Robust Camera Location Estimation by Convex Programming

Onur Özyeşil[1] and Amit Singer[1,2]
[1]Program in Applied and Computational Mathematics, Princeton University
[2]Department of Mathematics, Princeton University
Princeton, NJ 08544-1000, USA
{oozyesil,amits}@math.princeton.edu

Abstract

3D structure recovery from a collection of 2D images requires the estimation of the camera locations and orientations, i.e. the camera motion. For large, irregular collections of images, existing methods for the location estimation part, which can be formulated as the inverse problem of estimating n locations $\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_n$ in $\mathbb{R}^3$ from noisy measurements of a subset of the pairwise directions $\frac{\mathbf{t}_i - \mathbf{t}_j}{\|\mathbf{t}_i - \mathbf{t}_j\|}$, are sensitive to outliers in direction measurements. In this paper, we firstly provide a complete characterization of well-posed instances of the location estimation problem, by presenting its relation to the existing theory of parallel rigidity. For robust estimation of camera locations, we introduce a two-step approach, comprised of a pairwise direction estimation method robust to outliers in point correspondences between image pairs, and a convex program to maintain robustness to outlier directions. In the presence of partially corrupted measurements, we empirically demonstrate that our convex formulation can even recover the locations exactly. Lastly, we demonstrate the utility of our formulations through experiments on Internet photo collections.

1. Introduction

Structure from motion (SfM) is the problem of recovering a 3D (stationary) structure by estimating the camera motion corresponding to a collection of 2D images of the same structure. Classically, SfM involves three steps: (1) Estimation of point correspondences between pairs of images, and relative pose estimation of camera pairs based on corresponding points (2) Estimation of camera motion, *i.e*. global camera orientations and locations, from relative poses (3) 3D structure recovery based on the estimated motion by reprojection error minimization (*e.g*., using the bundle adjustment algorithm of [33]). Although there exist accurate and efficient algorithms for the first and the third steps, existing methods for camera motion estimation, and specifically for the camera location estimation part, are usually sensitive to noise. The camera location estimation problem can be formulated as a specific case (for $d = 3$) of the inverse problem of estimating n locations $\mathbf{t}_1, \ldots, \mathbf{t}_n$ in $\mathbb{R}^d$ from a subset of (potentially noisy) measurements of the pairwise directions γ_{ij}, given by

$$\gamma_{ij} = \frac{\mathbf{t}_i - \mathbf{t}_j}{\|\mathbf{t}_i - \mathbf{t}_j\|} \tag{1}$$

(see Figure 1 for a noiseless instance of the problem). In terms of this formulation, misidentified point correspondences may manifest themselves (see §3) as direction measurements with large errors (*i.e*., outlier directions), and hence, may induce instability in location estimation.

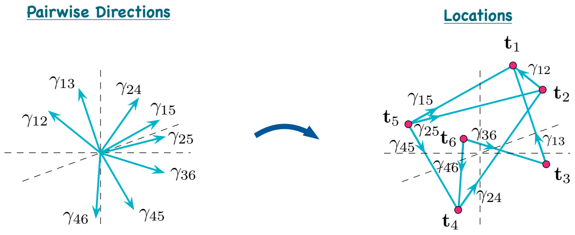


Figure 1. A (noiseless) instance of the location estimation problem in $\mathbb{R}^3$, with $n = 6$ locations and $m = 8$ pairwise directions.

Existing methods for SfM can roughly be classified into two main categories; incremental approaches (*e.g*. [1, 6, 11, 16, 31, 32, 44]), that integrate images to the estimation process one by one (or in small groups) and global methods, that aim to estimate the camera motion (and sometimes also the 3D structure) jointly for all images. Incremental methods are prone to accumulation of estimation errors at each step. On the other hand, for the global methods, since simultaneous estimation of motion and 3D structure is computationally expensive, a usual procedure is to estimate motion and structure separately. Given an accurate motion estimate, a single instance of reprojection error minimization is usually enough to obtain high quality structure estimates.

Since orientation estimation is a relatively well-posed problem, with several efficient and stable existing methods (*e.g.* [2, 7, 13, 14, 25, 27]), it is customary to estimate the locations separately (based on the orientation estimates). The works of [2, 5, 12], formulate the problem as finding a least squares solution to a linear system of equations derived from pairwise direction measurements (we refer to this method as the "least squares" (LS) solver). However, empirical observations (as in [27]) have pointed out the instability of the LS solver, in the form of a tendency to produce spurious solutions clustering around a few locations. The multistage linear method of [30] attempts to eliminate the clustering solutions by estimating the relative scales between cameras. The Lie algebraic averaging method of [13] is an efficient alternative, but may suffer from convergence to local minima. [29] formulates a quasi-convex method (based on iterative optimization of a functional of the ℓ_∞ norm). However, since the ℓ_∞ norm is prone to outlier directions, this method usually fails to produce accurate estimates (see, *e.g.*, [27]). A relatively accurate method, closely related to our formulation, is studied in [34, 35]. Based on minimizing the ℓ_2 norm of the error in direction measurements (linearized in $\mathbf{t}_i$'s), this method also employs constraints to eliminate clustering solutions (hence, we refer to this method as the "constrained least squares" (CLS) solver). However, in the presence of outlier directions, the accuracy of the CLS solver is degraded (see Figure 5). Another method closely related to our formulation, minimizing the ℓ_∞ norm of the error in direction measurements, is studied in [26]. The accuracy of this method is affected by the sensitivity of the ℓ_∞ norm to outlier directions. In [19], a global linear method, which uses triplets of images instead of pairwise directions, is studied. In the recent work [40], a preprocessing step (named 1DSfM, and designed to remove outlier directions), followed by a non-convex optimization method is introduced. Another recently introduced alternative is the "semidefinite relaxation" (SDR) solver of [27]. Formulated as an abstract problem to estimate locations from pairwise "lines" (*i.e.*, from measurements of $\pm\gamma_{ij}$, where the sign is unknown), this method aims to resolve the instability of the LS method by introducing extra non-convex constraints in the LS problem, and then relaxing them. However, semidefinite programming is computationally expensive for large data sets, and its accuracy is degraded in the presence of outlier lines (see Figure 5).

In this paper, we characterize well-posed instances of the camera location estimation problem, by presenting its relation to the existing results of parallel rigidity theory. For robust estimation of camera locations, we introduce a two-step formulation: robust estimation of pairwise directions (in the presence of outliers in point correspondences), and a convex program for robust estimation of camera locations in the presence of measurements corrupted by large errors, *i.e.* outlier directions. We provide empirical evaluation of our formulations using synthetic data, which demonstrate highly accurate location recovery performance compared to existing methods, and even exact location recovery in the presence of partially corrupted measurements (with sufficiently many noiseless directions). We also provide experimental results using real images, that present the accuracy and efficiency of our methods.

Notation: We denote vectors in $\mathbb{R}^d$, $d \geq 2$, in boldface. For $\mathbf{x} \in \mathbb{R}^d$, $\|\mathbf{x}\|$ denotes its Euclidean norm. S^d and $\mathrm{SO}(d)$ denote the (Euclidean) sphere in $\mathbb{R}^{d+1}$ and the special orthogonal group of rotations acting on $\mathbb{R}^d$, respectively. We use the hat accent, to denote estimates of our variables, as in $\hat{X}$ is the estimate of X. We use star to denote solutions of optimization problems, as in X^*. Lastly, we use the letters n and m to denote the number of locations $|V_t|$ and the number of edges $|E_t|$ of graphs $G_t = (V_t, E_t)$ that encode the pairwise direction information.

2. Location Estimation

The entire information of pairwise directions is represented using a measurement graph $G_t = (V_t, E_t)$, where the i'th node in $V_t = \{1, 2, \ldots, n\}$ corresponds to the location $\mathbf{t}_i$ and each edge $(i, j) \in E_t$ is endowed with the direction γ_{ij}. Provided with the set $\{\gamma_{ij}\}_{(i,j)\in E_t}$ of (noiseless) directions on $G_t = (V_t, E_t)$, we first study the problem of *unique realizability* of the locations. We will then introduce our robust formulation for location estimation from noisy pairwise directions.

2.1. Parallel Rigidity

The unique realizability of locations from (noiseless) pairwise directions was previously studied under the general title of *parallel rigidity theory* (see, *e.g.*, [9, 10, 20, 21, 28, 38, 39] and references therein). In the context of SfM, the implications of the parallel rigidity theory for the camera location estimation part were recognized in [27][1]. Here, we present a brief summary of fundamental results in parallel rigidity theory.

Provided with the noiseless pairwise directions $\{\gamma_{ij}\}_{(i,j)\in E_t} \subseteq S^{d-1}$ (termed a "formation"), we first consider the following fundamental questions: Can we uniquely realize $\{\mathbf{t}_i\}_{i\in V_t}$, of course, up to a global translation and scale (*i.e.* can we obtain a set of points *congruent* to $\{\mathbf{t}_i\}_{i\in V_t}$)? Is unique realizability a generic property of the measurement graph G_t (*i.e.* is it independent of the particular realization of the points, assuming they are in generic position) and can it be decided efficiently?

Certifying unique realizability of locations is more complicated, *e.g.*, compared to certifying uniqueness of camera orientations, which only requires (for arbitrary d)

[1] We note that, although the pairwise measurements studied in [27] are of the form $\pm\gamma_{ij}$ (where, the sign is unavailable), the results of parallel rigidity theory for unique realizability remain the same when the signs are given.

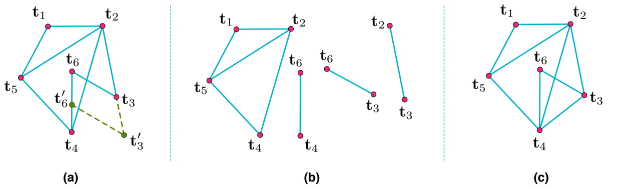


Figure 2. **(a)** A formation of 6 locations on a connected graph, which is parallel rigid in $\mathbb{R}^3$ but **not** parallel rigid in $\mathbb{R}^2$. Non-uniqueness in $\mathbb{R}^2$ is demonstrated by two non-congruent location solutions $\{\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3, \mathbf{t}_4, \mathbf{t}_5, \mathbf{t}_6\}$ and $\{\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}'_3, \mathbf{t}_4, \mathbf{t}_5, \mathbf{t}'_6\}$, each of which can be obtained from the other by an **independent rescaling** of the solution for one of its maximally parallel rigid components in $\mathbb{R}^2$, **(b)** Maximally parallel rigid components in $\mathbb{R}^2$, of the formation in (a), **(c)** A parallel rigid formation (in $\mathbb{R}^2$ and $\mathbb{R}^3$) obtained from the formation in (a) by adding the extra edge $(3, 4)$ linking its maximally parallel rigid components

the connectivity of the measurement graph (see Figure 2). On the other hand, parallel rigidity theory has a much simpler structure compared to the (classical) rigidity theory involving distance information (for a survey in rigidity theory, see [3]). The identification of parallel rigid formations is addressed in [9, 10, 38, 39, 20] (also see the survey [17]), where it is shown that parallel rigidity in $\mathbb{R}^d$ ($d \geq 2$) is a generic property of G_t that admits a complete combinatorial characterization:

Theorem 1 (Whiteley,1987)**.** *For a graph $G = (V, E)$, let $(d-1)E$ denote the set consisting of $(d-1)$ copies of each edge in E. Then, G is generically parallel rigid in $\mathbb{R}^d$ if and only if there exists a nonempty set $D \subseteq (d-1)E$, with $|D| = d|V| - (d+1)$, such that for all subsets D' of D,*

$$|D'| \leq d|V(D')| - (d+1)\ , \tag{2}$$

where $V(D')$ denotes the vertex set of the edges in D'.

The conditions of Theorem 1 can be used to design efficient algorithms (*e.g.*, adaptations of the pebble game algorithm [18], with a time complexity of $\mathcal{O}(n^2)$) for testing parallel rigidity. Also, [27] provides a randomized spectral test (having a time complexity of $\mathcal{O}(m)$) for testing parallel rigidity. Moreover, unique realizability turns out to be equivalent to parallel rigidity, for arbitrary d (see [9, 17, 20, 27, 39]).

For a formation that is not parallel rigid, the algorithms in [20, 21] can be used to decompose the graph into maximally parallel rigid components (*i.e.*, to obtain maximal subgraphs of G_t that can be uniquely realized).

The results of parallel rigidity theory are valid for *noiseless* directions. However, when provided with *noisy* directions (*e.g.*, computed from real images), instead of uniqueness of the solution of a specific camera location estimation algorithm, we consider the following question: Is there sufficient information for the location estimation problem to be *well-posed* (in the sense that, if direction measurement error is small enough, then locations can be estimated stably)? For formations which are not parallel rigid, instability results from independent scaling and translation of maximally rigid components. Hence, we consider problem instances on parallel rigid measurement graphs to be *well-posed*. As a result, given a (noisy) formation $\{\gamma_{ij}\}_{(i,j)\in E_t}$ on $G_t = (V_t, E_t)$, we firstly check for parallel rigidity of G_t, then, if the formation is not parallel rigid, we extract its maximally parallel rigid components (using the algorithm in [21]) and estimate the locations for the largest maximally parallel rigid component of G_t.

2.2. Robust Location Estimation

This section introduces our main formulation for robust location estimation. Suppose we are given a set of pairwise direction measurements $\{\gamma_{ij}\}_{(i,j)\in E_t} \subseteq S^{d-1}$, *i.e.*, for each $(i,j) \in E_t$, γ_{ij} satisfies

$$\gamma_{ij} = \frac{\mathbf{t}_i - \mathbf{t}_j}{\|\mathbf{t}_i - \mathbf{t}_j\|} + \epsilon^{\gamma}_{ij} \tag{3}$$

where, ϵ^{γ}_{ij} denotes the direction error. Our objective is to estimate the locations $\{\mathbf{t}_i\}_{i\in V_t}$ (from the directions $\{\gamma_{ij}\}_{(i,j)\in E_t}$) by maintaining robustness to outlier direction measurements (*i.e.*, γ_{ij}'s with large ϵ^{γ}_{ij}'s) in a computationally efficient manner. In this respect, we first rewrite (3) as

$$\begin{aligned} \mathbf{t}_i - \mathbf{t}_j &= \|\mathbf{t}_i - \mathbf{t}_j\|\gamma_{ij} + \epsilon^{\mathbf{t}}_{ij} && (4)\\ &= d_{ij}\gamma_{ij} + \epsilon^{\mathbf{t}}_{ij} && (5)\\ \iff \epsilon^{\mathbf{t}}_{ij} &= \mathbf{t}_i - \mathbf{t}_j - d_{ij}\gamma_{ij} && (6) \end{aligned}$$

where, $\epsilon^{\mathbf{t}}_{ij}$ denotes the displacement error, and we define $d_{ij} := \|\mathbf{t}_i - \mathbf{t}_j\|$ to rewrite $\epsilon^{\mathbf{t}}_{ij}$ linearly in $\mathbf{t}_i$, $\mathbf{t}_j$ and d_{ij}. Observe that, large errors in directions (*i.e.*, large ϵ^{γ}_{ij}'s) induce large displacement errors $\epsilon^{\mathbf{t}}_{ij}$'s. As a result, we can employ displacement error minimization as a substitute for direction error minimization for location estimation.

Hence, to maintain robustness to large $\epsilon^{\mathbf{t}}_{ij}$'s in (6), we choose to minimize the sum of *unsquared* norms of $\epsilon^{\mathbf{t}}_{ij}$'s. Also, for computational efficiency, we drop the intrinsic non-convex constraints $d_{ij} = \|\mathbf{t}_i - \mathbf{t}_j\|$ to obtain the convex "least unsquared deviations" (LUD) formulation

$$\begin{aligned} &\underset{\substack{\{\mathbf{t}_i\}_{i\in V_t}\subseteq\mathbb{R}^d \\ \{d_{ij}\}_{(i,j)\in E_t}}}{\text{minimize}} && \sum_{(i,j)\in E_t} \|\mathbf{t}_i - \mathbf{t}_j - d_{ij}\gamma_{ij}\| \\ &\text{subject to} && \sum_{i\in V_t} \mathbf{t}_i = \mathbf{0}\ ;\ d_{ij} \geq c,\ \forall (i,j) \in E_t \end{aligned} \tag{7}$$

where the constraints $\sum_i \mathbf{t}_i = \mathbf{0}$ and $d_{ij} \geq c$ remove the translational and the scale ambiguities of the solution, respectively (*wlog* we take $c = 1$)[2]. The constraints $d_{ij} \geq c$ are introduced to prevent trivial solutions of the form $d_{ij}^* \equiv 0$, $\mathbf{t}_i^* \equiv \mathbf{0}$, as well as solutions clustered around a few locations.

For well-posed instances of the location estimation problem (*i.e.*, for parallel rigid G_t), and in the presence of noiseless direction measurements (*i.e.*, $\epsilon_{ij}^{\gamma} \equiv \mathbf{0}$ in (3)), we expect the LUD and CLS solvers to recover the locations $\mathbf{t}_i$ exactly.

Proposition 1 (Exact Recovery in the Noiseless Case)**.** *Assume that the noiseless formation* $\{\gamma_{ij}\}_{(i,j)\in E_t}$*, corresponding to the locations* $\{\mathbf{t}_i\}_{i\in V_t}$ *(in general position), is parallel rigid. Then, the LUD (7) and CLS solvers recover the locations exactly, in the sense that any solution is congruent to* $\{\mathbf{t}_i\}_{i\in V_t}$*.*

Proof. Wlog, we assume $\min_{(i,j)\in E_t} \|\mathbf{t}_i - \mathbf{t}_j\| = 1$ and $\sum_i \mathbf{t}_i = \mathbf{0}$. Then, $\{\mathbf{t}_i\}_{i\in V_t}$ together with $d_{ij} = \|\mathbf{t}_i - \mathbf{t}_j\|$, $(i,j) \in E_t$, constitute an optimal solution for the LUD (7) and CLS problems, with zero cost value. Let, $\{\mathbf{t}_i'\}_{i\in V_t}$ and $\{d_{ij}'\}_{(i,j)\in E_t}$ be another solution of the LUD (7) and CLS problems, which must also have zero cost value. Then, for each $(i,j) \in E_t$, we get

$$\begin{aligned} &\|\mathbf{t}_i' - \mathbf{t}_j' - d_{ij}'\gamma_{ij}\| = 0 \iff \mathbf{t}_i' - \mathbf{t}_j' = d_{ij}'\gamma_{ij} \\ &\implies \frac{\mathbf{t}_i' - \mathbf{t}_j'}{\|\mathbf{t}_i' - \mathbf{t}_j'\|} = \frac{\mathbf{t}_i - \mathbf{t}_j}{\|\mathbf{t}_i - \mathbf{t}_j\|} \qquad (8) \end{aligned}$$

i.e., $\{\mathbf{t}_i'\}_{i\in V_t}$ induces a formation on G_t, which is parallel to the formation corresponding to $\{\mathbf{t}_i\}_{i\in V_t}$ on G_t. However, since G_t is parallel rigid, $\{\mathbf{t}_i'\}_{i\in V_t}$ has to be congruent to $\{\mathbf{t}_i\}_{i\in V_t}$ (in fact, $\mathbf{t}_i' = \alpha\mathbf{t}_i$, for $\alpha \geq 1$, by the feasibility of $\{\mathbf{t}_i', d_{ij}'\}$). □

2.3. Iteratively Reweighted Least Squares (IRLS)

In this section we formulate an iteratively reweighted least squares (IRLS) solver (see, *e.g.*, [8, 43]) for the LUD problem (7). The main idea of IRLS is to iteratively solve (successive smooth regularizations of) the convex problem by using quadratic programing (QP) approximations. A pseudo code version is provided in Algorithm 1 (where, we consider a single smooth regularization). At the r'th iteration, more emphasis is given to the directions that are better approximated by the estimates $\hat{\mathbf{t}}_i^r$'s and $\hat{d}_{ij}^r$'s. Also, the regularization parameter δ ensures that no single direction can attain unbounded influence. The iterations are repeated until a convergent behavior in the variables, and in the cost value of the problem is observed. We refer the reader to [4] for a proof of convergence of the IRLS solver (where, a sequence of smooth regularizations, with $\delta \searrow 0$, is assumed[3]).

Algorithm 1 Iteratively reweighted least squares (IRLS) algorithm for the LUD (7) solver

```
Initialize: $w_{ij}^0 = 1, \forall (i,j) \in E_t$
for $r = 0, 1, \ldots$ do
    • Compute $\{\hat{\mathbf{t}}_i^{r+1}\}, \{\hat{d}_{ij}^{r+1}\}$ by solving the QP:
        $\underset{\left\{\substack{\sum \mathbf{t}_i = \mathbf{0}, \\ d_{ij} \geq 1}\right\}}{\text{minimize}} \sum_{(i,j)\in E_t} w_{ij}^r \|\mathbf{t}_i - \mathbf{t}_j - d_{ij}\gamma_{ij}\|^2$
    • $w_{ij}^{r+1} \leftarrow \left(\left\|\hat{\mathbf{t}}_i^{r+1} - \hat{\mathbf{t}}_j^{r+1} - \hat{d}_{ij}^{r+1}\gamma_{ij}\right\|^2 + \delta\right)^{-1/2}$
```

3. Robust Pairwise Direction Estimation

We now present a pairwise direction estimation method designed to maintain robustness to outlier point correspondences between image pairs.

Let $\{\mathrm{I}_i\}_{i=1}^n$ be a collection of images of a stationary 3D scene. We use a pinhole camera model, and denote the orientations, locations, and focal lengths of the n cameras corresponding to these images by $\{R_i\}_{i=1}^n \subseteq \mathrm{SO}(3)$, $\{\mathbf{t}_i\}_{i=1}^n \subseteq \mathbb{R}^3$, and $\{f_i\}_{i=1}^n \subseteq \mathbb{R}^+$, respectively. Consider a scene point $\mathbf{P} \in \mathbb{R}^3$ represented in the i'th image plane by $\mathbf{p}_i \in \mathbb{R}^3$. To produce $\mathbf{p}_i$, $\mathbf{P}$ is firstly represented in the i'th camera's coordinate system by $\mathbf{P}_i = R_i^T(\mathbf{P} - \mathbf{t}_i) = (\mathbf{P}_i^x, \mathbf{P}_i^y, \mathbf{P}_i^z)^T$ and then projected onto the i'th image plane by $\mathbf{p}_i = (f_i/P_i^z)\mathbf{P}_i$. Note that, for the image I_i, we in fact observe $\mathbf{q}_i = (\mathbf{p}_i^x, \mathbf{p}_i^y)^T \in \mathbb{R}^2$ (*i.e.*, the coordinates on the image plane) as the measurement corresponding to $\mathbf{P}$.

For an image pair I_i and I_j, the essential matrix $E_{ij} = [\mathbf{t}_{ij}]_\times R_{ij}$ (where $R_{ij} = R_i^T R_j$ and $\mathbf{t}_{ij} = R_i^T(\mathbf{t}_j - \mathbf{t}_i)$ denote the pairwise rotation and translation, and $[\mathbf{t}_{ij}]_\times$ is the matrix of cross product with $\mathbf{t}_{ij}$) satisfies the "epipolar constraints" given by

$$\mathbf{p}_i^T E_{ij} \mathbf{p}_j = 0 \qquad (9)$$

$$\iff \begin{bmatrix} \mathbf{q}_i/f_i \\ 1 \end{bmatrix}^T E_{ij} \begin{bmatrix} \mathbf{q}_j/f_j \\ 1 \end{bmatrix} = 0 \qquad (10)$$

The estimates $\hat{R}_{ij}$ and $\hat{\mathbf{t}}_{ij}$, computed from the decomposition of $\hat{E}_{ij}$ (estimated via (10)), usually have large errors due to misidentified and/or small number of corresponding points. Hence, instead of using existing algorithms (*e.g.*, [2, 14, 25]) to estimate the orientations $\hat{R}_i$ and then computing the pairwise direction estimates $\hat{\gamma}_{ij} =$

[2] We note that the least squares version of (7) (*i.e.*, the program with the cost function $\sum_{(i,j)\in E_t} \|\mathbf{t}_i - \mathbf{t}_j - d_{ij}\gamma_{ij}\|^2$, and the same constraints as in (7)), which we name the "constrained least squares" (CLS) method, was previously studied in [34, 35]. However, as we experimentally demonstrate in §4, the CLS formulation fails to maintain robustness to outliers. Also, the ℓ_∞ version of (7), using the same constraints, was studied in [26].

[3] Although we use a single smooth approximation by fixing $\delta \ll 1$ for simplicity, we always obtained a convergent behavior in our experiments.

$\hat{R}_i\hat{\mathbf{t}}_{ij}/\|\hat{\mathbf{t}}_{ij}\|$, we take the following approach (a similar approach is used in [27]): First, the rotation estimates $\hat{R}_i$ are computed using the iterative method in §4.1 of [27] (using the robust algorithm of [7] for each iteration), and we then use the epipolar constraints (10) to robustly estimate the pairwise directions.

To that end, we rewrite the epipolar constraint (10) to emphasize its linearity in $\mathbf{t}_i$ and $\mathbf{t}_j$. Let $\{\mathbf{q}_i^k\}_{k=1}^{m_{ij}}$ and $\{\mathbf{q}_j^k\}_{k=1}^{m_{ij}}$ denote m_{ij} corresponding feature points. Then, for $\eta_i^k = \begin{bmatrix} \mathbf{q}_i^k/f_i \\ 1 \end{bmatrix}$ and $\eta_j^k = \begin{bmatrix} \mathbf{q}_j^k/f_j \\ 1 \end{bmatrix}$, we can rewrite (10) as (also see [2, 22, 27])

$$\begin{aligned} &(\eta_i^k)^T E_{ij}\eta_j^k = \left(R_i\eta_i^k \times R_j\eta_j^k\right)^T(\mathbf{t}_i-\mathbf{t}_j) = 0 \\ \Longleftrightarrow\ &(\nu_{ij}^k)^T(\mathbf{t}_i-\mathbf{t}_j) = 0, \text{ for } \nu_{ij}^k \text{ denoting} \\ &\nu_{ij}^k := \boldsymbol{\Theta}\left(R_i\eta_i^k \times R_j\eta_j^k\right) \end{aligned} \qquad (11)$$

where, the normalization function $\boldsymbol{\Theta}$ is defined by $\boldsymbol{\Theta}(\mathbf{x}) = \mathbf{x}/\|\mathbf{x}\|$, $\boldsymbol{\Theta}(\mathbf{0}) = \mathbf{0}$. Then, in the noiseless case (assuming $m_{ij} \geq 2$, and that we can find at least two ν_{ij}^k's not parallel to each other), $\{\nu_{ij}^k\}_{k=1}^{m_{ij}}$ determine a 2D subspace orthogonal to $\mathbf{t}_i - \mathbf{t}_j$, and hence the (undirected) "line" through $\mathbf{t}_i$ and $\mathbf{t}_j$ (*i.e.*, $\gamma_{ij}^0 = b_{ij}\gamma_{ij}$, where the sign $b_{ij} \in \{-1,+1\}$ is unknown, but can be determined by using the fact that the 3D scene points should lie in front of the cameras).

In the presence of noisy measurements, *i.e.* if we replace R_i's, f_i's and $\mathbf{q_i}$'s with their estimates in (11), we essentially obtain noisy samples $\hat{\nu}_{ij}^k$'s from the 2D subspace orthogonal to $\mathbf{t}_i - \mathbf{t}_j$. In order to maintain robustness to outliers among $\hat{\nu}_{ij}^k$'s in the estimation of (undirected) lines γ_{ij}^0, we first consider the following (non-convex) problem:

$$\begin{aligned} &\underset{\gamma_{ij}^0}{\text{minimize}} \quad \sum_{k=1}^{m_{ij}} |(\gamma_{ij}^0)^T\hat{\nu}_{ij}^k| \\ &\text{subject to} \quad \|\gamma_{ij}^0\| = 1 \end{aligned} \qquad (12)$$

In order to obtain the estimate $\hat{\gamma}_{ij}^0$, we use a (heuristic) IRLS method for (12). Here, although the program (12) is not convex, and hence the IRLS method is not guaranteed to converge to global optima, we empirically observed this approach to produce high quality estimates for the lines γ_{ij}^0, while preserving computational efficiency (for alternative methods, see [23, 37, 42]). Lastly, the estimates $\hat{b}_{ij}$ of the signs of the direction estimates $\hat{\gamma}_{ij} = \hat{b}_{ij}\hat{\gamma}_{ij}^0$ are computed using the fact that the 3D points should lie in front of the cameras.

In Figure 3, we provide a comparison of our robust direction estimation method, with a PCA-based estimator (comprised of solving (12) by replacing the cost function with the sum of squares version, *i.e.* with $\sum_k |(\gamma_{ij}^0)^T\hat{\nu}_{ij}^k|^2$). The results imply that, the accuracy of the direction estimates can be significantly improved by our robust method. We also note that the running time of our robust method is comparable to that of PCA and hence does not significantly increase the overall running time of the entire pipeline.

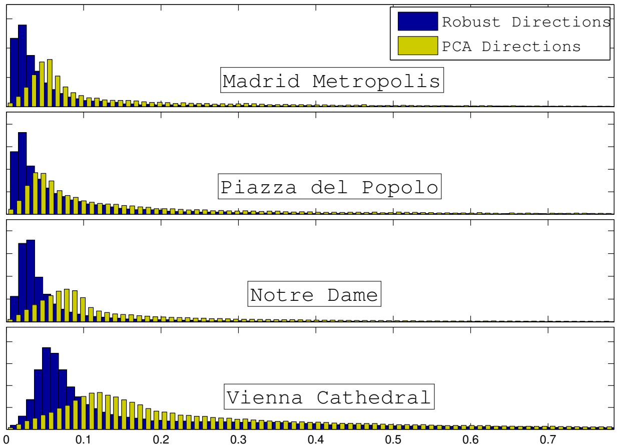


Figure 3. Histogram plots of the errors in direction estimates computed by our robust method (12) and the PCA method, for some of the datasets (from [40]) studied in §4.2. The errors represent the angles between the estimated directions and the corresponding ground truth directions (computed from a sequential SfM method based on [31], and provided in [40]). We note that the errors take values in $[0, \pi]$, yet the histograms are restricted to $[0, \pi/4]$ to emphasize the difference of the quality in the estimated directions.

A summary of our camera motion estimation algorithm is given in Table 1.

	Input: Images: $\{\mathrm{I}_i\}_{i=1}^n$, Focal lengths: $\{f_i\}_{i=1}^n$
Feature Points, Essential Matrices, Camera Orientations	**1.** Find corresponding points between images (using SIFT [24]) **2.** Compute $\hat{E}_{ij}$, using the eight-point algorithm [15] (for pairs with sufficiently many correspondences) **3.** Factorize $\hat{E}_{ij}$ to compute $\{\hat{R}_{ij}\}_{(i,j)\in E_R}$ and $G_R = (V_R, E_R)$ **4.** Compute the orientation estimates $\hat{R}_i$ from $\{\hat{R}_{ij}\}_{(i,j)\in E_R}$ (using the iterative approach of [27] with the robust method of [7] for each iteration)
Robust Pairwise Direction Estimation §3	**5.** Compute the 2D subspace samples $\{\hat{\nu}_{ij}^k\}_{k=1}^{m_{ij}}$ for each $(i,j) \in E_R$ (11) **6.** Estimate the pairwise directions $\{\hat{\gamma}_{ij}\}_{(i,j)\in E_R}$ using (12)
Location Estimation §2	**7.** Extract the largest maximally parallel rigid component $G_t = (V_t, E_t)$ of G_R (see [21]) **8.** Compute the location estimates $\{\hat{\mathbf{t}}_i\}_{i\in V_t}$ by the LUD (7) method (using the IRLS Algorithm 1 or classical interior point methods, *e.g.*, [36])
	Output: Camera orientations and translations: $\{\hat{R}_i, \hat{\mathbf{t}}_i\}$

Table 1. Algorithm for camera motion estimation

4. Experiments

4.1. Synthetic Data Experiments

In this section we provide synthetic data experiments for the LUD formulation (7). In particular, we provide evidence for exact location recovery from partially corrupted directions, and also compare the LUD solver to the CLS [34, 35], the SDR [27] and the LS [2, 5] methods.

The measurement graphs $G_t = (V_t, E_t)$ of our experiments are random graphs drawn from the Erdős-Rényi model $\mathcal{G}(n, q)$, *i.e.* each (i, j) is in the edge set E_t with probability q, independently of all other edges. In each experiment, we only record the results of problem instances defined on parallel rigid G_t. Given a set of locations $\{\mathbf{t}_i\}_{i=1}^n \subseteq \mathbb{R}^d$ and $G_t = (V_t, E_t)$, for each $(i, j) \in E_t$, we first let

$$\tilde{\gamma}_{ij} = \begin{cases} \gamma_{ij}^U \ , & \text{w.p. } \ p \\ (\mathbf{t}_i - \mathbf{t}_j)/\|\mathbf{t}_i - \mathbf{t}_j\| + \sigma\gamma_{ij}^G & \text{w.p. } 1 - p \end{cases} \tag{13}$$

and normalize $\tilde{\gamma}_{ij}$'s to obtain $\gamma_{ij} = \tilde{\gamma}_{ij}/\|\tilde{\gamma}_{ij}\|$ as the direction measurement for the pair (i, j). Here, $\{\gamma_{ij}^U\}_{(i,j)\in E_t}$ and $\{\gamma_{ij}^G\}_{(i,j)\in E_t}$ are i.i.d. random variables drawn from the uniform distribution on S^{d-1} and the standard normal distribution on $\mathbb{R}^d$, respectively. Also, the original locations $\mathbf{t}_i$'s are i.i.d. random variables drawn from standard normal distribution on $\mathbb{R}^d$.

We evaluate the performance in terms of the "normalized root mean squared error" (NRMSE) given by

$$\text{NRMSE}(\{\hat{\mathbf{t}}_i\}) = \sqrt{\frac{\sum_i \|\hat{\mathbf{t}}_i - \mathbf{t}_i\|^2}{\sum_i \|\mathbf{t}_i - \mathbf{t}_0\|^2}} \tag{14}$$

where $\hat{\mathbf{t}}_i$'s are the location estimates (after removal of the global scale and translation) and $\mathbf{t}_0$ is the center of $\mathbf{t}_i$'s.

The first set of experiments demonstrates the recovery performance of the LUD solver in the presence of partially corrupted directions, by setting $\sigma = 0$ in (13), and by controlling the proportion of outlier measurements via the parameter p. The results are summarized in Figure 4, where for each experiment the intensity of each pixel represents $\log_{10}$(NRMSE) (NRMSE values are averaged over 10 random realizations). These results demonstrate a striking feature of the LUD solver: In the presence of partially corrupted directions (with sufficiently small, but non-zero, proportion of corrupted directions), the LUD solver *recovers the original locations exactly* (*i.e.*, we get NRMSE $< \epsilon_{\text{IRLS}}$, where ϵ_{IRLS} is the convergence tolerance for the IRLS algorithm, set to $\epsilon_{\text{IRLS}} = 1\text{e-}8$ in our experiments). In Figure 4, we observe that, the exact recovery performance for $d = 3$ is improved as compared to the $d = 2$ case. Additionally, the transition to the exact recovery region becomes slightly sharper, and exact recovery performance for small values of outlier probability p is marginally improved when enlarging n from 100 to 200.

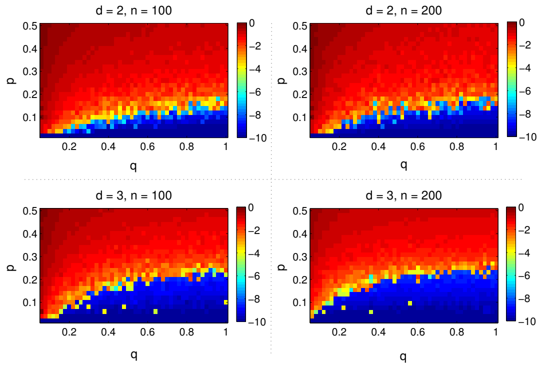


Figure 4. NRMSE (14) results of the LUD (7) solver for the exact recovery experiments. The color intensity of each pixel represents $\log_{10}$(NRMSE), depending on the edge probability q (x-axis), and the outlier probability p (y-axis). Measurements are generated by the noise model (13), assuming $\sigma = 0$, and NRMSE values are averaged over 10 trials.

The second set of experiments, depicted in Figure 5, presents a comparative evaluation of the NRMSE of the LUD, the CLS, the SDR and the LS solvers, for $d = 3$ (we observed similar performance for $d = 2$). The outcomes clearly present the robustness of the LUD formulation in the presence of outliers (up to a significant proportion of outliers, depending on q and n), while the recovery performance of the other methods is degraded significantly. Even if the measurement noise is dominated by small errors in the inlier directions (*i.e.*, when σ is relatively large compared to p), the LUD solver continues to outperform the other methods, in almost all cases.

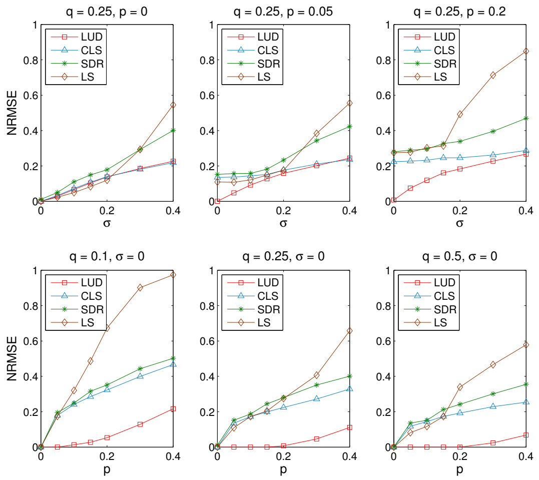


Figure 5. NRMSE (14) performance of the LUD (7) formulation vs. the CLS [34, 35], the SDR [27] and the LS [2, 5] solvers, for $n = 200$ locations. Measurements are generated by the noise model (13) and NRMSE values are averaged over 10 trials.

4.2. Real Data Experiments

We tested our location estimation algorithm on nine sets of real images from [40]. These are relatively irregular collections of images and hence estimating the camera loca-

Dataset			LUD							CLS [34, 35]					SDR [27]					1DSfM [40]				[12]	
Name	Size		Initial				After BA			Initial		After BA			Initial		After BA			Init.	After BA			After BA	
			PCA		Robust		Robust			Robust		Robust			Robust		Robust								
	m	N_c	$\tilde{e}$	$\hat{e}$	$\tilde{e}$	$\hat{e}$	N_c	$\tilde{e}$	$\hat{e}$	$\tilde{e}$	$\hat{e}$	N_c	$\tilde{e}$	$\hat{e}$	$\tilde{e}$	$\hat{e}$	N_c	$\tilde{e}$	$\hat{e}$	$\tilde{e}$	N_c	$\tilde{e}$	$\hat{e}$	N_c	$\tilde{e}$
Piazza del Popolo	60	328	3.0	7	**1.5**	**5**	305	**1.0**	**4**	3.5	6	305	1.4	5	1.9	8	305	1.3	7	3.1	308	2.2	200	93	16
NYC Library	130	332	4.9	9	**2.0**	**6**	320	1.4	7	5.0	8	320	3.9	8	5.0	8	320	4.6	8	2.5	295	**0.4**	**1**	271	1.4
Metropolis	200	341	4.3	8	**1.6**	**4**	288	1.5	**4**	6.4	10	288	3.1	7	4.2	8	288	3.1	7	9.9	291	**0.5**	70	240	18
Yorkminster	150	437	5.4	10	**2.7**	**5**	404	1.3	**4**	6.2	9	404	2.9	8	5.0	10	404	4.0	10	3.4	401	**0.1**	500	345	6.7
Tower of London	300	572	12	25	**4.7**	**20**	425	3.3	**10**	16	30	425	15	30	20	30	425	17	30	11	414	**1.0**	40	306	44
Montreal N. D.	30	450	1.4	2	**0.5**	**1**	435	**0.4**	**1**	1.1	2	435	0.5	**1**	–	–	–	–	–	2.5	427	**0.4**	**1**	357	9.8
Notre Dame	300	553	1.1	2	**0.3**	**0.8**	536	**0.2**	**0.7**	0.8	2	536	0.3	0.9	–	–	–	–	–	10	507	1.9	7	473	2.1
Alamo	70	577	1.5	3	**0.4**	**2**	547	**0.3**	**2**	1.3	3	547	0.6	**2**	–	–	–	–	–	1.1	529	**0.3**	2e7	422	2.4
Vienna Cathedral	120	836	7.2	12	**5.4**	**10**	750	4.4	**10**	8.8	**10**	750	8.2	**10**	–	–	–	–	–	6.6	770	**0.4**	2e4	652	12

Table 2. Performance comparison of various methods for datasets from [40]: Units are (approximately) in meters. N_c denotes number of estimated camera locations, $\hat{e}$ denotes the average distance, and $\tilde{e}$ denotes the median distance of the estimated camera locations to the corresponding cameras in the reference solution (computed using [31], and provided in [40]). 'PCA' and 'Robust' refers to the pairwise direction estimation method used (*c.f.* (12) and Figure 3).

tions for all of these images (or a large subset) is challenging. To solve the LUD problem (7), we use the IRLS algorithm 1, and to construct a 3D structure in our experiments, we use the parallel bundle adjustment (PBA) algorithm of [41]. We perform our computations on workstations with Intel(R) Xeon(R) X7542 CPUs, each with 6 cores, running at 2.67 GHz. In order to directly compare the accuracy of the location estimation by LUD to that of CLS [34, 35] and SDR [27] solvers, we use the same direction estimates (*c.f.* Table 1) for each method (except for the case where the PCA directions are used for the LUD solver, *c.f.* columns 4 and 5 of Table 2). These estimates produced more accurate location estimates for all data sets. We note that, the computation of the robust direction estimates is performed in parallel (using 10 cores for each dataset). Similar to [40], for performance evaluation, we consider the camera location estimates computed by a sequential SfM solver based on Bundler [31] (and provided in [40]) as the ground truth, and use a RANSAC-based method to compute the global transformation between our estimates and the ground truth.

We provide the accuracy comparisons in Table 2: The results are given in terms of the average distance $\hat{e}$, and the median distance $\tilde{e}$ of the estimated camera locations to the corresponding cameras in the reference solution (units are approximately in meters). The results of [40] correspond to the estimates computed by the combination of an outlier direction detection method (termed "1DSfM" in [40]) and a location estimation method employing a robust cost function. The results of [12] are cited from [40]. Also, the results of the SDR method [27] correspond to the estimates computed by applying the solver to the whole measurement graphs, and hence are not provided for the relatively larger datasets due to computational limitations. We also provide the running times corresponding to each experiment in Table 3 (note that the bundle adjustment times T_{BA} for the LUD, the CLS and the SDR solvers are computed after an initial 3D structure is provided). The comparison of the accuracy of the LUD solver given the robust directions (*c.f.* §3) to the case of the PCA directions, and the comparison of the LUD solver to the CLS, the SDR and [40] imply that, the combination of our robust direction estimation method and the LUD solver produces highly accurate initial estimates, with a computation cost that is slightly higher than the CLS method and [40]. Using the initial estimates, we apply PBA once, to obtain rich 3D structures and further improvements in accuracy. See Figure 6 for some of the 3D structures obtained from the initial LUD estimates.

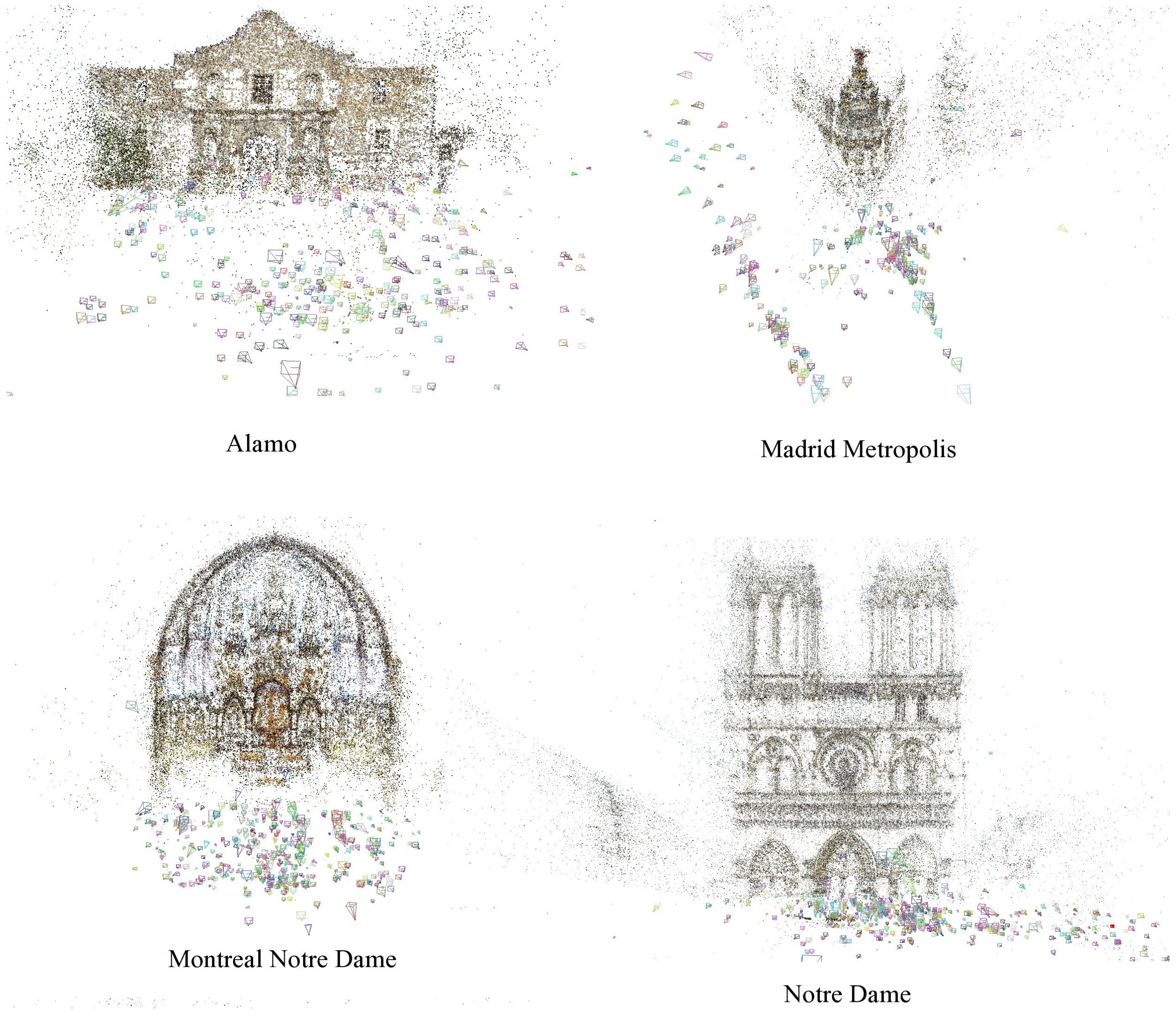
5. Conclusion and Future Work

We provided a complete characterization of well-posed instances of the camera location estimation problem, via the existing theory of parallel rigidity, and used it in practice to extract maximal image subsets for which estimation of camera location is well posed. For robust estimation of camera locations, we introduced a pairwise direction estimation method to maintain robustness to outliers in point correspondences, and we also presented a robust convex program, namely "the least unsquared deviations" (LUD) solver, to diminish the effects of outliers in pairwise direction measurements. We empirically demonstrated that unlike other estimators, the LUD formulation allows exact recovery of locations in the presence of partially corrupted direction measurements. In the context of structure from motion, our formulations can be used to efficiently and robustly estimate camera locations, in order to produce a high-quality initialization for reprojection error minimization algorithms, as demonstrated by our experiments on real image sets.

As future work, we plan to further investigate the phenomenon of exact recovery with partially corrupted directions, to characterize the conditions for its existence.

				LUD			CLS [34, 35]			SDR [27]			1DSfM [40]					[12]	[31]
Dataset	T_R	T_G	T_γ	$T_\mathbf{t}$	T_{BA}	T_{tot}	$T_\mathbf{t}$	T_{BA}	T_{tot}	$T_\mathbf{t}$	T_{BA}	T_{tot}	T_R	T_γ	$T_\mathbf{t}$	T_{BA}	T_{tot}	T_{tot}	T_{tot}
Piazza del Popolo	35	43	18	35	31	162	9	106	211	358	39	493	14	9	35	191	249	138	1287
NYC Library	27	44	18	57	54	200	7	47	143	462	52	603	9	13	54	392	468	220	3807
Metropolis	27	37	13	27	38	142	6	23	106	181	45	303	15	8	20	201	244	139	1315
Yorkminster	19	46	33	51	148	297	10	133	241	648	75	821	11	18	93	777	899	394	3225
Tower of London	24	54	23	41	86	228	8	202	311	352	170	623	9	14	55	606	648	264	1900
Montreal N. D.	68	115	91	112	167	553	21	270	565	–	–	–	17	22	75	1135	1249	424	2710
Notre Dame	135	214	325	247	126	1047	52	504	1230	–	–	–	53	42	59	1445	1599	1193	6154
Alamo	103	232	96	186	133	750	40	339	810	–	–	–	56	29	73	752	910	1403	1654
Vienna Cathedral	267	472	265	255	208	1467	46	182	1232	–	–	–	98	60	144	2837	3139	2273	10276

Table 3. Running times, in seconds, for the experiments in Table 2: times for orientation estimation (T_R), extraction of largest maximally parallel rigid component (T_G), robust pairwise direction estimation (T_γ), translation estimation ($T_\mathbf{t}$), bundle adjustment (T_{BA}), and total time (T_{tot}). (For the LUD, the CLS and the SDR solvers, the bundle adjustment times T_{BA} are computed after an initial 3D structure is provided, and the first three columns, *i.e.*, T_R,T_G,T_γ, are common).



Alamo

Madrid Metropolis

Montreal Notre Dame

Notre Dame

Figure 6. Snapshots of selected 3D structures computed using the camera location estimates of the LUD solver (7) (without bundle adjustment). Each 3D point is visible through at least three cameras.

Acknowledgements

The authors wish to thank Ronen Basri and Yuehaw Khoo for many valuable discussions related to this work.

The authors were partially supported by Award Number FA9550-12-1-0317 and FA9550-13-1-0076 from AFOSR, and by Award Number LTR DTD 06-05-2012 from the Simons Foundation.

References

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

[2] M. Arie-Nachimson, S. Kovalsky, I. Kemelmacher-Shlizerman, A. Singer, and R. Basri. Global motion estimation from point matches. In *3DimPVT*, ETH Zurich, Switzerland, October 2012. 2, 4, 5, 6

[3] J. Aspnes, T. Eren, D. K. Goldenberg, A. S. Morse, W. Whiteley, Y. R. Yang, B. D. O. Anderson, and P. N. Belhumeur. A theory of network localization. *IEEE Transactions on Mobile Computing*, 5(12):1663–1678, Dec. 2006. 3

[4] N. Bissantz, L. Dmbgen, A. Munk, and B. Stratmann. Convergence analysis of generalized iteratively reweighted least squares algorithms on convex function spaces. *SIAM Journal on Optimization*, 19(4):1828–1845, 2009. 4

[5] M. Brand, M. Antone, and S. Teller. Spectral solution of large-scale extrinsic camera calibration as a graph embedding problem. In *ECCV*, 2004. 2, 6

[6] W. Changchang. VisualSFM: A visual structure from motion system. http://homes.cs.washington.edu/~ccwu/vsfm/, 2011. 1

[7] A. Chatterjee and V. Govindu. Efficient and robust large-scale rotation averaging. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 521–528, Dec 2013. 2, 5

[8] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk. Iteratively reweighted least squares minimization for sparse recovery. *Communications on Pure and Applied Mathematics*, 63(1):1–38, 2010. 4

[9] T. Eren, W. Whiteley, and P. N. Belhumeur. Using angle of arrival (bearing) information in network localization. In *IEEE Conference on Decision & Control*, San Diego, CA, USA, December 2006. 2, 3

[10] T. Eren, W. Whiteley, A. S. Morse, P. N. Belhumeur, and B. D. O. Anderson. Sensor and network topologies of formations with direction, bearing, and angle information between agents. In *IEEE Conference on Decision & Control*, Maui, Hawaii, USA, December 2003. 2, 3

[11] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. Towards Internet-scale multi-view stereo. In *CVPR*, 2010. 1

[12] V. M. Govindu. Combining two-view constraints for motion estimation. In *CVPR*, 2001. 2, 7, 8

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

[14] R. Hartley, K. Aftab, and J. Trumpf. L_1 rotation averaging using the Weiszfeld algorithm. In *CVPR*, 2011. 2, 4

[15] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge Univ. Press, Cambridge, U.K., 2000. 5

[16] M. Havlena, A. Torii, J. Knopp, and T. Pajdla. Randomized structure from motion based on atomic 3D models from camera triplets. In *CVPR*, 2009. 1

[17] B. Jackson and T. Jordàn. Graph theoretic techniques in the analysis of uniquely localizable sensor networks. In B. F. G. Mao, editor, *Localization Algorithms and Strategies for Wireless Sensor Networks: Monitoring and Surveillance Techniques for Target Tracking*, pages 146–173. IGI Global, 2009. 3

[18] D. Jacobs and B. Hendrickson. An algorithm for two dimensional rigidity percolation: The pebble game. *Journal of Computational Physics*, 137(2):346–365, 1997. 3

[19] N. Jiang, Z. Cui, and P. Tan. A global linear method for camera pose registration. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 481–488, Dec 2013. 2

[20] B. Katz, M. Gaertler, and D. Wagner. Maximum rigid components as means for direction-based localization in sensor networks. In *Proceedings of SOFSEM '07*, pages 330–341, Berlin, Heidelberg, 2007. Springer-Verlag. 2, 3

[21] R. Kennedy, K. Daniilidis, O. Naroditsky, and C. J. Taylor. Identifying maximal rigid components in bearing-based localization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura, Algarve, Portugal, October 2012. 2, 3, 5

[22] L. Kneip, M. Chli, and R. Siegwart. Robust real-time visual odometry with a single camera and an IMU. In *Proc. BMVC*, pages 16.1–16.11, 2011. 5

[23] G. Lerman, M. McCoy, J. A. Tropp, and T. Zhang. Robust computation of linear models by convex relaxation. *Foundations of Computational Mathematics*, pages 1–48, 2014. 5

[24] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 5

[25] D. Martinec and T. Pajdla. Robust rotation and translation estimation in multiview reconstruction. In *CVPR*, 2007. 2, 4

[26] P. Moulon, P. Monasse, and R. Marlet. Global fusion of relative motions for robust, accurate and scalable structure from motion. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 3248–3255, Dec 2013. 2, 4

[27] O. Özyeşil, A. Singer, and R. Basri. Stable camera motion estimation using convex programming, 2015. *SIAM Journal on Imaging Sciences*, accepted. Also available at http://arxiv.org/abs/1312.5047. 2, 3, 5, 6, 7, 8

[28] B. Servatius and W. Whiteley. Constraining plane configurations in computer-aided design: Combinatorics of directions and lengths. *SIAM J. Discrete Math.*, 12(1):136–153, 1999. 2

[29] K. Sim and R. Hartley. Recovering camera motion using L_∞ minimization. In *CVPR*, 2006. 2

[30] S. N. Sinha, D. Steedly, and R. Szeliski. A multi-stage linear approach to structure from motion. In *RMLE*, 2010. 2

[31] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3D. In *SIGGRAPH*, 2006. 1, 5, 7, 8

[32] N. Snavely, S. M. Seitz, and R. Szeliski. Skeletal graphs for efficient structure from motion. In *CVPR*, 2008. 1

[33] B. Triggs, P. Mclauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment - a modern synthesis. In *Vision Algorithms: Theory and Practice, LNCS*, pages 298–375. Springer Verlag, 2000. 1

[34] R. Tron and R. Vidal. Distributed image-based 3-D localization of camera sensor networks. In *Proceedings of the 48th IEEE Conference on Decision and Control*, pages 901–908, 2009. 2, 4, 6, 7, 8

[35] R. Tron and R. Vidal. Distributed 3-D localization of camera sensor networks from 2-D image measurements. *Automatic Control, IEEE Transactions on*, 59(12):3325–3340, Dec 2014. 2, 4, 6, 7, 8

[36] R. H. Tütüncü, K. C. Toh, and M. J. Todd. Solving semidefinite-quadratic-linear programs using SDPT3. *Mathematical Programming Ser. B*, 95:189–217, 2003. 5

[37] D. E. Tyler. A distribution-free M-estimator of multivariate scatter. *The Annals of Statistics*, 15(1):234–251, Mar. 1987. 5

[38] W. Whiteley. A matroid on hypergraphs, with applications to scene analysis and geometry. *Discrete and Computational Geometry*, 4:75–95, 1989. 2, 3

[39] W. Whiteley. Matroids from discrete geometry. In J. G. Oxley, J. E. Bonin, and B. Servatius, editors, *Matroid Theory*, volume 197 of *Contemporary Mathematics*, pages 171–313. American Mathematical Society, 1996. 2, 3

[40] K. Wilson and N. Snavely. Robust global translations with 1DSfM. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part III*, pages 61–75, 2014. 2, 5, 6, 7, 8

[41] C. Wu, S. Agarwal, B. Curless, and S. M. Seitz. Multicore bundle adjustment. In *CVPR*, 2011. 7

[42] H. Xu, C. Caramanis, and S. Sanghavi. Robust PCA via outlier pursuit. *IEEE Transactions on Information Theory*, 58(5):3047–3067, 2012. 5

[43] T. Zhang and G. Lerman. A novel M-Estimator for robust PCA. *Journal of Machine Learning Research*, 15:749–808, 2014. 4

[44] Z. Zhang and Y. Shan. Incremental motion estimation through local bundle adjustment. Technical report, Microsoft Research, 2001. 1