y. Dai, and H. Li. Rotation averaging. International journal of computer vision, 103(3):267–305, 2013.

[12] R. Hartley and A. Zisserman. Multiple view geometry in computer vision. Cambridge university press, 2003.

[13] N. Jiang, Z. Cui, and P. Tan. A global linear method for camera pose registration. In ICCV, 2013.

[14] F. Kahl. Multiple view geometry and the $L_{\infty}$-norm. In ICCV, 2005.

[15] D. Martinec and T. Pajdla. Robust rotation and translation estimation in multiview reconstruction. In CVPR, 2007.

[16] P. Moulon, P. Monasse, and R. Marlet. Global fusion of relative motions for robust, accurate and scalable structure from motion. In ICCV, 2013.

[17] P. Moulon, P. Monasse, R. Perrot, and R. Marlet. Openmvg: Open multiple view geometry. In International Workshop on Reproducible Research in Pattern Recognition, pages 60–74. Springer, 2016.

[18] F. Nie, H. Huang, X. Cai, and C. H. Ding. Efficient and robust feature selection via joint $l_{2,1}$-norms minimization. In NIPS, 2010.

[19] C. Olsson and O. Enqvist. Stable structure from motion for unordered image collections. Image Analysis, pages 524–535, 2011.

[20] O. Ozyesil and A. Singer. Robust camera location estimation by convex programming. In CVPR, 2015.

[21] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In CVPR, 2016.

[22] S. Sengupta, T. Amir, M. Galun, T. Goldstein, D. W. Jacobs, A. Singer, and R. Basri. A new rank constraint on multi-view fundamental matrices, and its application to camera location recovery. In CVPR, 2017.

[23] T. Shen, S. Zhu, T. Fang, R. Zhang, and L. Quan. Graph-based consistent matching for structure-from-motion. In ECCV, 2016.

[24] K. Sim and R. Hartley. Recovering camera motion using $L_{\infty}$ minimization. In CVPR, 2006.

[25] S. Sinha, D. Steedly, and R. Szeliski. A multi-stage linear approach to structure from motion. In ECCV Workshops, 2010.

[26] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. In ACM transactions on graphics (TOG), volume 25, pages 835–846. ACM, 2006.

[27] C. Sweeney. Theia multiview geometry library: Tutorial & reference. [http://theia-sfm.org](http://theia-sfm.org)

[28] C. Sweeney, T. Sattler, T. Hollerer, M. Turk, and M. Pollefeys. Optimizing the viewing graph for structure-from-motion. In ICCV, 2015.

[29] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment: A modern synthesis. In International workshop on vision algorithms, pages 298–372. Springer, 1999.

[30] R. Tron and R. Vidal. Distributed 3-d localization of camera sensor networks from 2-d image measurements. IEEE Transactions on Automatic Control, 59(12):3325–3340, 2014.

[31] R. Tron, X. Zhou, and K. Daniilidis. A survey on rotation optimization in structure from motion. In CVPR Workshops, 2016.

[32] K. Wilson and N. Snavely. Robust global translations with 1dsfm. In ECCV, 2014.

[33] C. Wu et al. Visualsfm: A visual structure from motion system. 2011.

[34] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 68(1):49–67, 2006.

[35] C. Zach, M. Klopschitz, and M. Pollefeys. Disambiguating visual relations using loop constraints. In CVPR, 2010.