A Global Linear Method for Camera Pose Registration

Nianjuan Jiang[1,*] Zhaopeng Cui[2,*] Ping Tan[2]
[1]Advanced Digital Sciences Center, Singapore [2]National University of Singapore

Abstract

We present a linear method for global camera pose registration from pairwise relative poses encoded in essential matrices. Our method minimizes an approximate geometric error to enforce the triangular relationship in camera triplets. This formulation does not suffer from the typical 'unbalanced scale' problem in linear methods relying on pairwise translation direction constraints, i.e. an algebraic error; nor the system degeneracy from collinear motion. In the case of three cameras, our method provides a good linear approximation of the trifocal tensor. It can be directly scaled up to register multiple cameras. The results obtained are accurate for point triangulation and can serve as a good initialization for final bundle adjustment. We evaluate the algorithm performance with different types of data and demonstrate its effectiveness. Our system produces good accuracy, robustness, and outperforms some well-known systems on efficiency.

1. Introduction

Structure-from-motion (SfM) methods simultaneously estimate scene structure and camera motion from multiple images. Conventional SfM systems often consist of three steps. First, relative poses between camera pairs or triplets are computed from matched image feature points, e.g. by the five-point [25, 23] or six-point [32, 40] algorithm. Second, all camera poses (including orientations and positions) and scene point coordinates are recovered in a global coordinate system according to these relative poses. If camera intrinsic parameters are unknown, self-calibration algorithms, e.g. [30], should be applied. Third, a global nonlinear optimization algorithm, e.g. bundle adjustment (BA) [41], is applied to minimize the reprojection error, which guarantees a maximum likelihood estimation of the result.

While there are well established theories for the first and the third steps, the second step in existing systems are often ad-hoc and heuristic. Some well-known systems, such as [36, 2], compute camera poses in an incremental fashion, where cameras are added one by one to the global coordinate system. Other successful systems, e.g. [11, 22, 17], take a hierarchical approach to gradually merge short sequences or partial reconstructions. In either case, intermediate BA is necessary to ensure successful reconstruction. However, frequent intermediate BA causes reconstruction inefficiency, and the incremental approach often suffers from large drifting error. Thus, it is highly desirable that all camera poses are solved simultaneously for efficiency and accuracy. There are several interesting pioneer works in this direction, e.g. [13, 19, 24, 46]. More recently, Sinha et al. [35] designed a robust multi-stage linear algorithm to register pairwise reconstructions with some compromise in accuracy. Arie-Nachimson et al. [3] derived a novel linear algorithm that is robust to different camera baseline lengths. Yet it still suffers from the same degeneracy as [13] for collinear cameras (e.g. cameras along a street).

This paper presents a novel robust linear method. Like most solutions, we first calculate the camera orientation (rotations), e.g., using the method described in [24]. Unlike earlier algebraic methods, we compute the camera positions (translations) by minimizing a geometric error – the Euclidean distance between the camera centers and the lines collinear with their corresponding baselines. This novel approach generates more precise results, and does not degenerate with collinear camera motion. We want to stress that the robustness with collinear motion is an important advantage, since collinear motion is common (e.g., streetview images). Furthermore, our estimation of camera poses does not involve reconstructing any 3D point. Effectively, we first solve the 'motion' – camera poses, and then solve the 'structure' – scene points. This separation is advantageous, because there are much fewer unknowns in camera poses. Our algorithm is highly efficient and can be easily scaled up as a result of this separation. Once the camera poses are recovered, the scene points can be reconstructed from nearby cameras.

In the special case of three cameras, our algorithm effectively computes the trifocal tensor from three essential matrices. In our experiment, we find that our method is more robust than the four-point algorithm [26] which solves trifocal tensor from three calibrated images.

[*] These authors contributed equally to this work.

Disambiguate 3D Reconstruction. Modern SfM systems (e.g. [36, 2]) can reconstruct unordered internet images, while conventional methods are mainly designed for sequentially captured images. SfM with internet images opens a new door in 3D vision. One of its key challenges is to deal with incorrect epipolar geometries (EG) arising from suspicious feature matchings, especially for scenes with repetitive structures. Incorrect EGs cause ambiguity in 3D reconstruction – multiple valid yet different 3D reconstructions can be obtained from the same set of images. Significant efforts [44, 45, 33, 18] have been put to solve this ambiguity. Our method is applicable to both sequential and unordered image sets, though we do not address the ambiguity in this paper. Instead, we design a robust pipeline to recover a particular valid 3D reconstruction. It is straightforward to combine our method with [18], which evaluates multiple different 3D reconstructions and chooses the optimal one, to solve the ambiguity.

2. Related Work

Conventional Approach. Many well-known SfM systems take a sequential [31, 36, 2] or hierarchical [11, 22, 17] approach to register cameras incrementally to a global coordinate system from their pairwise relative poses. However, frequent intermediate BA is required for both types of methods to minimize error accumulation, and this results in computation inefficiency.

Factorization. Factorization based 3D reconstruction was proposed by Tomasi and Kanade [39] to recover all camera poses and 3D points simultaneously under weak perspective projection. This was further extended to more general projection models in [38]. However, the presence of missing data (often structured) and outliers poses theoretical challenges for both low-rank matrix approximation [6] and matrix factorization [5, 20].

Global Methods. Some global methods solve all camera poses together in two steps. Typically, they first compute camera rotations and solve translations in the next step. Our method belongs to this category. While global rotations can be computed robustly and accurately by rotation averaging [15], translations are difficult because the input pairwise relative translations are only known up to a scale. The pioneer works [13, 4] solved translations according to linear equations derived from pairwise relative translation directions. These earlier methods suffer from degeneracy of collinear camera motion and unbalanced constraint weighting caused by different camera baseline length. When distances between cameras are known beforehand, Govindu [14] provided an algebraic framework for motion averaging. For relatively small scale data, Courchay et al. [7] computed homographies to glue individual triplet reconstructions by loop analysis and nonlinear optimization. Sinha et al. [35] registered individual pairwise reconstructions by solving their individual global scaling and translation in a robust linear system. As reported in [3], this method generates less accurate results. Arie-Nachimson et al. [3] derived a highly efficient linear solution of translations from a novel decomposition of the essential matrix. This method is more robust to different baseline lengths between cameras. However, it still suffers from the degeneracy of collinear camera motion like [13, 4]. Unlike the previous algebraic methods, we derive our linear solution from an approximate geometric error, which does not suffer from such degeneracy and produces superior results.

Other global methods solve all camera poses and 3D scene points at once. Kahl [19] used L_∞-norm to measure the reprojection error of a reconstruction, which leads to a quasi-convex optimization problem. Later works along this line proposed to speed up the computation by selecting only representative points from image pairs [24], using fast optimization algorithms [29, 1], or customized cost function and optimization procedure [46]. It is also well known that L_∞-norm is highly sensitive to outliers. Therefore, careful outlier removal is required for the algorithm stability [9, 28].

There are also methods exploiting coarse or partial 3D information as initialization. For instance, with the aid of GPS, city scale SfM can be solved under the MRF framework [8].

Trifocal Tensor. In the special case of three cameras, the camera geometry is fully captured by a trifocal tensor. Trifocal tensors can be computed by the four-point algorithm [26] or the six-point algorithm [32, 40] from calibrated or uncalibrated images respectively. Trifocal tensors can also be estimated from three fundamental matrices [34] in the uncalibrated case. Effectively, our method provides a linear solution for trifocal tensor from three essential matrices (i.e. the calibrated case).

3. Overview

We first derive our algorithm under the assumption of known EGs without gross error. Later, this assumption is relaxed to deal with incorrect EGs with large error in Section 5.

The input to our system are essential matrices between image pairs, which are computed by the five-point algorithm [25]. An essential matrix E_{ij} between two images i, j provides the relative rotation R_{ij} and the translation direction t_{ij}. Here, R_{ij} is a 3×3 orthonormal matrix and t_{ij} is a 3×1 unit vector. Our goal is to recover all the absolute camera poses in a global coordinate system. We use a rotation matrix R_i and a translation vector c_i to denote the orientation and position of the i-th camera ($1 \leq i \leq N$). Ideally, the following equations should hold

$$R_j = R_{ij} R_i, \qquad R_j(c_i - c_j) \simeq t_{ij}. \qquad (1)$$

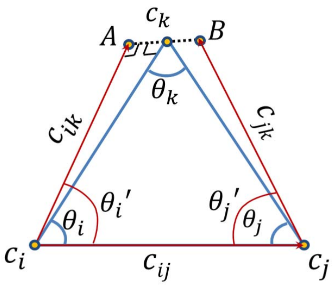


Figure 1. Geometric explanation of Equation (3).

Here, $\simeq$ means equality up to a scale. In real data, these equations will not hold precisely and we need to find a set of R_i, c_i that best satisfy these equations.

We design our method based on two criteria. Firstly, the solution should be simple and efficient. Approximate solutions are acceptable, since a final BA will be applied. Secondly, the camera poses should be solved separately from the scene points. There are often much more scene points than cameras so that solving camera poses without scene points will significantly reduce the number of unknowns.

We first apply the linear method described in [24] to compute the global camera rotations R_i. We find it provides good results in experiments, though a more sophisticated method [15] might be used. Basically, it over-parameterizes R_i by ignoring the orthonormal constraint on its column vectors and solves all the rotation matrices at once from the linear equations $R_j = R_{ij} R_i$. Once all rotations are fixed, we then solve all camera centers (c_i, $1 \leq i \leq N$) without reconstructing any 3D point.

4. Translation Registration

Given the global camera rotations computed in the previous section, we first transform each t_{ij} to the global rotation reference frame as $c_{ij} = -R_j^\top t_{ij}$. The constraint on camera centers in Equation (1) can be written as in [13],

$$c_{ij} \times (c_j - c_i) = 0. \tag{2}$$

Here, × is the cross product. This is a linear equation about the unknown camera centers. However, equations obtained this way degenerate for collinear camera motion. Furthermore, as discussed in [13], equations for image pairs with larger baseline lengths are given larger weights. Careful iterative re-weighting is required for good results. In fact, Equation (2) minimizes the cross product between c_{ij} and the baseline direction $c_j - c_i$. Minimizing such an algebraic error [16] is known to be sub-optimal in many 3D vision problems. In the following, we derive a linear algorithm that minimizes an approximate geometric error.

4.1. Triplet Translation Registration

We begin with the special case of three cameras. The relative translation c_{ij}, c_{ik}, and c_{jk} between camera pairs are known. We need to estimate camera centers c_i, c_j, and c_k. Ideally, the three unit vectors c_{ij}, c_{ik}, and c_{jk} should be coplanar. However, various measurement noise often makes them non-coplanar in real data, i.e. $(c_{ij}, c_{ik}, c_{jk}) \neq 0$. Here, $(\cdot, \cdot, \cdot)$ is the scalar triple product.

We first consider c_{ij} as perfect and minimize the Euclidean distance between c_k and the two lines $l(c_i, c_{ik})$ and $l(c_j, c_{jk})$. Here, $l(p, q)$ is the line passing through a point p with the orientation q. Due to measurement noise, $l(c_i, c_{ik})$ and $l(c_j, c_{jk})$ generally are non-coplanar. Thus, the optimal solution c_k lies on the midpoint of their common perpendicular AB as shown in Figure 1. In the following, we show that the optimal position c_k can be calculated as

$$c_k \approx \frac{1}{2}\left(\left(c_i + s_{ij}^{ik}||c_i - c_j||c_{ik}\right) + \left(c_j + s_{ij}^{jk}||c_i - c_j||c_{jk}\right)\right). \tag{3}$$

Here, $||c_i - c_j||$ is the distance between c_i and c_j. $s_{ij}^{ik} = \sin(\theta_j')/\sin(\theta_k') = ||c_i - c_k||/||c_i - c_j||$ and $s_{ij}^{jk} = \sin(\theta_i')/\sin(\theta_k') = ||c_j - c_k||/||c_i - c_j||$ are effectively the baseline length ratios. The angles are depicted in Figure 1. θ_k' is the angle between c_{ik} and c_{jk}. Please refer to the Appendix A for a derivation of this equation.

Equation (3) is nonlinear about the unknown camera centers. To linearize it, we observe that

$$||c_i - c_j||c_{ik} = ||c_i - c_j||R_i(\theta_i')c_{ij} = R_i(\theta_i')(c_j - c_i). \tag{4}$$

Here, $R_i(\phi)$ is the rotation matrix around the axis $c_{ij} \times c_{ik}$ for an angle ϕ (counter-clockwise). Thus we obtain the following linear equation,

$$2c_k - c_i - c_j = R_i(\theta_i')s_{ij}^{ik}(c_j - c_i) + R_j(-\theta_j')s_{ij}^{jk}(c_i - c_j). \tag{5}$$

Note $R_j(\cdot)$ is a rotation matrix around the direction $c_{ij} \times c_{jk}$. Similarly, we can obtain the following two linear equations of camera centers by assuming c_{ik} and c_{jk} are free from error respectively,

$$2c_j - c_i - c_k = R_i(-\theta_i')s_{ik}^{ij}(c_k - c_i) + R_k(\theta_k')s_{ik}^{jk}(c_i - c_k), \tag{6}$$

$$2c_i - c_j - c_k = R_j(\theta_j')s_{jk}^{ij}(c_k - c_j) + R_k(-\theta_k')s_{jk}^{ik}(c_j - c_k). \tag{7}$$

Solving these three linear equations can determine the camera centers. Note that Equation (5) does not require the orientation $c_j - c_i$ to be the same as c_{ij}. This introduces a rotation ambiguity in the plane defined by the camera centers. We can solve it by computing the average rotation to align $c_j - c_i$, $c_k - c_i$ and $c_k - c_j$ with the projection of c_{ij}, c_{ik} and c_{jk} in the camera plane, respectively, after the initial registration.

Collinear Camera Motion. Calculating baseline length ratios by the sine angles as described earlier is only valid when c_{ij}, c_{ik} and c_{jk} are not collinear. In order to be robust regardless of the type of camera motion, we compute all baseline length ratios from locally reconstructed scene points. Suppose a 3D scene point X is visible in all the three images. From the pairwise reconstruction with image i, j, we compute its depth d_j^{ij} in the image j while assuming unit baseline length. Similarly, we can calculate d_j^{jk} which is the depth of X in the image j from the reconstruction of image j, k. The ratio s_{jk}^{ij} is then estimated as d_j^{jk}/d_j^{ij}. In general, we have more than one scene points visible in all three images. We discard distant points and use RANSAC[10] to compute an average ratio. Note we only require local pairwise reconstructions to obtain baseline length ratios. The translation registration does not involve reconstructing any scene point in the global coordinate system.

4.2. Multi-view Translation Registration

Our method can be applied directly to register multiple cameras. Given a triplet graph (see definition in Section 5), we collect all equations (i.e. Equation [5–7]) from its triplets and solve the resulting sparse linear system $\mathbf{A}\mathbf{c} = 0$. Here, $\mathbf{c}$ is a vector formed by concatenating all camera centers. $\mathbf{A}$ is the matrix formed by collecting all the linear equations. The solution is a none trivial null vector of the matrix $\mathbf{A}$, and is given by the eigenvector associated with the fourth smallest eigenvalue of $\mathbf{A}^\top\mathbf{A}$. The eigenvectors associated with the three zero eigenvalues correspond to the three degrees of freedom of the origin of the world coordinate system. In the special case where all cameras are coplanar (i.e. the rotation ambiguity in all triplets share the same rotation axis), there is a global in-plane rotation ambiguity similar to the three-camera case. We can use the same method described before to compute this rotation.

In practice, every image participates in a different number of triplets. Therefore, the unknown camera centers are implicitly given different weights depending on the number of constraints containing that particular camera when we solve for $\mathbf{A}\mathbf{c} = 0$. Thus, for every camera i, we count the number of triplet constraints containing its center, denoted by $\mathcal{K}_i$. Each triplet constraint involving camera i, j, k is re-weighted by $\frac{1}{\min(\mathcal{K}_i,\mathcal{K}_j,\mathcal{K}_k)}$. This generates more stable results in practice.

5. Generalization to EG Outliers

The method described in Section 3 and Section 4 is applicable when there is no gross error in the pairwise epipolar geometries (EGs). However, many image sets, especially unordered internet images, can generate incorrect EGs with large error due to suspicious feature matching, especially for scenes with repetitive structures. Incorrect EGs result in wrong estimation of rotations and translations. We take the following steps to build a robust system.

Match Graph Construction. For each input image, we find its 80 nearest neighbors by the method described in [27]. The five-point algorithm [25] can compute EGs between these images. We then build a 'match graph', where each image is a vertex, and two vertices are connected if an EG can be computed between them. We only reconstruct the largest connected component of the match graph.

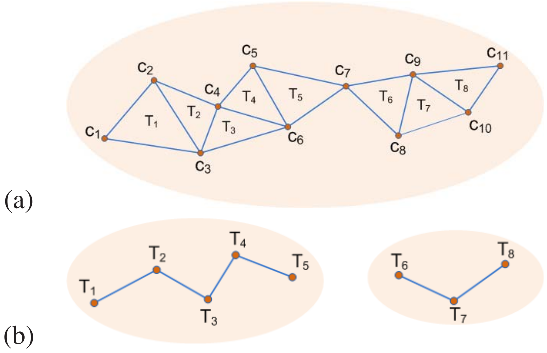


Figure 2. (a) A connected component of the match graph. (b) The two corresponding connected triplet graphs.

EG Verification. We perform various verifications to identify incorrect EGs. This involves several steps. 1) We verify every triplet in the match graph, and remove EGs which participate in no triplet that passes the verification. Specifically, we apply our translation registration to each triplet and calculate the average difference between the relative translation directions before and after the registration. If this average difference is larger than a threshold δ_1, we consider the verification fails. We further require that at least one good point (with reprojection error smaller than 4 pixels) can be triangulated by the registered triplet cameras. 2) Among the edges of the match graph, we extract a subset of 'reliable edges' to compute the global camera orientations as described in Section 3. We first weight each edge by its number of correspondences and take the maximum spanning tree. We then go through all the valid triplets. If two edges of a triplet are in the selected set of 'reliable edges', we insert its third edge as well. We iterate this insertion to include as many reliable edges as possible. 3) We further use these camera orientations to verify the match graph edges, and discard an edge if the geodesic distance [15] between the loop rotation matrix [45] and the identity matrix is greater than δ_2. Here, the loop rotation matrix in our case is simply $R_{ij}^\top R_j R_i^\top$. 4) Finally, we only consider the largest connected component of the remaining match graph. Typically, δ_1 and δ_2 is set to 3° and 5° respectively.

Connected Triplet Graph. We further extract connected triplet graphs from the match graph, where each triplet is represented by a vertex. Two vertices are connected if their triplets have a common edge in the match graph. A single connected component of the match graph could generate multiple connected triplet graphs, as illustrated in Figure 2. We then apply our method in Section 4 to compute the positions of cameras in each triplet graph respectively. We triangulate 3D scene points from feature tracks after solv-

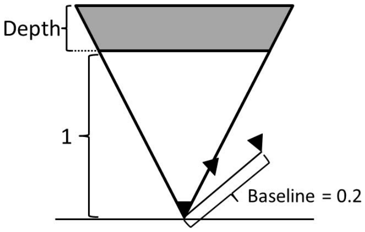


Figure 3. The test geometry used in comparison with the four-point algorithm [26].

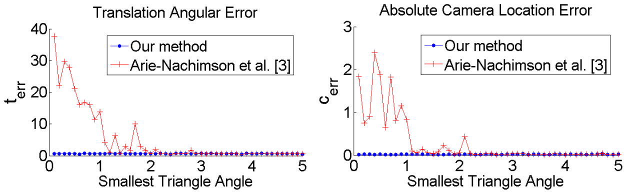


Figure 5. Comparison with [3]. Our method is much more stable in translation estimation for near collinear camera motions.

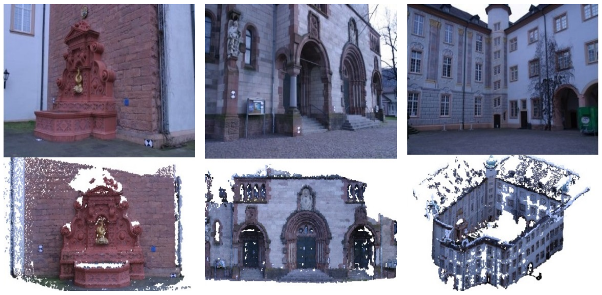
ing the camera positions. When there are multiple triplet graphs, their reconstructions are merged to obtain the final result. Specifically, we take their matched features to perform a 3D-to-3D registration for this merge.

6. Experiments

We verify our algorithm with various different experiments. We conduct our experiments on a 64-bit windows platform with Intel Xeon processor E5-2665, and 16 threads enabled. We parallelized the geometric verification of camera triplets. The ARPACK [21] is used to solve the sparse eigenvalue problem and PBA [43] is used for the final bundler adjustment.

6.1. Trifocal Tensor Estimation

We first evaluate our method with three synthetic input images with known ground truth to quantitatively evaluate our method. We use a similar test geometry as in [26] (shown in Figure 3). Camera 0 is placed at the world origin and camera 2 is placed at a random location away from camera 0 by 0.2 unit. The location of camera 1 is sampled randomly in the sphere centered at the middle point between camera 0 and 2, and passing through their camera centers. We further require the distance between any two cameras to be greater than 0.05 unit (which ensures the baseline length between any two cameras is not too small with respect to the scene distance, which is 1 unit here). The scene points are generated randomly within the viewing volume of the first camera and the distance between the nearest scene point and the furthest scene point is about 0.5 unit. The dimension of the synthetic image is 352×288 pixels and the field of view is 45°. Pairwise EG is computed using the five-point algorithm [25]. Zero mean Guassian noise is added to the image coordinates of the projected 3D points.



(a) (b) (c)

Figure 6. Input images and reconstructed point clouds of (a) *fountain-P11*, (b) *Herz-Jesu-P25*, (c) *castle-P30*.

We evaluate the reconstruction accuracy with three metrics. The error of camera orientations R_{err} is the mean geodesic distance (in degrees) between the estimated and the true camera rotation matrix. Translation angular error t_{err} is the mean angular difference between the estimated and the true baseline directions. Absolute camera location error c_{err} is the mean Euclidean distance between the estimated and the true camera centers. All these metrics reported below are the average results of 50 random trials.

Comparison with [26]. We compare with the four-point algorithm [26], which is the only practical algorithm to compute trifocal tensor from three calibrated images as far as we know. The reconstruction accuracy of both methods under different amount of noise is shown in Figure 4, where the horizontal axis shows the standard deviation of the Gaussian noise. Our linear algorithm outperforms the four-point algorithm in all metrics under various noise levels. It could be that the complex non-linear formulation in [26] makes their optimization harder to get good results.

Comparison with [3]. We also compare with the recent method [3] to demonstrate the robustness of our method on near collinear camera motions. Here, we generate c_0 and c_2 as described before. We sample c_1 along a random direction spanning an angle of 0.1 to 5 degrees with the line c_0c_2. Its location on that direction is randomly sampled while ensuring the angle $\angle c_1c_0c_2$ is the smallest angle in the triangle $c_0c_1c_2$. Gaussian noise with standard deviation of 0.5 pixels is used. The reconstruction accuracy is reported in Figure 5. It is clear that our method produces more stable results for near collinear motion.

6.2. Multi-view Reconstruction

We test the performance of our method with some standard benchmark datasets with known ground-truth camera motion to quantitatively evaluate the reconstruction accuracy. We also experiment with some relatively large scale image collections (sequential and unordered) to evaluate its scalability and robustness.

Evaluation on Benchmark Dataset. We compare our

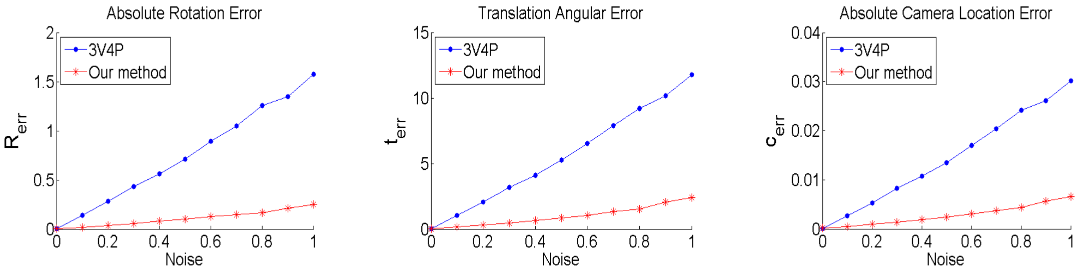


Figure 4. Comparison with the four-point algorithm [26] (3V4P). Our method generates better results in all the three metrics.

	fountain-P11		*Herz-Jesu-P25*		*castle-P30*	
	c_{err}	R_{err}	c_{err}	R_{err}	c_{err}	R_{err}
Ours (L)	0.053	0.517	0.106	0.573	1.158	1.651
Ours	**0.014**	**0.195**	0.064	**0.188**	**0.235**	0.48
VisualSFM[42]	0.036	0.279	0.055	0.287	0.264	**0.398**
Arie-Nachimson et al.[3]	0.023	0.421	**0.048**	0.313	-	-
Sinha et al.[35]	0.132	-	0.254	-	-	-

Table 1. Reconstruction accuracy of the three benchmark datasets. The absolute camera rotation error R_{err} and camera location error c_{err} are measured in degrees and meters, respectively.

method with some well-known and recent works[1] on the benchmark datasets provided in [37]. All results reported are computed using calibration information extracted from the EXIF tags unless stated otherwise. By our linear method, the average reprojection error is about 2 pixels for *fountain-P11* and *Herz-Jesu-P25*, and 4 pixels for *castle-P30*, respectively. After the final BA, it is reduced to below 0.3 pixels for all three datasets. To provide a visual validation, we apply the CMVS algorithm [12] to reconstruct dense point clouds with our recovered camera parameters (after the final BA). The results are visualized in Figure 6.

Quantitatively, all methods produce equally good reconstruction using ground truth calibration. Table 1 summarizes the quantitative results given EXIF calibration. On average our method produces error in c_i about 0.3% of the distance between the two farthest cameras. The results of our linear solution before BA are provided as 'Ours(L)'. Our method provides good initialization for BA, and it gives better accuracy than [35] on all available reported results. As compared to VisualSFM and [3], our method produces better results on *fountain-P11*, and performs similarly on *Herz-Jesu-P25*. Results for *castle-P30* are only available from VisualSFM, and we achieves similar accuracy. Bundler [36] produces similar or slightly inferior results as compared to VisualSFM on these datasets.

To assess the robustness of our method with bad EXIF, we added different levels of Gaussian noise to the ground truth focal length of *fountain-P11*. The average rotation (and location) errors are $0.2°$, $0.2°$, and $0.2°$ ($0.028m$, $0.031m$, and $0.036m$) when the standard deviation is 5%, 15%, and 25% of the true focal length. This experiment demonstrates the robustness of our method to imprecise information in EXIF.

Scalability and Time Efficiency. We evaluate the scalability and efficiency of our method with four relatively large scale image collections. The *Building*[2] example consists of 128 sequentially captured images. Our method recovers the cameras correctly regardless of the presence of a small fraction of erroneous epipolar geometries arising from symmetric scene structures. The *Trevi Fountain* and *Pisa* example consist of 1259 and 481 images [3] downloaded from Flickr.com respectively. We also test our method with the publically available *Notre Dame* example. We use 568 images with which we can extract EXIF tags from and the largest connected component on the match graph consists of 371 views. Each internet image collection is reconstructed as one single connected triplet graph by our algorithm. For *Pisa*, we performed a second round of BA after removing points with large reprojection errors due to large feature matching ambiguity in the data. We list the detailed comparison with VisualSFM in Table 2 (the time for computing pairwise matching and EGs is excluded). For fair comparison, we use the same set of EGs for both methods.

As we can see, more than 90% of the computation time in VisualSFM is spent on BA. By avoiding all the intermediate BA, we are 3 to 13 times faster depending on the scale of the problem. The speed advantage is clearer on larger scale datasets. Typically, the average reprojection error is about 5 pixels by our linear initialization, and is reduced to 1 pixel after BA.

We further manually identify the set of common cameras registered by our method and VisualSFM, respectively, for the *Notre Dame* example, and compute the difference between the estimated camera motion. The average rotation difference is 0.3 degrees, and the average translation difference is 0.007 (when the distance between the two farthest camera is 1).

To provide a visual validation, we feed our reconstructed cameras to the CMVS [12] and visualize the dense reconstruction in Figure 7.

[1] The results by the method [3] are kindly provided by its authors. The results by the method [35] are cited from [3].

[2] The dataset is available from the author's website of [45].

[3] Images with irrelevant content or no EXIF tag are removed manually.

example and # of input images	# of reconstructed cameras		# of reconstructed points		running time (s)					
					registration		BA		total	
	Ours	[42]	Ours	[42]	Ours	[42]	Ours	[42]	Ours	[42]
Building (128)	128	128	91290	78100	6	5	11	57	17	62
Notre Dame (371)	362	365	103629	104657	29	37	20	442	49	479
Pisa (481)	479	480	134555	129484	17	12	52	444	69	456
Trevi Fountain (1259)	1255	1253	297766	292277	74	75	61	1715	135	1790

Table 2. Comparison with VisualSFM on relatively large scale image collections. The time for computing pairwise matching and EGs is excluded.

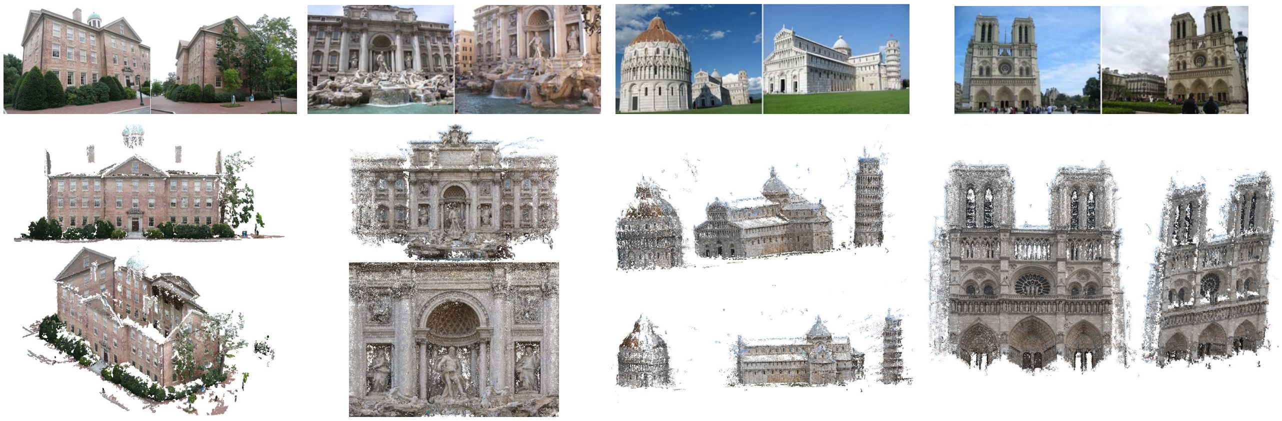

(a) (b) (c) (d)

Figure 7. Reconstruction results for relatively large scale datasets. (a) *Building*. (b) *Trevi Fountain*. (c) *Pisa*. (d) *Notre Dame*.

7. Conclusion

We present a novel linear solution for the global camera pose registration problem. Our method is derived by minimizing an approximate geometric error. It is free from the common degeneration of linear methods on collinear motion, and is robust to different baseline lengths between cameras. For the case of three cameras, it produces more accurate results than prior trifocal tensor estimation method on calibrated images. For general multiple cameras, it outperforms prior works on either accuracy, robustness or efficiency.

In our method, the rotation and translation are still estimated separately. It will be interesting to solve them together. The simplification of match graph and the selection of a subset of triplet constraints are important for even larger scale image collection, we will leave this for future study.

Acknowledgement

This study is supported by the HSSP research grant at the ADSC and PSF grant R-263-000-698-305 from Singapore's Agency for Science, Technology and Research (A*STAR).

References

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

[2] S. Agarwal, N. Snavely, I. Simon, S. Seitz, and R. Szeliski. Building rome in a day. In *Proc. ICCV*, 2009. 1, 2

[3] M. Arie-Nachimson, S. Z. Kovalsky, I. Kemelmacher-Shlizerman, A. Singer, and R. Basri. Global motion estimation from point matches. In *Proc. 3DPVT*, 2012. 1, 2, 5, 6

[4] M. Brand, M. Antone, and S. Teller. Spectral solution of large-scale extrinsic camera calibration as a graph embedding problem. In *Proc. ECCV*, 2004. 2

[5] A. Buchanan and A. Fitzgibbon. Damped newton algorithms for matrix factorization with missing data. In *Proc. CVPR*, pages 316–322, 2005. 2

[6] P. Chen and D. Suter. Recovering the missing components in a large noisy low-rank matrix: application to sfm. *IEEE Trans. PAMI*, 26(8):1051–1063, 2004. 2

[7] J. Courchay, A. 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. 2

[8] D. Crandall, A. Owens, N. Snavely, and D. Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. In *Proc. CVPR*, pages 3001–3008, 2011. 2

[9] A. Dalalyan and R. Keriven. L1-penalized robust estimation for a class of inverse problems arising in multiview geometry. In *NIPS*, 2009. 2

[10] M. Fischler and R. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 4

[11] A. Fitzgibbon and A. Zisserman. Automatic camera recovery for closed or open image sequences. *Proc. ECCV*, pages 311–326, 1998. 1, 2

[12] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. Towards internet-scale multi-view stereo. In *Proc. CVPR*, 2010. 6

[13] V. M. Govindu. Combining two-view constraints for motion estimation. In *Proc. CVPR*, pages 218–225, 2001. 1, 2, 3

[14] V. M. Govindu. Lie-algebraic averaging for globally consistent motion estimation. In *Proc. CVPR*, 2004. 2

[15] R. Hartley, J. Trumpf, Y. Dai, and H. Li. Rotation averaging. *IJCV*, pages 1–39, 2013. 2, 3, 4

[16] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. 3

[17] 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. 1, 2

[18] N. Jiang, P. Tan, and L. Cheong. Seeing double without confusion: Structure-from-motion in highly ambiguous scenes. In *Proc. CVPR*, pages 1458–1465, 2012. 2

[19] F. Kahl. Multiple view geometry and the l_∞ norm. In *Proc. ICCV*, 2005. 1, 2

[20] Q. Ke and T. Kanade. Robust l1 norm factorization in the presence of outliers and missing data by alternative convex programming. In *Proc. CVPR - Volume 1*, pages 739–746, 2005. 2

[21] R. Lehoucq and J. Scott. An evaluation of software for computing eigenvalues of sparse nonsymmetric matrices. *Preprint MCS-P547*, 1195, 1996. 5

[22] M. Lhuillier and L. Quan. A quasi-dense approach to surface reconstruction from uncalibrated images. *IEEE Trans. PAMI*, 27(3):418–433, 2005. 1, 2

[23] H. Li and R. Hartley. Five-point motion estimation made easy. In *Proc. ICPR*, pages 630–633, 2006. 1

[24] D. Martinec and T. Pajdla. Robust rotation and translation estimation in multiview reconstruction. In *Proc. CVPR*, pages 1–8, 2007. 1, 2, 3

[25] D. Nistér. An efficient solution to the five-point relative pose problem. *IEEE Trans. PAMI*, 26:756–777, 2004. 1, 2, 4, 5

[26] D. Nistér and F. Schaffalitzky. Four points in two or three calibrated views: Theory and practice. *IJCV*, 67(2):211–231, 2006. 1, 2, 5, 6

[27] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *Proc. CVPR*, 2006. 4

[28] C. Olsson, A. Eriksson, and R. Hartley. Outlier removal using duality. In *Proc. CVPR*, pages 1450–1457, 2010. 2

[29] C. Olsson, A. Eriksson, and F. Kahl. Efficient optimization for l_∞ problems using pseudoconvexity. In *Proc. ICCV*, 2007. 2

[30] M. Pollefeys, R. Koch, and L. Gool. Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters. *IJCV*, 32(1):7–25, 1999. 1

[31] 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. 2

[32] L. Quan. Invariants of six points and projective reconstruction from three uncalibrated images. *IEEE Trans. PAMI*, 17(1):34–46, 1995. 1, 2

[33] R. Roberts, S. Sinha, R. Szeliski, and D. Steedly. Structure from motion for scenes with large duplicate structures. In *Proc. CVPR*, 2011. 2

[34] P.-Y. Shen, W. Wang, C. Wu, L. Quan, and R. Mohr. From fundamental matrix to trifocal tensor. In *Proc. SPIE*, volume 3454, pages 340–347, 1998. 2

[35] 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. 1, 2, 6

[36] N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. *ACM Trans. on Graph.*, 25:835–846, 2006. 1, 2, 6

[37] 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. 6

[38] P. F. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *Proc. ECCV (2)*, pages 709–720, 1996. 2

[39] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *IJCV*, 9:137–154, 1992. 2

[40] P. Torr and A. Zisserman. Robust parameterization and computation of the trifocal tensor. *Image and Vision Computing*, 15:591–605, 1997. 1, 2

[41] B. Triggs, P. Mclauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment - a modern synthesis. *Lecture Notes in Computer Science*, pages 298–375, 2000. 1

[42] C. Wu. Visualsfm: A visual structure from motion system. 2011. 6, 7

[43] C. Wu, S. Agarwal, B. Curless, and S. Seitz. Multicore bundle adjustment. In *Proc. CVPR*, pages 3057–3064, 2011. 5

[44] C. Zach, A. Irschara, and H. Bischof. What can missing correspondences tell us about 3d structure and motion? In *Proc. CVPR*, 2008. 2

[45] C. Zach, M. Klopschitz, and M. Pollefeys. Disambiguating visual relations using loop constraints. In *Proc. CVPR*, 2010. 2, 4, 6

[46] C. Zach and M. Pollefeys. Practical methods for convex multi-view reconstruction. In *Proc. ECCV: Part IV*, pages 354–367, 2010. 1, 2

Appendix A. Derivation of Equation (3) We first show that the length of the line segments c_iA, c_jB are approximately $s_{ij}^{ik}||c_i - c_j||$ and $s_{ij}^{jk}||c_i - c_j||$ respectively. The three vectors c_{ij}, c_{ik} and c_{jk} should be close to coplanar, so the angle $\angle Ac_ic_k$ is close to zero, and the length of c_iA is close to that of c_ic_k. We can calculate the length of c_ic_k as:

$$\frac{\sin(\theta_j)}{\sin(\theta_k)}||c_i - c_j|| \approx \frac{\sin(\theta_j')}{\sin(\theta_k')}||c_i - c_j|| = s_{ij}^{ik}||c_i - c_j||.$$

Note that $\theta_j' \approx \theta_j, \theta_k' \approx \theta_k$ because the three vectors c_{ij}, c_{ik} and c_{jk} are close to coplanar. The 3D coordinate of A is then approximated by $c_i + s_{ij}^{ik}||c_i - c_j||c_{ik}$. Similarly, we can obtain the coordinate of B as $c_j + s_{ij}^{jk}||c_i - c_j||c_{jk}$. As a result, the coordinate of c_k, which is the midpoint of AB, can be computed by Equation (3).