Linear Global Translation Estimation with Feature Tracks

Zhaopeng Cui[1]
zhpcui@gmail.com
Nianjuan Jiang[2]
nianjuan.jiang@adsc.com.sg
Chengzhou Tang[1]
chengzhout@gmail.com
Ping Tan[1]
pingtan@sfu.ca

[1] GrUVi Lab
Simon Fraser University
Burnaby, Canada

[2] Advanced Digital Sciences Center of Illinois, Singapore

Abstract

This paper derives a novel linear position constraint for cameras seeing a common scene point, which leads to a direct linear method for global camera translation estimation. Unlike previous solutions, this method deals with collinear camera motion and weak image association at the same time. The final linear formulation does not involve the coordinates of scene points, which makes it efficient even for large scale data. We solve the linear equation based on L_1 norm, which makes our system more robust to outliers in essential matrices and feature correspondences. We experiment this method on both sequentially captured images and unordered Internet images. The experiments demonstrate its strength in robustness, accuracy, and efficiency.

1 Introduction

Structure-from-motion (SfM) algorithms aim to estimate scene structure and camera motion from multiple images, and they can be broadly divided into incremental and global methods according to their ways to register cameras. Incremental methods register cameras one by one [43, 47] or iteratively merge partial reconstructions [14, 26]. These methods require frequent intermediate bundle adjustment (BA) to ensure correct reconstruction, which is computationally expensive. Yet, their results often suffer from large drifting errors. In comparison, global methods (*e.g*. [10, 16, 23, 31, 36, 46]) register all cameras simultaneously, which has better potential in both efficiency and accuracy.

Global SfM methods often solve the camera orientations and positions separately. The global position estimation is more challenging than the orientation estimation due to the noisy pairwise translation encoded in essential matrices [12]. This paper focuses on the problem of global position (*i.e*. translation) estimation.

Essential matrix based translation estimation methods [4, 6, 16] can only determine camera positions in a parallel rigid graph [36], and they usually degenerate at collinear camera motion because the translation scale is not determined by an essential matrix. Trifocal tensor based methods [9, 23, 32, 41] could deal with collinear motion as the relative scales

Pages 46.1-46.13
DOI: https://dx.doi.org/10.5244/C.29.46

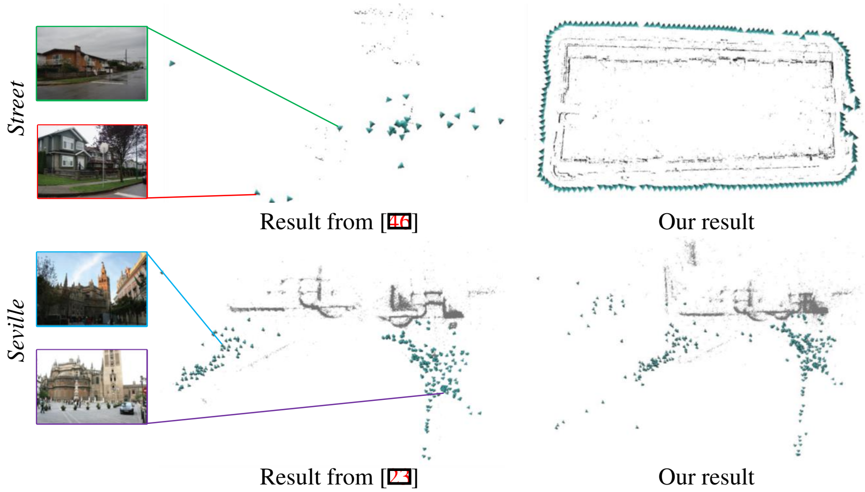

Figure 1: 1DSFM [46] and triplet-based methods (*e.g.* [23]) require strong association among images. As shown in the left, they fail for images with weak association. In comparison, as shown in the right, the results from our method do not suffer from such problems.

are encoded in a trifocal tensor. However, these methods usually rely on a strongly connected camera-triplet graph, where two triplets are connected by their common edge. The 3D reconstruction will distort or break into disconnected components when such strong association among images does not exist. By solving cameras and scene points together, some global methods [10, 24, 31, 40, 42] can deal with collinear motion. These methods usually need to filter epipolar geometries (EGs) carefully to avoid outliers. Including scene points in the formulation also hurts the scalability of the algorithm, since there are many more scene points than cameras. The recent 1DSfM method [46] designs a smart filter to discard outlier essential matrices and solves scene points and cameras together by enforcing orientation consistency. However, this method requires abundant association between input images, *e.g.* $\sim O(n^2)$ essential matrices for n cameras, which is more suitable for Internet images and often fails on sequentially captured data.

The data association problem of 1DSfM [46] and triplet-based methods (here, we take [23] as an example) is exemplified in Figure 1. The *Street* example on the top is a sequential data. 1DSfM fails on this example due to insufficient image association, since each image is only matched to its 4 neighbors at the most. In the *Seville* example on the bottom, those Internet images are mostly captured from two viewpoints (see the two representative sample images) with weak affinity between images at different viewpoints. This weak data association causes seriously distorted reconstruction for the triplet-based method in [23].

This paper introduces a direct linear algorithm to address the presented challenges. It avoids degeneracy at collinear motion and deals with weakly associated data. Our method capitalizes on constraints from essential matrices and feature tracks. For a scene point visible in multiple (at least three) cameras, we consider triangles formed by this scene point and two camera centers. We first generalize the camera-triplet based position constraint in [23] to our triangles with scene points. We then eliminate the scene point from these constraints. In this way, we obtain a novel linear equation for the positions of cameras linked by a feature track. Solving these linear equations from many feature tracks simultaneously register all cameras in a global coordinate system.

This direct linear method minimizes a geometric error, which is the Euclidean distance between the scene point to its corresponding view rays. It is more robust than the linear constraint developed in [40], which minimizes an algebraic error. A key finding in this paper is that, a direct linear solution (without involving scene points) exists by minimizing the point-to-ray error instead of the reprojection error. Since the point-to-ray error approximates the reprojection error well when cameras are calibrated, our method is a good linear initialization for the final nonlinear BA.

At the same time, this direct linear formulation lends us sophisticated optimization tools, such as L_1 norm optimization [6, 7, 15, 37]. We minimize the L_1 norm when solving the linear equation of camera positions. In this way, our method can tolerate a larger amount of outliers in both essential matrices and feature correspondences. The involved L_1 optimization is nontrivial. We derive a linearization of the alternating direction method of multipliers algorithm [5] to address it.

2 Related Work

Incremental approaches. Most of well-known SfM systems register cameras sequentially [3, 38, 39, 43, 44] or hierarchically [14, 22, 26] from pairwise relative motions. In order to minimize error accumulation, frequent intermediate bundle adjustment is required for both types of methods, which significantly reduces computation efficiency. The performance of sequential methods relies heavily on the choice of the initial image pair and the order of subsequent image additions [19].

Global rotation estimation. Global SfM methods solve all camera poses simultaneously. Most of these methods take two steps. Typically, they solve camera orientations first and then positions. The orientation estimation is well studied with an elegant rotation averaging algorithm presented in [21]. The basic idea was first introduced by Govindu [16], and then developed in several following works [17, 21, 31]. In particular, [8] introduced a robust L_1 method which was adopted in several recent works [36, 46].

Global translation estimation. The translation estimation is more challenging. Some pioneer works [4, 6, 16, 18] solved camera positions solely from constraints in essential matrices. Typically, they enforce consistency between pairwise camera translation directions and those encoded in essential matrices. Recently, Özyesil and Singer [36] prove that essential matrices only determine camera positions in a parallel rigid graph, and present a convex optimization algorithm to solve this problem. In general, all these essential matrix based methods degenerate at collinear motion, where cameras are not in a parallel rigid graph.

This degeneracy can be avoided by exploiting relative motion constraints from camera triplets [9, 32, 41, 42], as the trifocal tensor encodes the relative scale information. Recently, Jiang *et al.* [23] derived a novel linear constraint in a camera triplet and solved all cameras positions in a least square sense. While triplet-based methods avoid degenerated camera motion, they often require strong association among images – a connected triplet graph, where camera triplets are connected by common edges.

Some global methods estimate cameras and scene points together. Rother [40] solved camera positions and points by minimizing an algebraic error. Some works [2, 24, 27, 31, 34] solved the problem by minimizing the L_∞ norm of reprojection error. However, the L_∞ norm is known to be sensitive to outliers and careful outlier removal is necessary [11, 35]. Recently, Wilson and Snavely [46] directly solved cameras and points by Ceres Solver [1] after applying a smart filter to essential matrices. Generally speaking, involving scene points

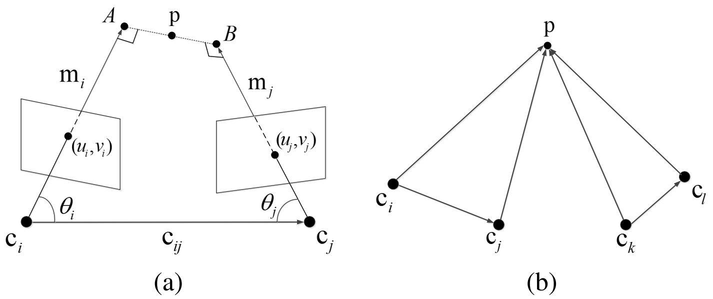

Figure 2: (a) The positions of a scene point $\mathbf{p}$ and two camera centers $\mathbf{c}_i$ and $\mathbf{c}_j$ satisfy a linear constraint detailed in Section 3.1. (b) The positions of four cameras seeing the same scene point satisfy a linear constraint detailed in Section 3.2.

improves the robustness/accuracy of camera registration, but also significantly increases the problem scale. Feature correspondence outliers also pose a significant challenge for these methods.

Our method capitalizes on constraints in essential matrices and feature tracks. It avoids degeneracy at collinear motion, handles weak data association, and is robust to feature correspondence outliers.

3 Global Translation Estimation

Given an essential matrix between two images i, j (*e.g.* computed by the five-point algorithm [28, 33]), we obtain the relative rotation $\mathbf{R}_{ij}$ and translation direction $\mathbf{t}_{ij}$ between the two cameras. Here, $\mathbf{R}_{ij}$ is a 3×3 orthonormal matrix and $\mathbf{t}_{ij}$ is a 3×1 unit vector. We further denote the global orientation and position of the i-th $(1 \leq i \leq N)$ camera as $\mathbf{R}_i$ and $\mathbf{c}_i$ respectively. These camera poses are constrained by the following equations

$$\mathbf{R}_j = \mathbf{R}_{ij}\mathbf{R}_i, \quad \mathbf{R}_j(\mathbf{c}_i - \mathbf{c}_j) \simeq \mathbf{t}_{ij}. \tag{1}$$

Here, $\simeq$ means equal up to a scale.

Like most global methods, we compute camera orientations first and solve camera positions after that. We adopt the global rotation estimation method in [8]. In order to enhance robustness, we adopt additional loop verifications [48] on the input pairwise relative camera rotations beforehand. Specifically, we chain the relative rotations along a three-camera loop as $\hat{\mathbf{R}} = \mathbf{R}_{ij}\mathbf{R}_{jk}\mathbf{R}_{ki}$, and compute the angular difference [21] between $\hat{\mathbf{R}}$ and the identity matrix. If the difference is larger than a threshold φ_1 (3 or 5 degrees for sequential data or unordered Internet data), we consider the verification fails. We discard an EG if every verification it participates in fails.

The key challenge in translation estimation is that an essential matrix does not tell the scale of translation. We seek to obtain linear equations for those unknown scales without resorting to camera-triplets. Our translation estimation is based on a linear position constraint arising from a triangle formed by two camera positions and a scene point. With this constraint, the positions of cameras linked by a feature point should satisfy a linear equation.

3.1 Constraints from a triangle

A linear constraint on positions of cameras in a triplet is derived in [23]. We generalize it to the case of triangles formed by a scene point and two cameras. As shown in Figure 2, we compute the location of a scene point $\mathbf{p}$ as the middle point of the mutual perpendicular line segment AB of the two rays passing through $\mathbf{p}$'s image projections. Specifically, it is computed as

$$\mathbf{p} = \frac{1}{2}(A+B) = \frac{1}{2}(\mathbf{c}_i + s_i\mathbf{m}_i + \mathbf{c}_j + s_j\mathbf{m}_j). \tag{2}$$

Here, $\mathbf{c}_i$ and $\mathbf{c}_j$ are the two camera centers. The two unit vectors $\mathbf{m}_i$ and $\mathbf{m}_j$ origin from the camera centers and point toward the image projections of $\mathbf{p}$. s_i and s_j are the distances from the points A, B to $\mathbf{c}_i, \mathbf{c}_j$ respectively, *i.e.* $A = \mathbf{c}_i + s_i\mathbf{m}_i$ and $B = \mathbf{c}_j + s_j\mathbf{m}_j$.

The rotation trick. The rotation trick used in [23] shows that we can compute $\mathbf{m}_i$ and $\mathbf{m}_j$ by rotating the relative translation direction $\mathbf{c}_{ij}$ between $\mathbf{c}_i$ and $\mathbf{c}_j$, *i.e.* $\mathbf{m}_i = \mathbf{R}(\theta_i)\mathbf{c}_{ij}$ and $\mathbf{m}_j = -\mathbf{R}(\theta_j)\mathbf{c}_{ij}$. Then Equation 2 becomes

$$\mathbf{p} = \frac{1}{2}\left(\mathbf{c}_i + s_i\mathbf{R}(\theta_i)\frac{\mathbf{c}_j - \mathbf{c}_i}{\|\mathbf{c}_j - \mathbf{c}_i\|} + \mathbf{c}_j + s_j\mathbf{R}(\theta_j)\frac{\mathbf{c}_i - \mathbf{c}_j}{\|\mathbf{c}_i - \mathbf{c}_j\|}\right). \tag{3}$$

The two 3D rotation matrices $\mathbf{R}(\theta_i)$ and $\mathbf{R}(\theta_j)$ rotate the relative translation direction $\mathbf{c}_{ij}$ to the directions $\mathbf{m}_i$ and $\mathbf{m}_j$. Both rotations can be computed easily in the local pairwise reconstruction. In addition, the two ratios $s_i/\left\|\mathbf{c}_j - \mathbf{c}_i\right\|$ and $s_j/\left\|\mathbf{c}_j - \mathbf{c}_i\right\|$ can be computed by the middle-point algorithm [20]. Specifically, assuming unit baseline length, in the local coordinate system attached to one of the cameras, $\mathbf{c}_i, \mathbf{c}_j$, $\mathbf{m}_i$, and $\mathbf{m}_j$ are all known. Thus, we can solve s_i and s_j (they are actually the two ratios in Equation 3 for general baseline length) from

$$(\mathbf{c}_i + s_i\mathbf{m}_i - \mathbf{c}_j - s_j\mathbf{m}_j) \times (\mathbf{m}_i \times \mathbf{m}_j) = 0. \tag{4}$$

Here, $\times$ is the cross product of vectors. Thus, Equation 3 becomes,

$$\mathbf{p} = \frac{1}{2}\left((\mathbf{A}_j^{ij} - \mathbf{A}_i^{ij})(\mathbf{c}_i - \mathbf{c}_j) + \mathbf{c}_i + \mathbf{c}_j\right) \tag{5}$$

where $\mathbf{A}_i^{ij} = s_i/||\mathbf{c}_j - \mathbf{c}_i||\mathbf{R}(\theta_i)$ and $\mathbf{A}_j^{ij} = s_j/||\mathbf{c}_i - \mathbf{c}_j||\mathbf{R}(\theta_j)$ are known matrices. This equation provides a linear constraint among positions of two camera centers and a scene point. Note this linear constraint minimizes a geometric error, the point-to-ray distance.

3.2 Constraints from a feature track

If the same scene point $\mathbf{p}$ is visible in two image pairs $\mathbf{c}_i, \mathbf{c}_j$ and $\mathbf{c}_k, \mathbf{c}_l$ as shown in Figure 2 (b), we obtain two linear equations about $\mathbf{p}$'s position according to Equation 5. We can eliminate $\mathbf{p}$ from these equations to obtain a linear constraint among four camera centers as the following,

$$(\mathbf{A}_j^{ij} - \mathbf{A}_i^{ij})(\mathbf{c}_i - \mathbf{c}_j) + \mathbf{c}_i + \mathbf{c}_j = (\mathbf{A}_l^{kl} - \mathbf{A}_k^{kl})(\mathbf{c}_k - \mathbf{c}_l) + \mathbf{c}_k + \mathbf{c}_l. \tag{6}$$

Given a set of images, we build feature tracks and collect such linear equations from camera pairs on the same feature track. Solving these equations will provide a linear global solution of camera positions. To resolve the gauge ambiguity, we set the orthocenter of all cameras at origin when solving these equations.

Equation 6 elegantly correlates the scales of translation (*i.e.* baseline length$\left\|\mathbf{c}_i - \mathbf{c}_j\right\|$ and $\left\|\mathbf{c}_k - \mathbf{c}_l\right\|$) for camera pairs sharing a common scene point. For example, in the *Seville* data in Figure 1 (bottom row), $\mathbf{c}_i, \mathbf{c}_j$ could come from one popular viewpoint of the building, and $\mathbf{c}_k, \mathbf{c}_l$ could come from a different viewpoint. As long as there is a feature track linking them together, Equation 6 provides constraints among the baseline lengths of these far apart cameras. In comparison, triplet-based methods (*e.g.* [23, 32, 41]) can only propagate the scale information over camera triplets sharing an edge. Clearly, the scale information can propagate further along feature tracks. Therefore, this new formulation can reconstruct images with weak association better than triplet-based methods.

3.3 Feature tracks selection

Since there are usually abundant feature tracks to solve camera positions, we carefully choose the most reliable ones to enhance system robustness. For better feature matching quality, we only consider feature correspondences that are inliers of essential matrix fitting. We sort all feature tracks by their lengths in descending order, and then try to find a small set of tracks that could cover all connected cameras at least K times. (We set $K = 30$ in our experiments.) Please see our supplementary material for the pseudo-code of the feature tracks selection.

For a feature track with N_t cameras, there are usually more than $N_t - 1$ EGs on it. So we select the most reliable ones to construct equations. We consider the match graph formed by these cameras, where two cameras are connected when their essential matrix is known. Since we only consider feature correspondences passing essential matrix verification, this graph has only one connected component. We weight each graph edge by $\frac{1}{M} + \alpha \frac{1}{\theta}$, where M is the number of feature matches between two images, and θ is the triangulation angle. The combination weight α is fixed at 0.1 in our experiments. We take the minimum spanning tree of this graph, and randomly choose two edges from the tree to build a linear equation until each edge is used twice.

4 Robust Estimation by L_1 Norm

Our linear global method requires solving a linear system like $\mathbf{A}\mathbf{x} = 0$ to estimate camera centers. $\mathbf{x}$ represents an unknown vector formed by concatenating all camera positions, and $\mathbf{A}$ is the coefficient matrix formed by collecting Equation 6 from feature tracks.

The 3D reconstruction process is noisy and involves many outliers, both in essential matrices and feature correspondences. We enhance system robustness by minimizing the L_1 norm, instead of the conventional L_2 norm. In other words, we solve the following optimization problem,

$$\arg\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x}\|_1, \qquad s.t. \quad \mathbf{x}^\top \mathbf{x} = 1. \tag{7}$$

This problem might be solved by iterative reweighted total least squares, which is often slow and requires good initialization. Recently, Ferraz *et al.* [13] proposed a robust method to discard outliers, while it is not applicable to our large sparse homogeneous system. We capitalize on the recent alternating direction method of multipliers (ADMM) [5] for better efficiency and large convergence region. Due to the quadratic constraint, *i.e.* $\mathbf{x}^\top \mathbf{x} = 1$, the original ADMM algorithm cannot be directly applied to our problem. We linearize the optimization problem in the inner loop of ADMM to solve Equation 7.

Let $\mathbf{e} = \mathbf{A}\mathbf{x}$, the augmented Lagrangian function of Equation 7 is

$$L(\mathbf{e}, \mathbf{x}, \lambda) = \|\mathbf{e}\|_1 + \langle \lambda, \mathbf{A}\mathbf{x} - \mathbf{e} \rangle + \frac{\beta}{2} \|\mathbf{A}\mathbf{x} - \mathbf{e}\|^2, \qquad s.t. \qquad \mathbf{x}^\top \mathbf{x} = 1, \tag{8}$$

where λ is the Lagrange multiplier, $\langle \cdot, \cdot \rangle$ is the inner product, and $\beta > 0$ is a parameter controlling the relaxation.

Dataset	c_{err} (GT,mm)							
	Rother[40]	Jiang[23]	Moulon[32]	Arie[4]	VisualSFM[47]	1DSfM[46]	L_2	L_1
fountain-P11	**2.5**	3.1	**2.5**	2.9	3.6	33.5	**2.5**	**2.5**
Herz-Jesu-P25	**5.0**	7.5	5.3	5.3	5.7	36.3	**5.0**	**5.0**
castle-P30	347.0	72.4	21.9	-	70.7	-	21.6	**21.2**

Table 1: Reconstruction accuracy comparison on benchmark data with ground truth (GT) camera intrinsics.

Dataset	c_{err} (EXIF,mm)						
	Rother[40]	Jiang[23]	Arie[4]	VisualSFM[47]	1DSfM[46]	L_2	L_1
fountain-P11	23.3	14.0	22.6	20.7	32.2	**6.9**	7.0
Herz-Jesu-P25	49.5	64.0	47.9	45.3	64.9	**25.5**	26.2
castle-P30	2651.8	235.0	-	190.1	-	317.3	**166.7**

Table 2: Reconstruction accuracy comparison on benchmark data with approximate intrinsics from EXIF. The results by Moulon[32] are not available.

We then iteratively optimize $\mathbf{e}$, $\mathbf{x}$, and λ in Equation 8. In each iteration, we update $\mathbf{e}_{k+1}$, $\mathbf{x}_{k+1}$, λ_{k+1} according to the following scheme,

$$\mathbf{e}_{k+1} = \arg\min_{\mathbf{e}} L(\mathbf{e}, \mathbf{x}_k, \lambda_k) = \arg\min_{\mathbf{e}} \|\mathbf{e}\|_1 + \langle \lambda_k, \mathbf{A}\mathbf{x}_k - \mathbf{e} \rangle + \frac{\beta}{2} \|\mathbf{A}\mathbf{x}_k - \mathbf{e}\|^2, \quad (9)$$

$$\mathbf{x}_{k+1} = \arg\min_{\mathbf{x}\in\Omega} L(\mathbf{e}_{k+1}, \mathbf{x}, \lambda_k) = \arg\min_{\mathbf{x}\in\Omega} \langle \lambda_k, \mathbf{A}\mathbf{x} - \mathbf{e}_{k+1} \rangle + \frac{\beta}{2} \|\mathbf{A}\mathbf{x} - \mathbf{e}_{k+1}\|^2, \quad (10)$$

$$\lambda_{k+1} = \lambda_k + \beta(\mathbf{A}\mathbf{x}_{k+1} - \mathbf{e}_{k+1}), \quad (11)$$

where $\Omega := \left\{\mathbf{x}^\top \mathbf{x} = 1 | \mathbf{x} \in \mathbb{R}^n\right\}$. A closed-form solution [29] exists for the minimization of Equation 9. (Please see Appendix A for the formula.) Solving Equation 10 is hard because of the quadratic constraint on $\mathbf{x}$. Therefore, we linearize Equation 10 and derive a closed-form solution as,

$$\mathbf{x}_{k+1} = C / \|C\|^2, \quad (12)$$

where $C = \mathbf{x}_k - \frac{1}{\eta}\mathbf{A}^\top(\mathbf{A}\mathbf{x}_k - \mathbf{e}_{k+1}) - \frac{1}{\beta\eta}\mathbf{A}^\top\lambda^k$, and $\eta > \sigma_{max}(\mathbf{A}^\top\mathbf{A})$. (Please see Appendix B for more details.) In order to speed up convergence [30], we adopt a dynamic parameter β as,

$$\beta_{k+1} = min\{\beta_{max}, \rho\beta_k\}, \quad (13)$$

where $\rho > 1$. We set ρ as 1.01 or 1.1 for sequential data and Internet data respectively in our experiments. Algorithm 1 summarizes our linearized ADMM algorithm.

Algorithm 1 Our linearized ADMM for Equation 7.

```
Initialize: Set $\mathbf{x}_0$ as to the $L_2$ solution (i.e. the eigenvector with smallest eigenvalue of
  $\mathbf{A}$), $\mathbf{e}_0 = \mathbf{0}$, $\lambda_0 = \mathbf{0}$, $\beta_0 = 10^{-6}$;
while not converged, do
    Step 1: Update $\mathbf{e}$ by solving Equation 9;
    Step 2: Update $\mathbf{x}$ by solving Equation 12;
    Step 3: Update $\lambda$ by solving Equation 11;
    Step 4: Update $\beta$ by solving Equation 13;
end while
```

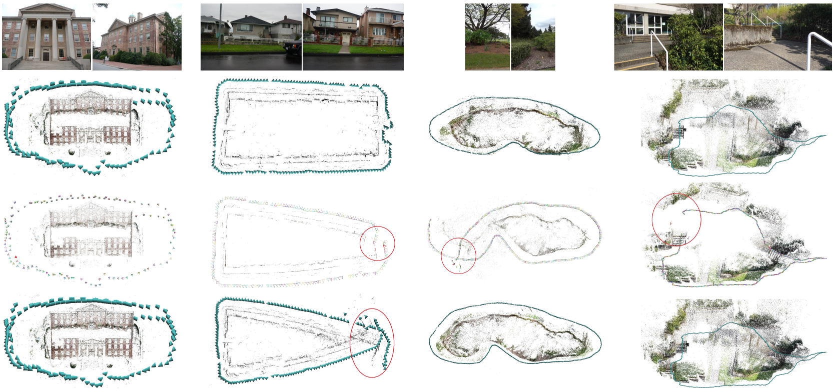
(a) *Building* (b) *Street* (c) *Park* (d) *Stair*

Figure 3: Evaluation on sequential data. From top to bottom, each row shows sample input images, 3D reconstructions generated by our method, VisualSFM [47], and the least unsquared deviations (LUD) method [36] respectively.

5 Experiments

5.1 Evaluation on benchmark data

We compare our method with VisualSFM [47], and several global SfM methods on the benchmark data provided in [45]. We use both ground truth camera intrinsics and approximate intrinsics from EXIF in the experiment. We implement the method in [40] by ourselves. The results on VisualSFM [47] and 1DSfM [46] are obtained by running the codes provided by the authors. To evaluate the L_1 norm optimization, we also experiment the conventional L_2 norm optimization instead of the L_1 norm in Equation 7. The results are indicated as L_1 and L_2 respectively.

We summarize all the results in Table 1 and Table 2. All results are evaluated after the final bundle adjustment. Our method generally produces the smallest errors with either ground truth intrinsics or approximate ones from EXIF. The L_2 and L_1 methods produce similar results on the *fountain-P11* and *Herz-Jesu-P25* data, since these data have few outliers. But the L_1 method outperforms L_2 significantly on the *castle-P30* data, whose essential matrix estimation suffers from repetitive scene structures. The noisy epipolar geometries also cause bad performance of the method in [40] on the *castle-P30* data, which solves cameras and scene points together by minimizing an algebraic error. In comparison, our method minimizes a geometric error, which is the point-to-ray distance, and achieves better robustness.

5.2 Experiment on sequential data

Figure 3 summarizes our experiment on sequentially captured data and compares our method with an incremental method, VisualSFM [47], and some recent global methods [36, 46]. The test data *Building*, *Street*, *Park* and *Stair* have 128, 168, 507 and 1221 input images respectively. Generally speaking, VisualSFM [47] suffers from large drifting errors when the input essential matrices are noisy or when the image sequence is long, as shown in the third row of Figure 3. The drifting errors in the *Park* and *Stair* examples are severe because of the poor essential matrix estimation due to poor feature localization in their tree images. We do not include results from 1DSfM [46] in Figure 3, since it fails on all these

Dataset		1DSfM[46]			LUD[36]				L_2		L_1			
Name	N_i	N_c	$\tilde{x}$	$\tilde{x}_{BA}$	$\bar{x}_{BA}$	N_c	$\tilde{x}$	$\tilde{x}_{BA}$	$\bar{x}_{BA}$	N_c	$\tilde{x}$	N_c	$\tilde{x}$	$\tilde{x}_{BA}$	$\bar{x}_{BA}$
Alamo	613	529	1.1	**0.3**	2e7	547	**0.4**	**0.3**	**2.0**	496	0.5	500	0.6	0.5	3.7
Ellis Island	242	214	3.7	**0.3**	3.0	-	-	-	-	183	9.4	211	**3.1**	0.6	**1.8**
Montreal N.D.	467	427	2.5	**0.4**	**1.0**	435	**0.5**	**0.4**	**1.0**	424	0.8	426	0.8	**0.4**	1.1
Notre Dame	552	507	10	1.9	7.0	536	**0.3**	**0.2**	**0.7**	537	**0.3**	539	**0.3**	**0.2**	0.8
NYC Library	369	295	2.5	**0.4**	**1.0**	320	2.0	1.4	7.0	-	-	288	**1.4**	0.9	6.9
Piazza del Popolo	350	308	3.1	2.2	2e2	305	**1.5**	**1.0**	4.0	302	3.6	294	2.6	2.4	**3.2**
Tower of London	499	414	11	**1.0**	40	425	4.7	3.3	10	311	17	393	**4.4**	1.1	**6.2**
Vienna Cathedral	897	770	6.6	**0.4**	2e4	750	5.4	4.4	10	574	3.6	578	**3.5**	2.6	**4.0**
Yorkminster	450	401	3.4	**0.1**	5e2	404	**2.7**	1.3	**4.0**	333	3.9	341	3.7	3.8	14

Table 3: Comparison with [46] on challenging data. N_i denotes the number of cameras in the largest connected component of our EG graph, and N_c denotes the number of reconstructed cameras. $\tilde{x}$ denotes the median error before BA. $\tilde{x}_{BA}$ and $\bar{x}_{BA}$ denote the median error and the average error after BA respectively. The errors are the distances in meters to corresponding cameras computed by an incremental SfM method [43].

Dataset	1DSfM[46]		LUD[36]		Bundler[43]	L_1	
	T_{BA}	T_Σ	T_{BA}	T_Σ	T_Σ	T_{BA}	T_Σ
Alamo	752	910	133	750	1654	362	621
Ellis Island	139	171	-	-	1191	64	95
Montreal N.D.	1135	1249	167	553	2710	226	351
Notre Dame	1445	1599	126	1047	6154	793	1159
NYC Library	392	468	54	200	3807	48	90
Piazza del Popolo	191	249	31	162	1287	93	144
Tower of London	606	648	86	228	1900	121	221
Vienna Cathedral	2837	3139	208	1467	10276	717	959
Yorkminster	777	899	148	297	3225	63	108

Table 4: Running times in seconds for the Internet data. T_{BA} and T_Σ denote the final bundle adjustment time and total running time respectively.

sequential data. Its result on the *Street* data is shown in Figure 1 (left of top row). 1DSfM cannot handle these examples because it is designed for Internet images which tend to have $O(n^2)$ essential matrices for n images [1]. The least unsquared deviations (LUD) method [36] generates distortion on the *Street* example, because it degenerates at collinear motion. In comparison, our method does not have visible drifting and is robust to collinear motion and weak image association.

5.3 Experiment on unordered Internet data

The input epipolar geometries for Internet data are quite noisy because of the poor feature matching. Besides using L_1 optimization, we adopt two additional steps to improve the robustness of our method. After solving camera orientations by the method in [8], we further filter input essential matrices with the computed camera orientations. Specifically, for each camera pair, we compare their relative rotation from their global orientation with the relative rotation encoded in their essential matrix. If the difference is larger than a threshold φ_2 (set to 5 or 10 degrees), we discard that essential matrix. What's more, we refine the relative translations with the camera orientations fixed using the method in [25].

We test our method on the challenging Internet data released by [46] and compare our method with several global methods. We use the results of an optimized incremental SfM system based on Bundler [43] as the reference 'ground-truth' and compute the camera position errors for evaluation. As shown in Table 3, our method with L_1 optimization has smaller initial median errors than [46] and comparable errors with [36]. Our method with L_1 opti-

[1] This is according to our discussion with the authors of 1DSfM [46].

mization performs better than L_2 solutions, which shows the effectiveness of the proposed L_1 method. All the methods have similar results after the final bundle adjustment.

Table 4 lists the running time of different methods. All our experiments were run on a machine with two 2.4GHz Intel Xeon E5645 processors with 16 threads enabled. We cite the running time for [46], [36] and [43] for comparison. Our method is around 10 times faster than the optimized incremental method [43] and also faster than the global methods [36, 46].

6 Conclusion

We derive a novel linear method for global camera translation estimation. This method is based on a novel position constraint on cameras linked by a feature track, which minimizes a geometric error and propagates the scale information across far apart camera pairs. In this way, our method works well even on weakly associated images. The final linear formulation does not involve coordinates of scene points, so it is easily scalable and computationally efficient. We further develop an L_1 optimization method to make the solution robust to outlier essential matrices and feature correspondences. Experiments on various data and comparison with recent works demonstrate the effectiveness of this new algorithm.

Acknowledgements. This work is supported by the NSERC Discovery grant 611664, Discovery Acceleration Supplements 611663, and the HCCS research grant at the ADSC from Singapore's Agency for Science, Technology and Research (A*STAR).

Appendix

A. Solution for Equation 9. From Equation 9, we have

$$\begin{aligned}\mathbf{e}_{k+1} &= \arg\min_{\mathbf{e}} \frac{1}{\beta}\|\mathbf{e}\|_1 + \frac{1}{2}\left\|\mathbf{A}\mathbf{x}_k - \mathbf{e} + \frac{\lambda_k}{\beta}\right\|^2 \\ &= \arg\min_{\mathbf{e}} \varepsilon\|\mathbf{e}\|_1 + \frac{1}{2}\|\mathbf{e}-\mathbf{u}\|^2, \end{aligned} \tag{14}$$

where $\varepsilon = \frac{1}{\beta}$, and $\mathbf{u} = \mathbf{A}\mathbf{x}_k + \frac{\lambda_k}{\beta}$. According to [29], the solution for Equation 14 is

$$\mathbf{e}^i_{k+1} = \begin{cases} \mathbf{u}^i - \varepsilon, & if \;\; \mathbf{u}^i > \varepsilon, \\ \mathbf{u}^i + \varepsilon & if \;\; \mathbf{u}^i < -\varepsilon, \\ 0, & otherwise, \end{cases} \tag{15}$$

where $\mathbf{e}^i_{k+1}$ and $\mathbf{u}^i$ are the i-th element of $\mathbf{e}$ and $\mathbf{u}$.

B. Derivation of Equation 12. We linearize the quadratic term $\frac{\beta}{2}\|\mathbf{A}\mathbf{x} - \mathbf{e}_{k+1}\|^2$ in Equation 10 at $\mathbf{x}_k$, which gives

$$\begin{aligned}\mathbf{x}_{k+1} &= \arg\min_{\mathbf{x}\in\Omega} \left\langle \mathbf{A}^\top\lambda_k, \mathbf{x}\right\rangle + \left\langle \beta\mathbf{A}^\top(\mathbf{A}\mathbf{x}_k - \mathbf{e}_{k+1}), \mathbf{x}-\mathbf{x}_k\right\rangle + \frac{\beta\eta}{2}\|\mathbf{x}-\mathbf{x}_k\|^2 \\ &= \arg\min_{\mathbf{x}\in\Omega} \frac{\beta\eta}{2}\|\mathbf{x} - C\|^2, \end{aligned} \tag{16}$$

where $\Omega := \left\{\mathbf{x}^\top\mathbf{x} = 1 | \mathbf{x}\in\mathbb{R}^n\right\}$, $C = \mathbf{x}_k - \frac{1}{\eta}\mathbf{A}^\top(\mathbf{A}\mathbf{x}_k - \mathbf{e}_{k+1}) - \frac{1}{\beta\eta}\mathbf{A}^\top\lambda^k$, and $\eta > \sigma_{max}(\mathbf{A}^\top\mathbf{A})$ is a proximal parameter. Therefore, we can get Equation 12 directly.

References

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

[2] S. Agarwal, N. Snavely, and S. M. Seitz. Fast algorithms for l_∞ problems in multiview geometry. In *Proc. CVPR*, pages 1–8, 2008.

[3] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. Building rome in a day. In *Proc. ICCV*, 2009.

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

[5] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, 2011. ISSN 1935-8237.

[6] M. Brand, M. Antone, and S. Teller. Spectral solution of large-scale extrinsic camera calibration as a graph embedding problem. In *Proc. ECCV*, 2004.

[7] E. J. Candes and T. Tao. Decoding by linear programming. *Information Theory, IEEE Transactions on*, 51(12):4203–4215, 2005.

[8] A. Chatterjee and V. M. Govindu. Efficient and robust large-scale rotation averaging. In *Proc. ICCV*, pages 521–528, 2013.

[9] J. Courchay, A. S. Dalalyan, R. Keriven, and P. Sturm. Exploiting loops in the graph of trifocal tensors for calibrating a network of cameras. In *Proc. ECCV*, pages 85–99, 2010.

[10] D. Crandall, A. Owens, N. Snavely, and D. P. Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. In *Proc. CVPR*, pages 3001–3008, 2011.

[11] A. Dalalyan and R. Keriven. L1-penalized robust estimation for a class of inverse problems arising in multiview geometry. In *NIPS*, 2009.

[12] O. Enqvist, F. Kahl, and C. Olsson. Non-sequential structure from motion. In *Workshop on Omnidirectional Vision, Camera Networks and Non-Classical Cameras*, 2011.

[13] L. Ferraz, X. Binefa, and F. Moreno-Noguer. Very fast solution to the pnp problem with algebraic outlier rejection. In *Proc. CVPR*, pages 501–508, 2014.

[14] A. Fitzgibbon and A. Zisserman. Automatic camera recovery for closed or open image sequences. *Proc. ECCV*, pages 311–326, 1998.

[15] T. Goldstein and S. Osher. The split bregman method for l1-regularized problems. *SIAM Journal on Imaging Sciences*, 2(2):323–343, 2009.

[16] V. M. Govindu. Combining two-view constraints for motion estimation. In *Proc. CVPR*, pages 218–225, 2001.

[17] V. M. Govindu. Lie-algebraic averaging for globally consistent motion estimation. In *Proc. CVPR*, 2004.

[18] V. M. Govindu. Robustness in motion averaging. In *Proc. ACCV*, 2006.

[19] S. Haner and A. Heyden. Covariance propagation and next best view planning for 3d reconstruction. In *Proc. ECCV*, pages 545–556, 2012.

[20] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[21] R. Hartley, J. Trumpf, Y. Dai, and H. Li. Rotation averaging. *IJCV*, pages 1–39, 2013.

[22] M. Havlena, A. Torii, J. Knopp, and T. Pajdla. Randomized structure from motion based on atomic 3d models from camera triplets. In *Proc. CVPR*, pages 2874–2881, 2009.

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

[24] F. Kahl and R. Hartley. Multiple view geometry under the l_∞-norm. *IEEE Trans. PAMI*, 30:1603–1617, 2007.

[25] L. Kneip, M. Chli, and R. Siegwart. Robust real-time visual odometry with a single camera and an imu. In *Proc. BMVC*, pages 1–11, 2011.

[26] M. Lhuillier and L. Quan. A quasi-dense approach to surface reconstruction from uncalibrated images. *IEEE Trans. PAMI*, 27(3):418–433, 2005.

[27] H. Li. Efficient reduction of l-infinity geometry problems. In *Proc. CVPR*, pages 2695–2702, 2009.

[28] H. Li and R. Hartley. Five-point motion estimation made easy. In *Proc. ICPR*, pages 630–633, 2006.

[29] Z. Lin, M. Chen, and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055*, 2010.

[30] Z. Lin, R. Liu, and Z. Su. Linearized alternating direction method with adaptive penalty for low-rank representation. In *Advances in neural information processing systems*, pages 612–620, 2011.

[31] D. Martinec and T. Pajdla. Robust rotation and translation estimation in multiview reconstruction. In *Proc. CVPR*, pages 1–8, 2007.

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

[33] D. Nistér. An efficient solution to the five-point relative pose problem. *IEEE Trans. PAMI*, 26:756–777, 2004.

[34] C. Olsson, A. Eriksson, and F. Kahl. Efficient optimization for l_∞ problems using pseudoconvexity. In *Proc. ICCV*, 2007.

[35] C. Olsson, A. Eriksson, and R. Hartley. Outlier removal using duality. In *Proc. CVPR*, pages 1450–1457, 2010.

[36] O. Özyesil and A. Singer. Robust camera location estimation by convex programming. In *Proc. CVPR*, 2015.

[37] N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):123–231, 2013.

[38] M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual modeling with a hand-held camera. *IJCV*, 59:207–232, 2004.

[39] M. Pollefeys, D. Nistér, J-M Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S-J Kim, P. Merrell, et al. Detailed real-time urban 3d reconstruction from video. *IJCV*, 78(2-3):143–167, 2008.

[40] C. Rother. *Multi-View Reconstruction and Camera Recovery using a Real or Virtual Reference Plane*. PhD thesis, January 2003.

[41] K. Sim and R. Hartley. Recovering camera motion using l_∞ minimization. In *Proc. CVPR*, 2006.

[42] S. Sinha, D. Steedly, and R. Szeliski. A multi-stage linear approach to structure from motion. In *ECCV Workshop on Reconstruction and Modeling of Large-Scale 3D Virtual Environments*, 2010.

[43] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. *ACM Trans. on Graph.*, 25:835–846, 2006.

[44] N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *IJCV*, 80(2):189–210, 2008.

[45] C. Strecha, W. von Hansen, L. Van Gool, P. Fua, and U. Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *Proc. CVPR*, 2008.

[46] K. Wilson and N. Snavely. Robust global translations with 1dsfm. In *Proc. ECCV (3)*, pages 61–75, 2014.

[47] C. Wu. Towards linear-time incremental structure from motion. In *Proc. 3DV*, 2013.

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