Stable Camera Motion Estimation Using Convex Programming[*]

Onur Özyeşil[†], Amit Singer[‡], and Ronen Basri[§]

Abstract. We study the inverse problem of estimating n locations $\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_n$ (up to global scale, translation, and negation) in $\mathbb{R}^d$ from noisy measurements of a subset of the (unsigned) pairwise lines that connect them, that is, from noisy measurements of $\pm \frac{\mathbf{t}_i - \mathbf{t}_j}{\|\mathbf{t}_i - \mathbf{t}_j\|_2}$ for some pairs (i, j) (where the signs are unknown). This problem is at the core of the structure from motion (SfM) problem in computer vision, where the $\mathbf{t}_i$ represent camera locations in $\mathbb{R}^3$. The noiseless version of the problem, with exact line measurements, has been considered previously under the general title of parallel rigidity theory, mainly in order to characterize the conditions for unique realization of locations. For noisy pairwise line measurements, current methods tend to produce spurious solutions that are clustered around a few locations. This sensitivity of the location estimates is a well-known problem in SfM, especially for large, irregular collections of images. In this paper we introduce a semidefinite programming (SDP) formulation, specially tailored to overcome the clustering phenomenon. We further identify the implications of parallel rigidity theory for the location estimation problem to be well-posed, and prove exact (in the noiseless case) and stable location recovery results. We also formulate an alternating direction method to solve the resulting semidefinite program, and provide a distributed version of our formulation for large numbers of locations. Specifically for the camera location estimation problem, we formulate a pairwise line estimation method based on robust camera orientation and subspace estimation. Finally, we demonstrate the utility of our algorithm through experiments on real images.

Key words. structure from motion, parallel rigidity, semidefinite programming, convex relaxation, alternating direction method of multipliers

AMS subject classifications. 68T45, 52C25, 90C22, 90C25

DOI. 10.1137/140977576

1. Introduction. Global positioning of n objects from partial information about their relative locations is prevalent in many applications spanning fields such as sensor network localization [8, 57, 15, 17], structural biology [30], and computer vision [26, 9]. A well-known instance that attracted much attention from both the theoretical and algorithmic perspectives is that of estimating the locations $\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_n \in \mathbb{R}^d$ from their pairwise Euclidean distances

$\|\mathbf{t}_i - \mathbf{t}_j\|_2$. In this case, the large body of literature in rigidity theory (cf. [4, 53]) provides conditions under which the localization is unique given a set of noiseless distance measurements. Also, much progress has been made with algorithms that estimate positions from noisy distances, starting with classical multidimensional scaling [47] to the more recent semidefinite programming (SDP) approaches (see, e.g., [8, 7]).

Here we consider a different global positioning problem, in which the locations $\mathbf{t}_1, \ldots, \mathbf{t}_n$ need to be estimated from a subset of (potentially noisy) measurements of the pairwise lines that connect them (see Figure 1 for a noiseless instance). The line connecting $\mathbf{t}_i$ and $\mathbf{t}_j$ is identified with the rank-1 projection matrix Γ_{ij} defined by

$$(1.1) \qquad \Gamma_{ij} = (\mathbf{t}_i - \mathbf{t}_j)(\mathbf{t}_i - \mathbf{t}_j)^T / \|\mathbf{t}_i - \mathbf{t}_j\|_2^2 \,.$$

Notice that there is no available information about the Euclidean distances between the points. The entire information of pairwise lines is represented as a measurement graph $G_t = (V_t, E_t)$, where the ith 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 corresponding projection matrix Γ_{ij}.
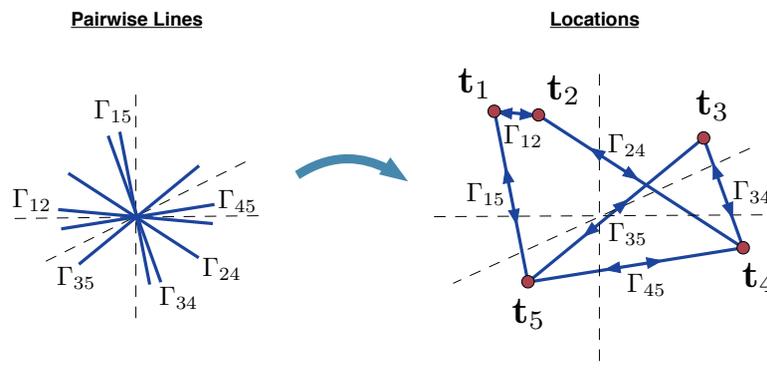

Figure 1. *A (noiseless) instance of the line estimation problem in $\mathbb{R}^3$, with $n = 5$ locations and $m = 6$ pairwise lines.*

The noiseless version of this problem (i.e., realization of locations from exact line measurements (1.1)) was previously studied in several different contexts, such as discrete geometry, sensor network localization, and robotics, under various formulations (see [64, 63, 16, 17, 48]). The concepts and the results for the noiseless case, to which we refer here as *parallel rigidity theory*, are aimed at characterizing the conditions for the existence of a unique realization of the locations $\mathbf{t}_i$ (of course, up to global translation, scaling, and negation of the locations $\mathbf{t}_i$, since the pairwise lines Γ_{ij} are invariant under these transformations).

However, location estimation from (potentially) noisy line measurements did not receive much attention previously. The camera location estimation part of the structure from motion (SfM) problem in computer vision (see, e.g., [26]), where $\mathbf{t}_i$ represent the camera locations in $\mathbb{R}^3$, is an important example of the abstract problem with noisy measurements. To the best of our knowledge, a structured formulation (in terms of the pairwise lines) of the camera location estimation problem and its relation to the existing results of parallel rigidity theory (characterizing conditions for *well-posed* instances of the problem) were not considered previ-

ously. We now give more details on the camera location estimation problem and the existing techniques for its solution.

Camera location estimation in SfM. SfM (depicted in Figure 2) is the problem of recovering a three-dimensional (3D) structure by estimating the camera motion corresponding to a collection of two-dimensional (2D) images (cf. section 4 for technical details).
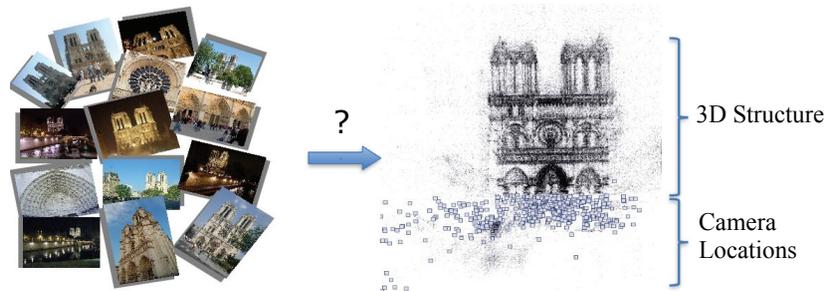

3D Structure

Camera Locations

Figure 2. *The SfM problem.*

Classically, SfM is solved in three stages: (1) feature point matching between image pairs (as in Figure 8) and relative pose estimation of camera pairs based on extracted feature points, (2) estimation of camera motion, i.e., global camera orientations and locations, from relative poses, and (3) 3D structure recovery based on estimated camera motion by reprojection error minimization (e.g., bundle adjustment of [55]). Although the first and third stages are relatively well understood and there exist accurate and efficient algorithms in the literature for these stages, existing methods for camera motion estimation, and specifically for the camera location estimation part, are sensitive to errors that result from mismatched feature correspondences. Among the widely used methods are incremental approaches (e.g., [1, 51, 52, 66, 70, 27, 18]), which integrate images to the estimation process one by one or in small groups. These incremental methods usually result in accumulation of estimation errors at each step and also require several applications of bundle adjustment for improved accuracy, hence leading to computational inefficiency. Alternatively, global estimation of the camera motion, i.e., solving for all camera orientations and/or locations simultaneously, can potentially yield a more accurate estimate. A generally adapted procedure to prevent high computational costs is to estimate the camera motion and the 3D structure separately. In principle, considering the large reduction in the number of variables, camera motion estimation can be performed jointly for all images, followed by 3D structure recovery using a single instance of reprojection error minimization. Obviously, such a procedure presumes a stable and efficient motion estimation algorithm. Most of the existing methods for motion estimation perform orientation and location estimation separately. Orientation estimation turns out to be a relatively well-posed problem, and there exist several efficient and stable methods, e.g., [41, 3, 25, 23, 56]. On the other hand, global estimation of camera locations, specifically for large, unordered sets of images, usually suffers from instability to errors in feature correspondences (resulting in solutions clustered around a few locations, as for [3, 9, 50]), sensitivity to outliers in pairwise line measurements (e.g., the ℓ_∞ approach of [49, 33]), and susceptibility

to local solutions in nonconvex formulations (e.g., [23]). Hence, a well-formulated, robust, efficient method for global location estimation (scalable to large sets of images) with provable convergence and stability guarantees is of high value.

Early works [22, 9] on location estimation reduce the problem to that of solving a set of linear equations that originate from the pairwise line measurements. Finding solutions to these linear systems can be done in a computationally efficient manner. However, these solutions have been empirically observed to be sensitive to errors in the pairwise line measurements. The sensitivity of such solutions is expressed by the tendency of the estimated locations to cluster, regardless of the true locations (cf. Figure 3 for such a clustering solution for a real data set, and also the optimization problem (2.2) and the following discussion). The multistage linear method of [50] attempts to resolve this issue by first performing pairwise reconstructions, then registering these in pairs, and finally aligning them by estimating relative scales and locations. Nonetheless, this approach does not produce satisfactory results in terms of estimation accuracy. Another interesting and efficient method is the Lie algebraic averaging approach of [23]. However, this nonconvex method is susceptible to convergence to local optima. The spectral formulation of [3], which is based on a novel decomposition of the essential matrix and is similar to [22, 9] in its formulation of the problem, yields an efficient linear solution for the locations, though it also suffers from spurious clustered solutions. Yet another formulation related to our work is the quasi-convex formulation of [49], which relies on (iteratively) optimizing a functional of the ℓ_∞ norm and requires the estimation of the signed directions of the pairwise lines, i.e., knowledge of $\frac{\mathbf{t}_i - \mathbf{t}_j}{\|\mathbf{t}_i - \mathbf{t}_j\|_2}$. However, the ℓ_∞ norm is highly susceptible to outliers, resulting in unsatisfactory solutions for real image sets. Also, the requirement for estimating signed line directions may introduce additional error to the problem. A similar idea is employed in [33] to simultaneously estimate the 3D structure, which exhibits the same difficulties. There are also various works that aim to improve the high sensitivity and inefficiency of these quasi-convex methods (see, e.g., [41, 44]). Another method requiring the estimation of the signed directions of the pairwise lines is studied in [56]. In contrast to the sensitivity of the quasi-convex method of [49] to outliers, the method in [56] is based on optimizing a functional of the ℓ_2 norm and hence produces more accurate location estimates. Additionally, [39] introduces a framework based on classical rigidity theory (involving pairwise distance information), which aims to identify rigid instances of the joint motion and structure estimation problem. Also, an SDP approach is introduced in [39] in order to jointly estimate motion and structure from noisy feature correspondences. We note that our work is significantly different from [39]: While we use parallel rigidity, [39] employs classical rigidity, leading to completely different SDP formulations.

Main contributions and broader context. In this paper we make the following principal contributions for the problem of location estimation from pairwise lines:

(i) The main contribution of our work is the introduction of a new semidefinite relaxation (SDR) formulation for estimating locations from pairwise line measurements. Empirically, we observe that solutions obtained by the SDR do not suffer from the clustering phenomenon of location estimates that plague many of the existing algorithms.

(ii) To quantify the performance of our SDR formulation, we prove exact (in the noiseless case; cf. Proposition 2.2) and stable (in the noisy case; cf. Theorem 2.3 and Corollary 2.4)
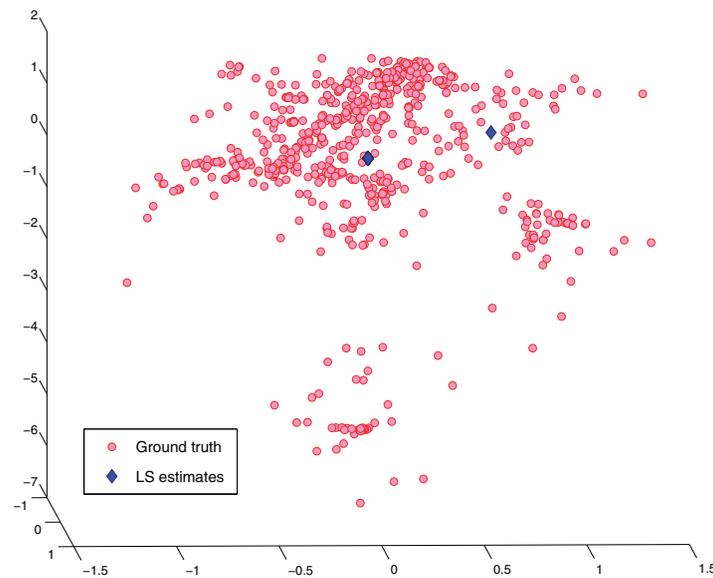
Figure 3. *A 2D snapshot of the collapsing solution of the least squares (LS) method* [3, 9] *for the Notre Dame data set from* [51]. *The solution of* [51] *is taken as the ground truth.*

location recovery results. We also provide a provably convergent and efficient alternating direction method to solve our SDR.

(iii) We provide a distributed version (scalable to a large number of locations) of the SDR formulation, based on spectral partitioning and convex programming. Additionally, we prove exact location recovery (in the noiseless case, cf. Proposition 3.1) for the distributed approach.

(iv) We formulate the camera location estimation problem of SfM in terms of the pairwise line measurements (between the camera locations in $\mathbb{R}^3$). Moreover, we show how to improve the stability of our approach to outlier pairwise measurements by robust preprocessing of the available pairwise camera information and describe a robust iterative approach for camera orientation estimation.

(v) We also demonstrate the efficiency and the accuracy of our formulations via synthetic and real data experiments. We note that these experiments show a specifically important characteristic of the SDR formulation: As long as the level of noise in the pairwise lines is below some threshold, the SDR always produces rank-1 solutions; i.e., it actually solves the original nonconvex program, which means the relaxation is tight.[1] Also, for higher noise levels, even when the SDR does not produce rank-1 solutions, its solution typically has a large spectral gap (i.e., it can be well approximated by a rank-1 matrix). In other words, we do not observe a sharp phase transition in the quality of the relaxation.

(vi) Since the existing results of parallel rigidity theory (cf. Appendix A and, e.g., [64, 63, 16, 17, 48]) have not been previously applied to the camera location estimation problem, we provide a summary of the main results of parallel rigidity theory, which completely characterize

---

[1]This is sometimes referred to as "exact rank recovery" and is not to be confused with our exact "location" recovery results for the SDR in the presence of noiseless pairwise lines.

the conditions for the problem to be well-posed (in $\mathbb{R}^d$, for arbitrary d). Also, we formulate a randomized algorithm to efficiently decide in the existence of these conditions.

In the literature, convex programming relaxations (and in particular semidefinite programs) have previously served as convex surrogates for nonconvex (particularly NP-hard) problems arising in several different areas, such as sensor network localization (from pairwise distances) [53, 8], low-rank matrix completion [12], phase retrieval [13], robust principal component analysis (PCA) [11], multiple-input multiple-output (MIMO) channel detection [42, 65], and many others (also see [20, 37, 60, 71]). Notably, the SDP formulation for sensor network localization [53, 8] is not guaranteed (even in the noiseless case) to provide the unique configuration of a globally rigid framework (cf. [4] for global rigidity and other fundamental concepts in "classical" rigidity theory). Only if the framework is "uniquely localizable" [53] is the SDP guaranteed to provide the unique solution in the noiseless case. In contrast, our SDR formulation is guaranteed to provide the unique solution (up to global scale, translation, and negation) for a parallel rigid framework (cf. section 2.1). Similar to our case, the tightness of the relaxation, i.e., obtaining rank-1 solutions from SDRs, is also observed in several different SDR formulations (see, e.g., [14, 5] and the survey [71]).

Organization of the paper. In section 2 we provide the connection to parallel rigidity theory and introduce the SDR formulation. Also, we prove exact (in the noiseless case) and stable (in the noisy case) location recovery results and formulate an alternating direction method for the SDR. In section 3 we introduce the distributed approach and prove its well-posedness. In section 4, we formulate the camera location estimation problem in SfM as a problem of estimating locations from their pairwise line measurements. We also present the robust camera orientation and pairwise line estimation procedures. We evaluate the experimental performance of our algorithm in section 5, using synthetic and real data sets. Finally, section 6 is a summary.

Reproducibility. The methods and algorithms presented in this paper are packaged in a MATLAB toolbox that is freely available for download from the first author's webpage at http://www.math.princeton.edu/~oozyesil/.

Notation. We denote vectors in $\mathbb{R}^d$, $d \geq 2$, in boldface. I_d and J_d are used for the $d \times d$ identity and all-ones matrices, respectively. S^d and 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; e.g., $\hat{X}$ is the estimate of X. We use an asterisk to denote solutions of optimization problems, as in X^*. For an $n \times n$ symmetric matrix A, $\lambda_1(A) \leq \lambda_2(A) \leq \cdots \leq \lambda_n(A)$ denote its eigenvalues (in ascending order), and $A \succeq 0$ denotes that A is positive semidefinite (i.e., $\lambda_i(A) \geq 0$ for all $i = 1, 2, \ldots, n$). Also, for an $n \times n$ matrix X, diag(X) denotes the vector formed by the diagonal entries of X, and conversely, for $\mathbf{x} \in \mathbb{R}^n$, Diag($\mathbf{x}$) denotes the diagonal matrix formed by the entries of $\mathbf{x}$. Finally, 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 line information.

2. Location estimation. Consider a graph $G_t = (V_t, E_t)$ of pairwise lines, with each edge $(i, j) \in E_t$ endowed with a pairwise line Γ_{ij} corresponding to the locations $\{\mathbf{t}_i\}_{i \in V_t}$ (i.e., satisfying (1.1)). Given this information, we first address the question of unique realizability of

the locations in the next subsection, followed by our SDP formulation for location estimation (from noisy pairwise lines).

2.1. Parallel rigidity. The unique realizability of locations (or *the solvability problem*) from pairwise line measurements was previously studied, mainly under the name of *parallel rigidity theory* (see, e.g., [64, 63, 16, 17, 48]). However, to the best of our knowledge, the concepts and the results of parallel rigidity theory have not been related to the well-posedness of the camera location estimation part of SfM, which is why we study them again here. Note that while camera orientation estimation from noiseless pairwise ratios (cf. section 4) only requires the connectivity of the measurement graph, connectivity alone is insufficient for camera location estimation (see Figure 4 for such an instance). To address this problem, we now briefly discuss the main results of parallel rigidity theory. For further details on fundamental concepts and results in parallel rigidity theory, see Appendix A.
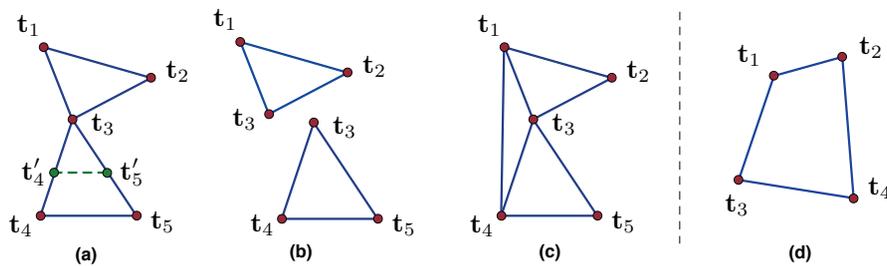

Figure 4. (a) *A formation of five locations on a connected graph, which is* not *parallel rigid (in* $\mathbb{R}^2$ *and* $\mathbb{R}^3$ *). Nonuniqueness is demonstrated by two noncongruent location solutions* $\{\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3, \mathbf{t}_4, \mathbf{t}_5\}$ *and* $\{\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3, \mathbf{t}'_4, \mathbf{t}'_5\}$, *each of which can be obtained from the other by an* independent rescaling *of the solution for one of its maximally parallel rigid components.* (b) *Maximally parallel rigid components 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* (1, 4) *linking its maximally parallel rigid components.* (d) *A formation of four locations, which is parallel rigid in* $\mathbb{R}^3$, *but is not parallel rigid in* $\mathbb{R}^2$.

We call the (noiseless) pairwise line information $\{\Gamma_{ij}\}_{(i,j)\in E_t}$ a "formation" and consider the following fundamental questions: Using this information, can we *uniquely* realize the points $\{\mathbf{t}_i\}_{i\in V_t}$, of course, up to a global translation, scale, and negation (i.e., can we realize a set of points *congruent* to $\{\mathbf{t}_i\}_{i\in V_t}$)? Is unique realizability a generic property (i.e., is it only a function of the underlying graph, independent of the particular realization of the points, assuming they are in generic position), and can it be decided efficiently? If we cannot uniquely realize $\{\mathbf{t}_i\}_{i\in V_t}$, can we efficiently determine maximal components of G_t that can be uniquely realized? These questions were previously studied in several different contexts, such as discrete geometry, bearing- and angle-based sensor network localization, and robotics, under various formulations and names (see [64, 63, 16, 17, 48, 34, 35, 36] and references therein).

The identification of parallel rigid formations is addressed in [64, 63, 16, 17] (also see [34] and the survey [31]), where it is shown that parallel rigidity in $\mathbb{R}^d$ ($d \geq 2$) is a generic property of the measurement graph G_t equivalent to unique realizability, and admits a complete combinatorial characterization (a generalization of Laman's condition from classical rigidity to parallel rigidity).

Theorem 2.1. *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 $D \subseteq (d-1)E$, with $|D| = d|V| - (d+1)$, such that for all subsets D' of D, we have

$$(2.1) \qquad\qquad |D'| \leq d|V(D')| - (d+1),$$

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

To elucidate the conditions of Theorem 2.1, let us consider the simple examples provided in Figure 4. For $\mathbb{R}^2$, if it exists, the certificate set D satisfying the conditions in Theorem 2.1 is simply a subset of the edge set E. The graphs in subfigures (b) and (c) are minimally parallel rigid in $\mathbb{R}^2$; i.e., $D = E$ is the certificate. On the other hand, the graphs in subfigures (a) and (d) do not have sufficiently many edges to be parallel rigid; i.e., even if we consider D to be the set of all edges E, we get $|D| < 2|V| - 3$. For $\mathbb{R}^3$, let us first consider the (triangle) graphs in subfigure (b): If we set $D \subseteq 2E$ to be the set of two copies of any of the two edges in E and a single copy of the remaining edge (e.g., $D = \{(1,2)_1, (1,2)_2, (2,3)_1, (2,3)_2, (1,3)_1\}$, where $(i,j)_k$ denotes the kth copy of the edge $(i,j) \in E$), then D is a certificate. For the graph in subfigure (c), $D = \{(1,2)_1, (1,2)_2, (1,3)_1, (1,3)_2, (1,4)_1, (2,3)_1, (3,4)_1, (3,4)_2, (3,5)_1, (4,5)_1, (4,5)_2\}$ satisfies the conditions of Theorem 2.1. Also, for the graph in subfigure (d), $D = 2E$ is the certificate. On the other hand, if we consider the graph in subfigure (a), D can satisfy the first condition $|D| = 11$ if and only if $D = 2E \setminus \{(i,j)_k\}$ for some $i, j \in V$, $k \in 1, 2$. However, in this case, D has a subset D' consisting of two copies of each of the three edges in a triangle graph, which violates the condition (2.1).

The conditions of Theorem 2.1 can be used to design efficient algorithms (e.g., adaptations of the pebble game algorithm [32], with a time complexity of $\mathcal{O}(n^2)$) for testing parallel rigidity. In Appendix A we detail a randomized algorithm (having time complexity $\mathcal{O}(m)$) for testing parallel rigidity. Moreover, polynomial time algorithms for finding maximally parallel rigid components of nonparallel rigid formations are provided in [35, 34].

In the presence of noiseless pairwise line measurements, parallel rigidity (and hence, unique realizability) is equivalent to error-free location recovery. However, for real images, we are provided with *noisy* measurements. In this case, instead of uniqueness of the solution of a specific location estimation algorithm, we consider the following question: Is there enough information for the problem to be *well-posed;* i.e., if the noise is small enough, can we estimate the locations stably? For formations which are not parallel rigid, instability may result from independent scaling and translation of maximally rigid components. In this sense, we consider problem instances on parallel rigid graphs to be well-posed.

Considering the existing results of parallel rigidity theory, we take the following approach for location estimation: Given a (noisy) formation $\{\Gamma_{ij}\}_{(i,j)\in E_t}$ on $G_t = (V_t, E_t)$, we first check for parallel rigidity of G_t; if the formation is nonrigid, we extract its maximally rigid components (using the algorithm in [35]) and estimate the locations for the largest maximally rigid component. The details of our formulation for (stable) location estimation are provided in the next subsection.

2.2. Location estimation by SDR. In this section we introduce and describe our location estimation algorithm. Suppose that we are given a set of (noisy) pairwise line measurements $\{\Gamma_{ij}\}_{(i,j)\in E_t}$ on $G_t = (V_t, E_t)$. Also, for each $(i,j) \in E_t$, let $Q_{ij} := I_d - \Gamma_{ij}$ denote the matrix of

projection onto the $(d-1)$-dimensional subspace orthogonal to this line. First, let us consider the LS approach studied in [3, 9],[2] which is equivalent to solving the following (nonconvex) program:

$$
\text{(2.2a)} \qquad \underset{\{\mathbf{t}_i\}_{i \in V_t} \subseteq \mathbb{R}^d}{\text{minimize}} \quad \sum_{(i,j) \in E_t} (\mathbf{t}_i - \mathbf{t}_j)^T Q_{ij} (\mathbf{t}_i - \mathbf{t}_j)
$$

$$
\text{(2.2b)} \qquad \text{subject to} \quad \sum_i \mathbf{t}_i = \mathbf{0} \;,\;\; \sum_i \|\mathbf{t}_i\|_2^2 = 1 \,.
$$

Note that we can rewrite the cost function in (2.2) as $\sum_{(i,j) \in E_t} \|Q_{ij}(\mathbf{t}_i - \mathbf{t}_j)\|_2^2$ (since the projection matrices Q_{ij} satisfy $Q_{ij} = Q_{ij}^T Q_{ij}$), which is why we call (2.2) the *least squares* approach. The constraints in (2.2) are designed to exclude the trivial case $\mathbf{t}_i = \mathbf{t}_0$ for all $i \in V_t$, for some $\mathbf{t}_0 \in \mathbb{R}^d$. In fact, (2.2) is an eigenvalue problem and hence can be efficiently solved. However, for large and noisy data sets, this formulation turns out to be "ill-conditioned" in the following sense: The solution has the tendency to "collapse"; i.e., the problem is not sufficiently constrained to prevent the (less trivial, but still undesired) solutions of the form $\hat{\mathbf{t}}_i \simeq \mathbf{t}_0$ for all $i \in V_t \setminus \{i^*\}$ and $\hat{\mathbf{t}}_{i^*} \simeq -\sum_{i \in V_t \setminus \{i^*\}} \hat{\mathbf{t}}_i$, where i^* has a (relatively) small degree in G_t (for such collapsing solutions in $\mathbb{R}^3$, see Figure 3 for a real data set and Figure 9 for synthetic data). For large data sets having nodes with significantly varying degrees, collapsing solutions of (2.2) (sometimes occurring in more than one group of points) can be observed even for low levels of noise in the Q_{ij}. It is worthwhile noting that the problem of collapsing solutions is not caused by the quadratic nature of the cost function in (2.2): Formulations of (2.2) using sum of (unsquared) projection errors, i.e., $\sum_{(i,j) \in E_t} \|Q_{ij}(\mathbf{t}_i - \mathbf{t}_j)\|_2$, as the cost function (which are also significantly harder to solve), which we studied using (heuristic) iteratively reweighted LS solvers, exhibit even worse characteristics in terms of collapsing solutions.

We overcome the collapsing behavior in two steps, first by introducing nonconvex "repulsion constraints" in (2.2) and then by formulating an SDR version of the resulting nonconvex problem. The nonconvex problem is given by

$$
\text{(2.3a)} \qquad \underset{\{\mathbf{t}_i\}_{i \in V_t} \subseteq \mathbb{R}^d}{\text{minimize}} \quad \sum_{(i,j) \in E_t} \text{Tr}\left(Q_{ij} (\mathbf{t}_i - \mathbf{t}_j)(\mathbf{t}_i - \mathbf{t}_j)^T \right)
$$

$$
\text{(2.3b)} \qquad \text{subject to} \quad \|\mathbf{t}_i - \mathbf{t}_j\|_2^2 \geq c \quad \forall (i,j) \in E_t \,,
$$

$$
\text{(2.3c)} \qquad \qquad\qquad\quad \sum_i \mathbf{t}_i = \mathbf{0} \,,
$$

where $c \in \mathbb{R}^+$ is a constant fixing the (undetermined) scale of the solution (without loss of generality (w.l.o.g.) we take $c = 1$) and the cost function is rewritten in a slightly different way. The repulsion constraints $\|\mathbf{t}_i - \mathbf{t}_j\|_2^2 \geq 1$ are nonconvex constraints making the estimation problem difficult even for small sizes. We introduce a matrix T of size $dn \times dn$, whose $d \times d$

---

[2] In [3], the LS approach is studied specifically for the location estimation problem in $\mathbb{R}^3$ with a slightly different definition for the pairwise measurements Γ_{ij} (also see section 4), whereas in [9], it is studied in arbitrary dimension using *unnormalized* Γ_{ij} (i.e., Γ_{ij} do not necessarily satisfy $\text{Tr}(\Gamma_{ij}) = 1$).

blocks are given by $T_{ij} = \mathbf{t}_i \mathbf{t}_j^T$. Consequently, $T \succeq 0$ and $\text{rank}(T) = 1$. To rewrite the cost function in (2.3) linearly in terms of T, we define a Laplacian matrix $L \in \mathbb{R}^{dn \times dn}$, whose $d \times d$ blocks are given by

$$(2.4) \qquad L_{ij} = \begin{cases} -Q_{ij} & \text{for } (i,j) \in E_t \,, \\ \sum_{\{k:(i,k) \in E\}} Q_{ik} & \text{for } i = j \,, \\ 0_d & \text{else} \end{cases}$$

and which satisfies $\text{Tr}(LT) = \sum_{(i,j) \in E_t} \text{Tr}(Q_{ij} (\mathbf{t}_i - \mathbf{t}_j)(\mathbf{t}_i - \mathbf{t}_j)^T)$. Note that L is symmetric, since $Q_{ij} = Q_{ij}^T$ and $Q_{ij} = Q_{ji}$, and also positive semidefinite, since for $\mathbf{t} \in \mathbb{R}^{dn}$, $\mathbf{t}^T L \mathbf{t} = \sum_{(i,j) \in E_t} \|Q_{ij}(\mathbf{t}_i - \mathbf{t}_j)\|_2^2 \geq 0$. Also, for every edge $(i,j) \in E_t$, we define a matrix $C^{ij} \in \mathbb{R}^{dn \times dn}$, whose klth $d \times d$ block is given by

$$(2.5) \qquad C_{kl}^{ij} = \begin{cases} I_d & \text{for } (k,l) = (i,i) \text{ or } (k,l) = (j,j) \,, \\ -I_d & \text{for } (k,l) = (i,j) \text{ or } (k,l) = (j,i) \,, \\ 0_d & \text{else} \end{cases}$$

and which allows us to rewrite the inequality constraints $\|\mathbf{t}_i - \mathbf{t}_j\|_2^2 \geq 1$ in (2.3), linearly in T as $\text{Tr}(C^{ij}T) \geq 1$. Moreover, the equality constraint $\sum_i \mathbf{t}_i = \mathbf{0}$ can be rewritten linearly in T as $\text{Tr}(HT) = 0$ for $H = J_n \otimes I_d$.

We now relax the only nonconvex constraint, that is, $\text{rank}(T) = 1$, to formulate the following SDR (known as "matrix lifting"):

$$(2.6a) \qquad \underset{T \in \mathbb{R}^{dn \times dn}}{\text{minimize}} \quad \text{Tr}(LT)$$

$$(2.6b) \qquad \text{subject to} \quad \text{Tr}(C^{ij}T) \geq 1 \quad \forall (i,j) \in E_t \,,$$

$$(2.6c) \qquad \qquad \qquad \text{Tr}(HT) = 0 \,,$$

$$(2.6d) \qquad \qquad \qquad T \succeq 0 \,.$$

After solving for T^* in (2.6), we compute the location estimates $\{\hat{\mathbf{t}}_i\}_{i \in V_t}$ by a deterministic rounding procedure; i.e., we compute the leading eigenvector $\hat{\mathbf{t}}$ of T^* and let $\hat{\mathbf{t}}_i$ be given by the ith $d \times 1$ block of $\hat{\mathbf{t}}$.

2.3. Stability of SDR. In this section we analyze the SDR (2.6) in terms of exact location recovery in the presence of noiseless pairwise line information and stable recovery with noisy line information.

We first introduce some notation to simplify our statements. Consider a set of locations $\{\mathbf{t}_i^0\}_{i \in V_t} \subseteq \mathbb{R}^d$ in generic position, and let $\gamma_{ij}^0 = \frac{\mathbf{t}_i^0 - \mathbf{t}_j^0}{\|\mathbf{t}_i^0 - \mathbf{t}_j^0\|_2}$ denote the unit vector from $\mathbf{t}_j^0$ to $\mathbf{t}_i^0$. Then, the (noiseless) formation corresponding to $\{\mathbf{t}_i^0\}_{i \in V_t}$ is given by $\{\Gamma_{ij}^0\}_{(i,j) \in E_t}$, where $\Gamma_{ij}^0 = \gamma_{ij}^0 (\gamma_{ij}^0)^T$. We also let Q_{ij}^0 denote the projection matrices onto the $(d-1)$-dimensional subspace orthogonal to γ_{ij}^0, i.e., $Q_{ij}^0 = I_d - \Gamma_{ij}^0$, L^0 denote the Laplacian matrix corresponding to Q_{ij}^0 (cf. (2.4)), and $T_0 \in \mathbb{R}^{dn \times dn}$ denote the matrix of the noiseless locations, i.e., $T_0 = \mathbf{t}^0 (\mathbf{t}^0)^T$, where the ith $d \times 1$ block of $\mathbf{t}^0$ is equal to $\mathbf{t}_i^0$.

We start with the simpler exact recovery result.

Proposition 2.2 (exact recovery in the noiseless case). *Assume that the (noiseless) formation* $\{\Gamma_{ij}^0\}_{(i,j)\in E_t}$ *is parallel rigid. Then, the SDR* (2.6) *(with* $L = L^0$*), followed by the deterministic rounding procedure, recovers the locations exactly, in the sense that any rounded solution of the SDR is congruent to* $\{\mathbf{t}_i^0\}_{i\in V_t}$*.*

Proof. W.l.o.g., we assume $\min_{(i,j)\in E_t}\|\mathbf{t}_i^0 - \mathbf{t}_j^0\|_2 = 1$ and $\sum_i \mathbf{t}_i^0 = \mathbf{0}$. Then we have $\mathrm{Tr}(L^0 T_0) = 0$, i.e., T_0 is a minimizer of (2.6) (since T_0 is also feasible by construction). The parallel rigidity of the formation implies $\mathrm{rank}(L^0) = dn - (d+1)$, where the only eigenvector $\mathbf{t}^0$ of T_0 with nonzero eigenvalue and the d eigenvectors $\mathbf{v}_H^1, \mathbf{v}_H^2, \ldots, \mathbf{v}_H^d$ of H (in (2.6)) with nonzero eigenvalues form an orthogonal basis for the nullspace of L^0 (see Appendix A). Let $\mathbf{u}_0^i$, $i = d+2, \ldots, dn$, denote the (normalized) eigenvectors of L^0 corresponding to its positive eigenvalues. Consider an arbitrary minimizer T^* of (2.6). Since $\mathrm{Tr}(L^0 T^*) = \sum_{i=d+2}^{dn} \lambda_i(L^0)(\mathbf{u}_0^i)^T T^* \mathbf{u}_0^i = 0$, where $\lambda_i(L^0) > 0$, $T^* \succeq 0$ satisfies $(\mathbf{u}_0^i)^T T^* \mathbf{u}_0^i = 0$ for all $i = d+2, \ldots, dn$. Also, by the feasibility of T^*, we get $\mathrm{Tr}(HT^*) = \sum_{i=1}^d (\mathbf{v}_H^i)^T T^* \mathbf{v}_H^i = 0$, i.e., $(\mathbf{v}_H^i)^T T^* \mathbf{v}_H^i = 0$ for all $i = 1, \ldots, d$. Hence, $\{\mathbf{v}_H^1, \ldots, \mathbf{v}_H^d, \mathbf{u}_0^{d+2}, \ldots, \mathbf{u}_0^{dn}\}$ form an orthogonal basis for the nullspace of T^*. This implies $\mathrm{rank}(T^*) = 1$, where T^* is of the form $T^* = \alpha T_0$ for some $\alpha \geq 1$ (by the feasibility of T^*), establishing the uniqueness of the solution up to scale. As a result, applying the rounding procedure to any solution of (2.6) yields exact recovery of $\mathbf{t}_i^0$ (of course, up to congruence). ∎

Our next result is the stability of the SDR with noisy pairwise line information.

Noise model and error function. We let each edge (i,j) of the measurement graph $G_t = (V_t, E_t)$ be endowed with a line measurement $\Gamma_{ij} = \gamma_{ij}\gamma_{ij}^T$, where $\gamma_{ij} = \gamma_{ij}^0 + \epsilon_{ij}$ with $\|\epsilon_{ij}\|_2 \leq \epsilon$ and $\|\gamma_{ij}\|_2 = 1$. Also, $L_G = D_G - A_G$ denotes the Laplacian of the graph G_t, where D_G is the (diagonal) degree matrix of G_t (whose ith diagonal element is equal to the degree of the ith node) and A_G is the adjacency matrix of G_t.

This time we assume (w.l.o.g.) that $\mathrm{Tr}(T_0) = 1$ and $\mathrm{Tr}(HT_0) = 0$. For a solution T^* of the SDR (2.6), we consider the following error function as our measure of stability:

$$(2.7) \qquad \delta(T^*, T_0) = \min_{c \geq 0} \|cT^* - T_0\|_F = \left\|\frac{\mathrm{Tr}(T^*T_0)}{\|T^*\|_F^2}T^* - T_0\right\|_F.$$

We are now ready to present the main result of this section.

Theorem 2.3 (stability of SDR solution). *Consider a set of noisy pairwise line measurements* $\{\Gamma_{ij}\}_{(i,j)\in E_t}$ *related to the (noiseless) parallel rigid formation* $\{\Gamma_{ij}^0\}_{(i,j)\in E_t}$ *as in the noise model given above, and let* T^* *be a solution of* (2.6)*. Then,*

$$(2.8) \qquad \delta(T^*, T_0) \leq \epsilon\left[\alpha_1 + \left(\alpha_1^2 + 2\alpha_2\right)^{\frac{1}{2}}\right],$$

where the (data-dependent) constants α_1 *and* α_2 *are given by* $\alpha_1 = \frac{\sqrt{2}m}{\lambda_{d+2}(L^0)}$ *and* $\alpha_2 = (\frac{\kappa\sqrt{d}\|L_G\|_F}{m} + 1)\frac{\lambda_n(L_G)}{\lambda_{d+2}(L^0)}$*. Here, the parameter* κ *is given by* $\kappa = (\min_{(i,j)\in E_t}\|\mathbf{t}_i^0 - \mathbf{t}_j^0\|_2^2)^{-1}$*.*

Proof. See Appendix B. ∎

We can obtain the stability of the estimated locations, i.e., the rounded solution of (2.6), as a corollary of Theorem 2.3.

Corollary 2.4 (location stability). *Let T^* be as in Theorem 2.3, and let $\hat{\mathbf{t}}$ denote its normalized eigenvector corresponding to its largest eigenvalue. Then*

$$(2.9) \qquad \min_{a \in \mathbb{R}} \|a\hat{\mathbf{t}} - \mathbf{t}^0\|_2 \le \epsilon \left[\frac{\pi d(n-1)}{2} \left(\alpha_1 + \left(\alpha_1^2 + 2\alpha_2 \right)^{\frac{1}{2}} \right) \right].$$

Proof. We use a particular implication of the Davis–Kahan theorem (see, e.g., Chapter 7 in [6]) in order to relate the recovery error in the rounded solution to the error in the solution of the SDR. To this end, observe that

$$(2.10) \qquad \min_{a \in \mathbb{R}} \|a\hat{\mathbf{t}} - \mathbf{t}^0\|_2 = \|((\mathbf{t}^0)^T\hat{\mathbf{t}})\hat{\mathbf{t}} - \mathbf{t}^0\|_2 = \|(I_{dn} - \mathbf{t}^0(\mathbf{t}^0)^T)\hat{\mathbf{t}}\hat{\mathbf{t}}^T\|_F.$$

For a given symmetric matrix A and a subset S of the real line, let $P_A(S)$ denote the projection matrix onto the subspace spanned by the eigenvectors of A whose eigenvalues are in S. Then the Davis–Kahan theorem implies

$$(2.11) \qquad \|P_A(S_1)P_B(S_2)\|_F \le \frac{\pi}{2\rho(S_1, S_2)}\|A - B\|_F,$$

where $\rho(S_1, S_2) = \min\{|x - y| : x \in S_1, y \in S_2\}$. In our case, if we let $S_0 = \{0\}$ for T_0 and $S^* = \{\lambda_{dn}(\tilde{c}T^*)\}$ for $\tilde{c}T^*$, where $\tilde{c} = (\mathrm{Tr}(T^*))^{-1}$, we obtain

$$(2.12) \qquad \|P_{T_0}(S_0)P_{\tilde{c}T^*}(S^*)\|_F = \|(I_{dn} - \mathbf{t}^0(\mathbf{t}^0)^T)\hat{\mathbf{t}}\hat{\mathbf{t}}^T\|_F \le \frac{\pi\,\mathrm{Tr}(T^*)}{2\lambda_{dn}(T^*)}\|\tilde{c}T^* - T_0\|_F.$$

Here, we use the fact that $\frac{\mathrm{Tr}(T^*)}{\lambda_{dn}(T^*)} \le \mathrm{rank}(T^*)$ and the feasibility of T^*, i.e., that $\mathrm{Tr}(HT^*) = 0$, to get $\frac{\mathrm{Tr}(T^*)}{\lambda_{dn}(T^*)} \le d(n-1)$ (in fact, we can construct a solution T^* of the SDR satisfying the stronger bound $\mathrm{rank}(T^*) \le (\sqrt{8(m+1)+1} - 1)/2$ (see, e.g., [46]); however, we ignore this slight improvement for simplicity). Also, considering (2.10) and (B.14) from the proof of Theorem 2.3, we recover the claim of the corollary. ∎

Remark 1. We note that the bounds in Theorem 2.3 and Corollary 2.4 are quite loose when compared to our experimental observations. Nevertheless, the recovery error is within a constant factor of the noise level ϵ. Also observe that Proposition 2.2, i.e., exact recovery in the noiseless case, is implied by Theorem 2.3 when $\epsilon = 0$.

Remark 2. The proximity of the solution T^* of (2.6) to the space of positive semidefinite rank-1 matrices can be considered as a measure of the quality of the relaxation. In our experiments with real images and simulated data, we make the following observations: As long as the noise level ϵ is below some threshold, we always get $\mathrm{rank}(T^*) = 1$; i.e., we can actually solve the nonconvex problem efficiently by the SDR (2.6). For higher levels of noise, T^* is no longer a rank-1 matrix, but it typically has a large spectral gap $(\lambda_{dn}(T^*) - \lambda_{dn-1}(T^*))/\lambda_{dn}(T^*)$. In other words, we do not observe a sharp phase transition in the quality of the SDR, and the relaxation is stable under various noise models. Figure 5 provides an experimental evaluation in $\mathbb{R}^3$ of our observations about the stability of relaxation, using synthetic measurements generated under the noise model (5.1) of section 5.1 (and assuming $p = 0$ in (5.1), i.e., no outlier measurements, for simplicity) for graphs of $n = 50$ nodes and various edge densities $\theta = \frac{2m}{n(n-1)}$. We observe that even if the location recovery performance (represented by *normalized root mean squared error* (NRMSE) defined in (5.2)) degrades as the noise level increases, the tightness of the relaxation is preserved up to relatively high noise levels.
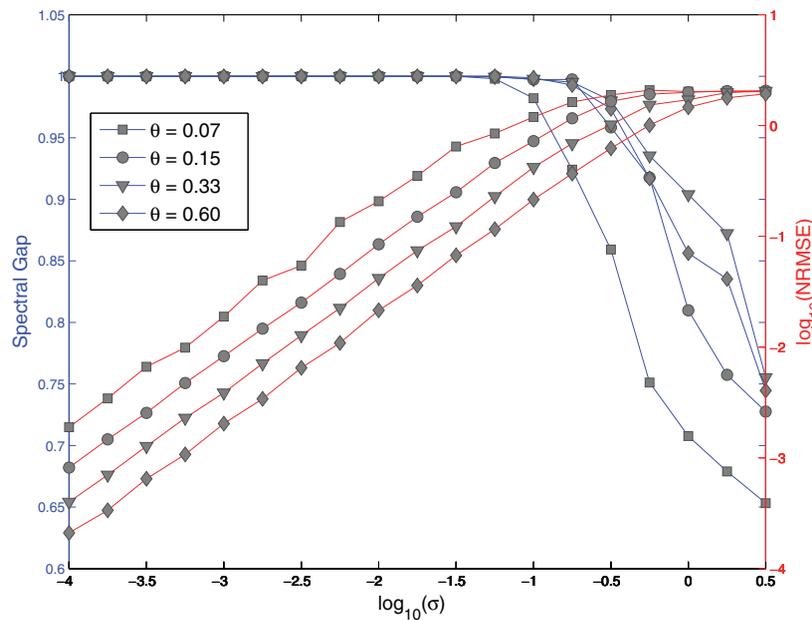
Figure 5. *Spectral gap $(\lambda_{3n}(T^*) - \lambda_{3n-1}(T^*))/\lambda_{3n}(T^*)$ of solutions T^* of (2.6) and logarithmic recovery error $\log_{10}(NRMSE)$ (see (5.2) for NRMSE) versus noise level $\log_{10}(\sigma)$ for graphs with $n = 50$ nodes and various edge densities θ (results are averaged over 10 random realizations of noisy line measurements and locations).*

2.4. Alternating direction augmented Lagrangian method. The SDR (2.6) is solvable in polynomial time by the classical primal-dual interior-point SDP algorithms (e.g., [58]). However, in case of a dense measurement graph G_t (i.e., assuming $m = \mathcal{O}(n^2)$), the interior-point methods become impractical for large numbers of locations, with a time complexity of $\mathcal{O}(n^6)$ (and a space complexity of $\mathcal{O}(n^4)$) for each iteration of the algorithm. In practice, the computational bottleneck becomes an issue for problem sizes of $n \geq 200$. In this respect, here we provide the details of *alternating direction augmented Lagrangian method* (ADM), which is a first-order method with superior computational efficiency [62]. ADM is an iterative algorithm based on minimization of an augmented Lagrangian function of the dual SDP. In comparison to interior-point methods that aim to satisfy complementary slackness while maintaining primal-dual feasibility at each iteration, ADM aims to construct a primal-dual feasible pair while maintaining complementary slackness. At each iteration, ADM minimizes the dual augmented Lagrangian function first with respect to dual variables, then with respect to dual slack variables, and finally updates the primal variables. In the minimization over each variable, ADM regards the other variables as fixed.

In order to obtain an ADM framework for the SDP (2.6), we rewrite it in standard form and procure the ADM framework (involving variables of larger dimensions) for standard form SDPs developed in [62]. Such an approach yields a (provably) convergent algorithm; however, in general it has a high computational cost (due to the high dimensionality of the variables associated with the standard form SDP). In our case, we are able to simplify the ADM framework for the standard form of (2.6) significantly and hence do not artificially increase the

computational cost by rewriting (2.6) in standard form (we also experimentally observed that the "ad hoc" approach in [62] developed for SDPs involving inequality constraints, which is at least as costly as our resulting algorithm, did not produce a convergent algorithm for (2.6)). We provide the details of rewriting (2.6) in standard form, constructing the ADM framework for the augmented Lagrangian of its dual and the simplification of the resulting algorithm in Appendix C. A pseudocode version of our ADM algorithm is given as Algorithm 1 (see Appendix C for the linear operators $\tilde{\mathcal{B}}, \tilde{\mathcal{B}}^*$ and the efficient computation of $(\tilde{\mathcal{B}}\tilde{\mathcal{B}}^* + I)^{-1}$).

Algorithm 1. Alternating direction augmented Lagrangian method (ADM) for SDP (2.6).

Initialize: $T^0 \succeq 0$ such that $\text{Tr}(HT^0) = 0$, $R^0 \succeq 0$, $\nu^0 \geq \mathbf{0}_m$, and $\eta^0 \geq \mathbf{0}_m, \mu > 0$

for $k = 0, 1, \ldots$ **do**

$$\mathbf{z}^{k+1} \leftarrow -\left(\tilde{\mathcal{B}}\tilde{\mathcal{B}}^* + I\right)^{-1}\left(\frac{1}{\mu}(\tilde{\mathcal{B}}(T^k) - \nu^k - \mathbf{1}_m) + \tilde{\mathcal{B}}(R^k - L) - \eta^k\right)$$

$$F^{k+1} \leftarrow L - \frac{1}{\mu}T^k - \tilde{\mathcal{B}}^*(\mathbf{z}^{k+1})$$

Compute the spectral decomposition of F^{k+1}:

$$F^{k+1} = [V_+ \ V_-]\begin{bmatrix} D_+ & 0 \\ 0 & D_- \end{bmatrix}\begin{bmatrix} V_+^T \\ V_-^T \end{bmatrix}, \text{ where } \text{diag}(D_+) > \mathbf{0}$$

$$R^{k+1} \leftarrow V_+ D_+ V_+^T$$

$$\eta^{k+1} \leftarrow \max\left\{\mathbf{z}^{k+1} - \frac{1}{\mu}\nu^k, \mathbf{0}_m\right\}$$

$$T^{k+1} \leftarrow -\mu V_- D_- V_-^T$$

$$\nu^{k+1} \leftarrow -\mu \min\left\{\mathbf{z}^{k+1} - \frac{1}{\mu}\nu^k, \mathbf{0}_m\right\}$$

end for

We refer the reader to [62] for practical details related to termination rules using measures of infeasibility, stagnation detection, updating the parameter μ for faster convergence, and using an additional step size parameter to update the primal variables T^k and ν^k, and also for convergence analysis of ADM. Considering the convergence rate analysis of ADM provided in [28], we need $\mathcal{O}(1/\epsilon)$ iterations in order to achieve an ϵ accuracy. Note that, at each iteration, the most computationally expensive step of Algorithm 1 is the spectral decomposition of F^{k+1}. However, since we experimentally observe a stable SDP relaxation resulting in a low-rank primal solution T^*, computation of V_- and D_- can be greatly simplified by computing only a few negative eigenvalues of F^{k+1} (e.g., by using Arnoldi iterations [2]). As a result, assuming $\mathcal{O}(n^3)$ complexity for spectral decomposition, the time complexity of $\mathcal{O}(n^3/\epsilon)$ (already significantly less compared to interior-point methods) can be reduced even further.

3. Distributed approach. The ADM framework introduced in section 2.4 provides a computationally feasible alternative to classical SDP solvers and allows us to solve the SDR (2.6) beyond $n \simeq 200$. However, for large sets of images ($n \gg 1000$), the need for a distributed algorithm is apparent. In this section, we provide the details of a distributed algorithm for translation estimation, based on spectral graph partitioning and convex programming.

The main structure of our distributed location estimation algorithm is the following: Given

a maximum problem size, i.e., an integer N_{max} denoting the maximum number of locations that our computational resources can efficiently estimate by (2.6), we first partition V_t into subsets (which we call "patches") of sizes at most N_{max} (by maintaining sufficient connectivity in the induced subgraphs and sufficient overlaps between patches). Then, for each induced subgraph, we extract the maximally parallel rigid components. We then find for each rigid component the "local" coordinate estimates by the SDR (2.6). Finally, we stitch the local estimates into a global solution by convex programming.

We note that the main idea of our distributed approach, i.e., division of the problem into smaller subproblems and then constructing the global solution from the local solutions, is also adapted for various problems in the literature (see, e.g., [15]). However, depending on the structure and the challenges of the specific problem studied, these distributed methods usually have significant differences. For instance, as compared to [15], while the same algorithm (namely, *the eigenvector method* (EVM)) is used in our approach to remove the pairwise sign ambiguities between local solutions (cf. section 3.2), the steps involving the graph partitioning and extraction of well-posed local problems, computation of local solutions, and estimation of global locations from (sign corrected) local estimates are significantly different.

3.1. Graph partitioning. In order to partition V_t into sufficiently overlapping subsets (for high quality global reconstruction) with sufficiently dense induced graphs (for high quality local estimation) of sizes bounded by N_{max}, we use the following algorithm, which bears a resemblance to the graph partitioning algorithm of [38]. Starting with $\mathcal{G}_t^1 = \{G_t\}$, at the kth iteration partition each graph in $\mathcal{G}_t^k$ (where $\mathcal{G}_t^k$ denotes the set of graphs to be partitioned) into two subgraphs using the spectral clustering algorithm of [43]. Then, extend the computed partitions to include the 1-hop neighborhoods of their vertices in G_t (and, of course, the induced edges). Assign the (extended) partitions with sizes smaller than N_{max} to the set of patches, and those with sizes larger than N_{max} to $\mathcal{G}_t^{k+1}$. Repeat until there is no subgraph left to partition, i.e., until the Kth iteration, where $\mathcal{G}_t^{K+1} = \emptyset$.

After the partitioning step, we extract the maximally parallel rigid components of each patch as described in section 2.1 (after this stage we use the term "patch" for parallel rigid patches). We then remove the patches that are subsets of another patch from the set of patches. We also remove the patches that do not have sufficient overlap (i.e., overlap size ≥ 2; also see the next subsection) with any other patch (which happens very rarely and is required since they cannot be used in the global location estimation). At this stage, we get a patch graph $G_P = (V_P, E_P)$, where V_P denotes the patches and $(i, j) \in E_P$ if and only if there is sufficient overlap between the patches P_i and P_j. Here, if G_P is not connected (which was never the case in our experiments), we can either extract the largest connected component of G_P or extend the patches to include their 1-hop neighborhoods until G_P is connected for the next steps of our algorithm. We then compute the "local" coordinate estimates for these rigid patches (whose negation signs, scales, and translations with respect to the global coordinate system are undetermined at this stage) by solving the SDR (2.6). The computation of the local coordinates for each patch can be done in parallel in a multicore processing environment, where each processing unit computes the local coordinates of one or more patches.

3.2. Pairwise patch registration. After solving the SDR (2.6) for each patch P_i, we obtain estimates $\{\hat{\mathbf{t}}_k^i\}_{k \in P_i}$ of the representations $\{\mathbf{t}_k^i\}_{k \in P_i}$ of the locations in the coordinate system

of each patch. The representations $\{\mathbf{t}_k^i\}_{k \in P_i}$ satisfy

$$(3.1) \qquad\qquad \mathbf{t}_k = c^i \mathbf{t}_k^i + \mathbf{t}^i, \quad k \in P_i, \ i \in V_P,$$

where $\mathbf{t}_k$ denotes the global coordinates of the kth location, and c^i and $\mathbf{t}^i$ denote the *signed* scale and translation of patch P_i, respectively (we use the signed scale, i.e., $c^i \in \mathbb{R}$ can assume negative values, because of the unknown negation). Given $\{\hat{\mathbf{t}}_k^i\}_{k \in P_i}$, $i \in V_P$, our objective is to estimate the locations $\{\mathbf{t}_k\}_{k \in \bigcup P_i}$ by formulating an efficient algorithm, which will be robust to possible outliers in the estimates $\hat{\mathbf{t}}_k^i$. In this respect, first observe that any algorithm designed to minimize the errors in the linear equations (3.1) should also exclude trivial solutions of the form $c^i = 0$ and $\mathbf{t}_k = \mathbf{t}^i = \mathbf{t}$ (for some $\mathbf{t} \in \mathbb{R}^d$) for all i and k. However, similar to the location estimation from noisy pairwise lines problem, the existence of this nullspace (spanned by the trivial solutions) results in collapsing solutions for underconstrained optimization programs. As in the case of the LS solver for the location estimation problem, we experimentally observed such collapsing solutions for the spectral method designed to minimize the sum of squared ℓ_2 norms of the errors in equations (3.1) by excluding the solutions in the nullspace.

Collapsing solutions can be avoided simply by requiring $|c^i|^2 \geq 1$ for all $i \in V_P$, which is a nonconvex constraint. Similar to the construction of the SDR (2.6), the nonconvexity (resulting from the unknown patch signs allowing c^i to assume negative values) can be resolved by using matrix lifting. An efficient method in this direction is the adaptation of the partial matrix lifting method of [14] to our problem (applied only to the variables involved in nonconvex constraints). In this method, using the sum of squared ℓ_2 norms of the errors in equations (3.1) as the cost function, the unconstrained variables ($\mathbf{t}_k$ and $\mathbf{t}^i$) are analytically computed as functions of the constrained variables (in our case, c^i), and the resulting quadratic form (in c^i) is used to define a matrix lifting relaxation for the constrained variables (see [14] for further details). However, this time, instead of using a matrix lifting method, we pursue a different approach: To overcome the nonconvexity in $|c^i|^2 \geq 1$, we first estimate the unknown sign of each patch P_i and then impose the convex constraints $c^i \geq 1$ for the sign-corrected patches (i.e., patches with the local estimates $\hat{\mathbf{t}}_k^i$ replaced with $\hat{z}^i \hat{\mathbf{t}}_k^i$, where $\hat{z}^i \in \{-1, +1\}$ is the estimate of the negation $z^i = \text{sign}(c^i)$ of patch P_i). Estimation of patch signs from pairwise signs (see (3.2) for pairwise sign estimation) is performed using the EVM (see, e.g., [15]), which is a robust and efficient spectral algorithm allowing a reliable estimation of patch signs. Using the estimated signs, we can minimize the sum of *unsquared* ℓ_2 norms in equations (3.1) (which cannot be used as a convex cost function in the matrix lifting approach) and hence maintain robustness to outliers in the estimates $\hat{\mathbf{t}}_k^i$. In our experiments with simulated data and real images, this two-step formulation produced more accurate location estimates compared to the matrix lifting alternative (with similar running times, since the partial matrix lifting results in a semidefinite program with a matrix variable of size $|V_P| \times |V_P|$), making it our choice for stitching the global solution. We now provide the details of the sign estimation procedure, whereas the final step of location estimation from sign-corrected patches is covered in section 3.3.

In order to estimate the patch signs $\{z^i\}_{i \in V_P}$, the relative pairwise signs $z^{ij} = z^i z^j$, $(i,j) \in E_P$, are estimated first. This is accomplished by solving the following LS problem for

each $(i, j) \in E_P$:

$$(3.2) \qquad \underset{c^{ij} \in \mathbb{R}, \mathbf{t}^{ij} \in \mathbb{R}^d}{\text{minimize}} \quad \sum_{k \in P_i \cap P_j} \left\| \hat{\mathbf{t}}_k^i - \left(c^{ij} \hat{\mathbf{t}}_k^j + \mathbf{t}^{ij} \right) \right\|_2^2 ,$$

where $c^{ij}, \mathbf{t}^{ij}$ denote the relative (signed) scale and translation between P_i and P_j, respectively. The relative sign estimate $\hat{z}^{ij} \in \{-1, +1\}$ is given by $\hat{z}^{ij} = \text{sign}((c^{ij})^*)$.

Using the relative sign estimates $\{\hat{z}^{ij}\}_{(i,j) \in E_P}$, the sign estimates $\{\hat{z}^i\}_{i \in V_P}$ are computed by EVM, which is a spectral method for finding signs with the goal of satisfying as many equations of the form $\hat{z}^i \hat{z}^j = \hat{z}^{ij}$ for $(i, j) \in E_P$ as possible (see [15] for details). Here, we note that, although the sum of squared norms cost function in (3.2) can be replaced by the sum of (unsquared) norms cost to improve robustness to outliers in $\hat{\mathbf{t}}_k^i$, we prefer the more efficient LS version (in fact, (3.2) has a closed-form solution) since we did not experimentally observe a significant improvement in the accuracy of signs estimated by EVM.

3.3. Global stitching of local patches. Stitching the local patches into a globally consistent d-dimensional map comprises the last step of our distributed approach. As we discussed in section 3.2, we aim to efficiently estimate the global locations $\mathbf{t}_k$ using the linear equations (3.1), while maintaining robustness to outliers in $\hat{\mathbf{t}}_k^i$ and preventing collapsing solutions. In this respect, using the estimated patch signs (i.e., estimates of signs of c^i) in (3.1), we maintain robustness by minimizing sum of (unsquared) norms of errors in equations (3.1), while simply constraining $c^i \geq 1$ to prevent collapse. Hence, we solve the following convex program (using, e.g., [58]), in which we jointly estimate the scales $\{c^i\}_{i \in V_P}$ and translations $\{\mathbf{t}^i\}_{i \in V_P}$ of the sign-corrected patches (i.e., patches with the local estimates $\hat{\mathbf{t}}_k^i$ replaced with $\hat{z}^i \hat{\mathbf{t}}_k^i$) and the global locations $\{\mathbf{t}_k\}_{k \in \bigcup P_i}$:

$$(3.3\text{a}) \qquad \underset{\{\mathbf{t}_k, \, c^i, \, \mathbf{t}^i\}}{\text{minimize}} \quad \sum_{i \in V_P} \sum_{k \in P_i} \left\| \mathbf{t}_k - \left(c^i \hat{z}^i \hat{\mathbf{t}}_k^i + \mathbf{t}^i \right) \right\|_2$$

$$(3.3\text{b}) \qquad \text{subject to} \quad c^i \geq 1 \quad \forall i \in V_P .$$

3.4. Well-posedness of the distributed problem. Similar to the well-posedness of location estimation from pairwise lines, we consider the following question for the distributed problem: Do the local coordinate estimates $\{\hat{\mathbf{t}}_k^i\}_{k \in P_i, i \in V_P}$ provide enough information to yield a well-posed instance of the global location estimation problem? Once again, we consider an instance of the distributed problem to be well-posed if the global locations can be uniquely (of course, up to congruence) estimated from the noiseless local coordinates $\{\mathbf{t}_k^i\}_{k \in P_i, i \in V_P}$. We (partially) answer this question in Proposition 3.1, where it is shown that the local coordinate estimates provided via the specific construction of the patch graph G_P given in section 3.1 are sufficient for well-posedness. This result is established by proving exact recovery of global locations from noiseless local coordinates using the two-step global location construction algorithm.

Proposition 3.1 (exact recovery from noiseless local coordinates). *Consider a graph $G_P = (V_P, E_P)$ of patches $\{P_i\}_{i \in V_P}$ and a set of (noiseless) local coordinates $\{\mathbf{t}_k^i\}_{k \in P_i, i \in V_P}$ corresponding to the global locations $\{\mathbf{t}_k\}_{k \in \bigcup P_i}$ (i.e., $\mathbf{t}_k^i$ satisfy (3.1) for a set of signed scales c^i and translations $\mathbf{t}^i$ of the patches for all $k \in P_i$ and $i \in V_P$). Then, if G_P is connected and satisfies $(i, j) \in E_P$ if and only if $|P_i \cap P_j| \geq 2$, the two-step global location construction*

algorithm (i.e., estimation of patch signs by (3.2) *and EVM* [15] *followed by global location estimation by* (3.3)*) recovers the global locations exactly when provided with the noiseless local coordinates (i.e.,* $\hat{\mathbf{t}}_k^i = \mathbf{t}_k^i$ *for* (3.2) *and* (3.3)*), in the sense that any solution of the algorithm is congruent to* $\{\mathbf{t}_k\}_{k \in \bigcup P_i}$.

Proof. See Appendix D. ∎

Remark 3. We note that, in the presence of noiseless pairwise lines $\{\Gamma_{ij}\}_{(i,j) \in E_t}$ (on parallel rigid $G_t = (V_t, E_t)$), and assuming that we can obtain a connected patch graph G_P from G_t using the graph partitioning procedure of section 3.1, Propositions 2.2 and 3.1 imply exact recovery of $\{\mathbf{t}_k\}_{k \in \bigcup P_i}$ from the lines $\{\Gamma_{ij}\}_{(i,j) \in E_t}$.

Remark 4. The conditions imposed on G_P in Proposition 3.1 (i.e., connectivity and that, for all $(i,j) \in E_P$, $|P_i \cap P_j| \geq 2$) are usually not difficult to satisfy. Also, observe that these conditions are independent of the dimension d of the locations (which is not the case, e.g., for the combinatorial conditions in [69, 21]). However, it may be possible to assume even weaker conditions on G_P to obtain exact recovery results using, e.g., the (partial) matrix lifting method discussed in section 3.2: We conjecture that if the patches $\{P_i\}_{i \in V_P}$ satisfy a specific 2-lateration[3] condition, i.e., if there exists a reordering of the patch indices such that, for every $2 \leq i \leq |V_P|$, P_i and $P_1 \cup \cdots \cup P_{i-1}$ have at least two points in common (which is, obviously, a weaker condition compared to the conditions in Proposition 3.1), then the matrix lifting method should recover the locations exactly. On the other hand, since in our two-step method, the first step requires the estimation of pairwise signs (in order to estimate the patch signs), the condition that $|P_i \cap P_j| \geq 2$ for all $(i,j) \in E_P$ is, in fact, necessary for exact recovery.

4. Camera motion estimation. In this section, we provide the details of the application of our location estimation algorithm (developed for the general problem in d dimensions) to the camera location estimation part of the SfM problem in computer vision (defined in $\mathbb{R}^3$). In the SfM problem (see Figure 2), camera motion estimation is based on point correspondences between pairs of images. As a result, misidentification of corresponding points can induce estimation errors in pairwise line estimates, denoted $\hat{\Gamma}_{ij}$, and manifest itself through large errors in the estimated camera locations. In that respect, our primary goal in this section is to formulate a robust (to misidentified corresponding points) and efficient procedure for pairwise line estimation that would then be used as input to our SDR framework (see Figure 6 for a comparison of the accuracy of the line estimates computed using our robust procedure and a simpler estimator, which is not robust to misidentified corresponding points). We also devise a robust algorithm for the camera orientation estimation part, which directly affects the recovery performance of the location estimation part. We start with a brief description of the measurement process.

Let $\{\mathrm{I}_1, \mathrm{I}_2, \ldots, \mathrm{I}_n\}$ represent a collection of images of a stationary 3D scene. We use a pinhole camera model (see Figure 7) and denote the orientations, locations of the focal points, 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 ith image plane by $\mathbf{p}_i \in \mathbb{R}^3$ (as in Figure 7). To produce $\mathbf{p}_i$, we first represent $\mathbf{P}$ in the ith

---

[3]We note that the notion of lateration considered here should not be confused with the classical laterated graphs, although it resembles the classical concept in some aspects.
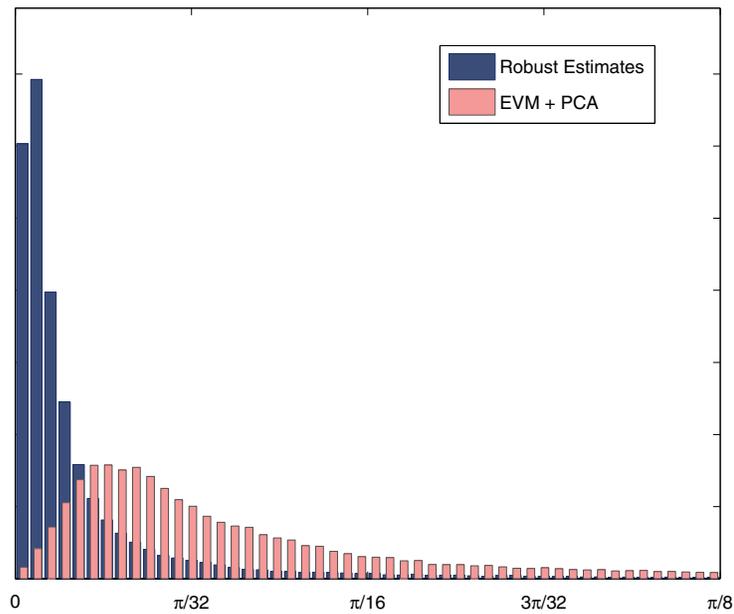
Figure 6. *Histogram plots of the errors in line estimates computed by our robust method (cf. sections 4.1 and 4.2) and a simpler estimator. The simpler estimator uses the EVM [3] for rotation estimation and PCA for subspace estimation (using the noisy estimates of the 2D subspace samples ν_{ij}^k in (4.3)). The errors represent the angles between the estimated lines and the corresponding ground truth lines (computed from the camera location estimates of [51] for the Notre Dame dataset, studied in section 5.2). The errors take values in $[0, \pi/2]$; however, the histograms are restricted to $[0, \pi/8]$ to emphasize the difference of the quality in the estimated lines. We note that, for the robust method, the percentage of the line estimates having errors larger than $\pi/8$ is 3.7%, whereas, for the simple estimator, it is 11.5%.*
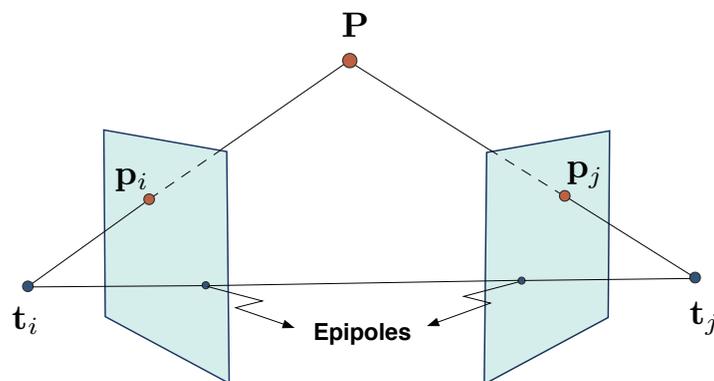
Figure 7. *3D projective geometry of the pinhole camera model (using virtual image planes for mathematical simplicity).*

camera's coordinate system, that is, we compute $\mathbf{P}_i = R_i^T(\mathbf{P} - \mathbf{t}_i) = (\mathbf{P}_i^x, \mathbf{P}_i^y, \mathbf{P}_i^z)^T$, and then project it to the ith 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 $\mathbf{P}$.

Following the conventions from the SfM literature, for an image pair I_i and I_j, the pairwise rotation and translation between the ith and jth camera coordinate frames are denoted by $R_{ij} = R_i^T R_j$ and $\mathbf{t}_{ij} = R_i^T(\mathbf{t}_j - \mathbf{t}_i)$ (not to be confused with $\mathbf{t}_i - \mathbf{t}_j$ or γ_{ij}, $\mathbf{t}^{ij}$ used previously), respectively. The essential matrix E_{ij} is then defined by $E_{ij} = [\mathbf{t}_{ij}]_\times R_{ij}$, where $[\mathbf{t}_{ij}]_\times$ is the skew-symmetric matrix corresponding to the cross product with $\mathbf{t}_{ij}$. If the projections of a 3D scene point $\mathbf{P}$ onto the ith and jth image planes are denoted by $\mathbf{p}_i \in \mathbb{R}^3$ and $\mathbf{p}_j \in \mathbb{R}^3$, respectively, the essential matrix E_{ij} satisfies the "epipolar constraint" given by

$$(4.1) \qquad \mathbf{p}_i^T E_{ij} \mathbf{p}_j = 0 \,.$$

In fact, the epipolar constraint (4.1) is a restatement of the coplanarity of the three vectors $\mathbf{P} - \mathbf{t}_i$, $\mathbf{P} - \mathbf{t}_j$, and $\mathbf{t}_i - \mathbf{t}_j$ (see Figure 7). However, since (4.1) is given in terms of the measurable variables $\mathbf{p}_i$, it is used as an extra constraint (on the special structure of E_{ij} having six degrees of freedom) in the estimation of E_{ij}.

Provided with the image set $\{I_i\}_{i=1}^n$, to estimate the essential matrices, we first extract feature points and find pairs of corresponding points between images (see Figure 8 for an image pair with corresponding points) using SIFT [40], and then estimate the essential matrices using the eight-point algorithm[4] (see, e.g., [26]), while also employing the RANSAC protocol (to reduce the effect of outliers in point correspondences). For image pairs with sufficiently many inliers, the estimated essential matrices $\hat{E}_{ij}$ are then (uniquely) factorized into $\hat{R}_{ij}$ and $[\hat{\mathbf{t}}_{ij}]_\times$.
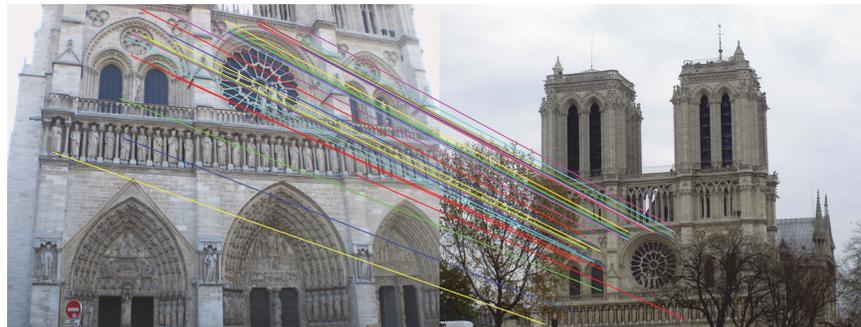

Figure 8. *Two images of the Notre Dame Cathedral set from* [51]*, with corresponding feature points (extracted using SIFT* [40]*). The essential matrix E_{ij} can be estimated using the (nonlinear) five-point or the (linear) eight-point algorithms (see, e.g.,* [26]*), which require at least five or eight pairs of corresponding feature points, respectively.*

Classically, the relative rotation and translation estimates, $\hat{R}_{ij}$ and $\hat{\mathbf{t}}_{ij}$, computed from the decomposition of $\hat{E}_{ij}$, are used to estimate the camera locations $\mathbf{t}_i$. However, for large and unordered collections of images, these estimates usually have errors resulting from misidentified and/or insufficiently many corresponding points. In particular, the erroneous estimates $\hat{\mathbf{t}}_{ij}$

---

[4] We note that the essential matrix E_{ij} can be estimated using only five point correspondences between the images I_i and I_j (while assuming $\|\mathbf{t}_i - \mathbf{t}_j\|_2 = 1$). In practice, however, we use the eight-point algorithm, which is an efficient linear method requiring a minimum of eight point correspondences.

result in large errors for the location estimation part. As a result, instead of using the existing algorithms (e.g., [25, 3, 41]) to find the rotation estimates $\hat{R}_i$ and then computing the pairwise line estimates $\hat{\Gamma}_{ij} = (\hat{R}_i \hat{\mathbf{t}}_{ij})(\hat{R}_i \hat{\mathbf{t}}_{ij})^T$ (assuming $\|\hat{\mathbf{t}}_{ij}\|_2 = 1$) for the SDR solver (2.6), we follow a different procedure: First, the rotation estimates $\hat{R}_i$ are computed using an iterative, robust algorithm (as detailed in section 4.1), and then we go back to the epipolar constraints (4.1) (as explained below) to estimate the pairwise lines using a robust subspace estimation algorithm (cf. section 4.2).

To clarify the main idea of our robust pairwise line estimation method using the epipolar constraints, we first emphasize the linearity of (4.1) in the camera locations $\mathbf{t}_i$ and $\mathbf{t}_j$ by rewriting it as (also see [3])

$$\mathbf{p}_i^T E_{ij} \mathbf{p}_j = \mathbf{p}_i^T \left[R_i^T (\mathbf{t}_j - \mathbf{t}_i) \right]_\times R_i^T R_j \mathbf{p}_j$$
$$= \mathbf{p}_i^T R_i^T \left\{ (\mathbf{t}_j - \mathbf{t}_i) \times R_j \mathbf{p}_j \right\}$$
$$(4.2) \qquad = (R_i \mathbf{p}_i \times R_j \mathbf{p}_j)^T (\mathbf{t}_i - \mathbf{t}_j) = 0 \,.$$

As mentioned before, for an image I_i, the measurement corresponding to a 3D point $\mathbf{P}$ is given in terms of the coordinates of the 2D image plane by $\mathbf{q}_i = (\mathbf{p}_i^x, \mathbf{p}_i^y)^T \in \mathbb{R}^2$. For an image pair I_i and I_j, let $\{\mathbf{q}_i^k\}_{k=1}^{m_{ij}} \subseteq \mathbb{R}^2$ and $\{\mathbf{q}_j^k\}_{k=1}^{m_{ij}} \subseteq \mathbb{R}^2$ denote m_{ij} corresponding feature points. Then, using (4.2), we get (in the noiseless case)

$$(4.3a) \qquad (\nu_{ij}^k)^T (\mathbf{t}_i - \mathbf{t}_j) = 0, \quad k = 1, \dots, m_{ij}, \quad \text{for}$$

$$(4.3b) \qquad \nu_{ij}^k := \boldsymbol{\Theta} \left[\left(R_i \begin{bmatrix} \mathbf{q}_i^k / f_i \\ 1 \end{bmatrix} \right) \times \left(R_j \begin{bmatrix} \mathbf{q}_j^k / f_j \\ 1 \end{bmatrix} \right) \right],$$

where ν_{ij}^k are normalized (using the homogeneity of (4.2) and the normalization function $\boldsymbol{\Theta}[\mathbf{x}] = \mathbf{x}/\|\mathbf{x}\|_2$, $\boldsymbol{\Theta}[\mathbf{0}] = \mathbf{0}$), and f_i and f_j denote the focal lengths of the ith and jth cameras, respectively. Hence, in the noiseless case, we make the following observation: Assuming $m_{ij} \geq 2$ (and that we can find at least two ν_{ij}^k not parallel to each other), $\{\nu_{ij}^k\}_{k=1}^{m_{ij}}$ determine a 2D subspace to which $\mathbf{t}_i - \mathbf{t}_j$ is orthogonal (hence the "line" through $\mathbf{t}_i$ and $\mathbf{t}_j$). This 2D subspace spanned by $\{\nu_{ij}^k\}_{k=1}^{m_{ij}}$ can be obtained by, e.g., PCA (i.e., as the subspace spanned by the two left singular vectors of the matrix $\left[\nu_{ij}^1 \dots \nu_{ij}^{m_{ij}} \right]$ corresponding to its two nonzero singular values). However, in the presence of noisy measurements, if we replace R_i, f_i, and $\mathbf{q}_i$ in (4.3) with their estimates, then we essentially obtain noisy samples $\hat{\nu}_{ij}^k$ from these subspaces (for which, e.g., PCA might not produce robust estimates in the presence of outliers among $\hat{\nu}_{ij}^k$). Hence, for pairwise line estimation, our approach is to reduce the effects of noise by employing robust rotation and subspace estimation steps, which we discuss next.

4.1. Rotation estimation. In this section, we provide the details of the rotation estimation step using the pairwise rotation estimates $\hat{R}_{ij}$, extracted from the essential matrix estimates $\hat{E}_{ij}$. Our main objective here is to reduce the effects of outliers in $\hat{R}_{ij}$ in the estimation of the rotations R_i, while preserving computational efficiency. The outliers in $\hat{R}_{ij}$, which mainly occur due to misidentified and/or insufficiently many corresponding points used to estimate the essential matrices, can result in large errors in rotation estimation, which manifest as large errors in the pairwise line estimates through the noisy subspace samples $\hat{\nu}_{ij}^k$ computed

via (4.3). Specifically, for large and sparsely correlated image sets (i.e., image sets for which we can obtain the estimates $\hat{R}_{ij}$ for a relatively small fraction of all pairs), the proportion of outliers in $\hat{R}_{ij}$ is typically large enough to degrade the quality of rotation estimates (e.g., when estimated via a single iteration of the EVM used in [3]). In the literature, there are various algorithms to estimate camera rotations from the pairwise rotation estimates $\hat{R}_{ij}$ (see, e.g., [61, 25, 3, 41]) with various theoretical and experimental robustness properties. Our procedure for estimating the rotations is iterative; i.e., in each iteration we apply EVM in order to detect the outliers in $\hat{R}_{ij}$ with respect to the rotation estimates evaluated at that iteration, and continue to the next iteration by removing the detected outliers. We now provide the details.

We represent the pairwise rotation estimates $\hat{R}_{ij}$ as the edges of a rotation measurement graph $G_R = (V_R, E_R)$, where the set of vertices V_R represents the cameras. We assume that G_R is connected. At the kth iteration of our procedure, we are given a set $\{\hat{R}_{ij}\}_{(i,j)\in E_R^k}$ of pairwise rotation measurements represented by the connected rotation measurement graph $G_R^k = (V_R^k, E_R^k)$. First, we apply EVM to compute the kth iteration rotation estimates $\{\hat{R}_i^k\}_{i\in V_R^k}$. We then identify the outlier edges, denoted by $(i,j) \in E_O^k$, for which the consistency errors $\|(\hat{R}_i^k)^T \hat{R}_j^k - \hat{R}_{ij}^k\|_F$ are relatively large, i.e., "significantly" larger than the mean (one can also identify the outliers as a fixed portion, say 10%, of the edges with largest consistency error, or as the edges with a consistency error above a certain threshold). At this stage, we remove the outlier edges from the measurement graph and identify G_R^{k+1} as the largest connected component of $G' = (V_R^k, E_R^k \setminus E_O^k)$. The iterations are then repeated until a convergent behavior in the consistency errors is observed (one can also repeat the iterations a fixed number of times, up to an allowed number of removed points). We note that since the eigenvector-based method is computationally very efficient, the extra iterations induce a negligible computational cost for the whole algorithm; however, the change in the estimated rotations can significantly improve the final estimation of camera locations.

4.2. Subspace estimation. Let $\bar{G}_R = (\bar{V}_R, \bar{E}_R)$ denote the resulting graph of the rotation estimation step. For each $(i,j) \in \bar{E}_R$, the estimates $\{\hat{\nu}_{ij}^k\}_{k=1}^{m_{ij}}$ are evaluated using the rotation estimates $\{\hat{R}_i\}_{i\in \bar{V}_R}$ in (4.3). $\{\hat{\nu}_{ij}^k\}_{k=1}^{m_{ij}}$ are noisy samples of unit vectors in a 2D subspace. As also mentioned previously, we can estimate this subspace, e.g., by PCA; however, PCA is not robust to outliers in the sample set. There are various algorithms for robust subspace estimation from noisy sample points (e.g., see [37, 59, 68] and references therein), with different performance and convergence guarantees, computational efficiency, and numerical stability. We choose to use the S-REAPER algorithm introduced in [37]. S-REAPER solves the following convex problem:

$$(4.4a) \qquad \underset{Q_{ij}}{\text{minimize}} \ \sum_{k=1}^{m_{ij}} \|\hat{\nu}_{ij}^k - Q_{ij}\hat{\nu}_{ij}^k\|_2$$

$$(4.4b) \qquad \text{subject to} \ \ 0 \preceq Q_{ij} \preceq I_3, \ \text{Tr}(Q_{ij}) = 2 \,.$$

After finding the solution Q_{ij}^* of (4.4), the robust subspace $\hat{Q}_{ij}$ is defined to be the subspace spanned by the two normalized leading eigenvectors, $\mathbf{q}_{1,ij}^*, \mathbf{q}_{2,ij}^*$, of Q_{ij}^*. Hence, we set $\hat{\Gamma}_{ij} := I_3 - (\mathbf{q}_{1,ij}^*(\mathbf{q}_{1,ij}^*)^T + \mathbf{q}_{2,ij}^*(\mathbf{q}_{2,ij}^*)^T)$ as our robust line estimates.

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

Table 1
Algorithm for camera motion estimation.

	Input: Images: $\{I_i\}_{i=1}^n$, focal lengths: $\{f_i\}_{i=1}^n$
Feature points, Essential matrices, Relative rotations	**1.** Find corresponding points between image pairs (using SIFT [40]) **2.** Compute essential matrices $\hat{E}_{ij}$, using RANSAC (for pairs with sufficiently many correspondences) **3.** Factorize $\hat{E}_{ij}$ to compute $\{\hat{R}_{ij}\}_{(i,j)\in E_R}$ and $G_R = (V_R, E_R)$

Rotation estimation (section 4.1)	**4.** Starting with $G_R^1 = G_R$, at the kth iteration:
	  - Given $G_R^k = (V_R^k, E_R^k)$ and $\{\hat{R}_{ij}\}_{(i,j)\in E_R^k}$, compute $\{\hat{R}_i\}_{i\in V_R^k}$ using EVM [3]
	  - Detect outlier edges E_O^k
	  - Identify G_R^{k+1} to be the largest connected component of $G' = (V_R^k, E_R^k \setminus E_O^k)$
	5. Repeat until convergence, output $\{\hat{R}_i\}_{i\in \bar{V}_R}$ and $\bar{G}_R = (\bar{V}_R, \bar{E}_R)$

Pairwise line estimation (section 4.2)	**6.** Compute the 2D subspace samples $\{\hat{\nu}_{ij}^k\}_{k=1}^{m_{ij}}$ for each $(i,j) \in \bar{E}_R$ using (4.3) **7.** Estimate the pairwise lines $\{\hat{\Gamma}_{ij}\}_{(i,j)\in\bar{E}_R}$ using the S-REAPER algorithm (4.4)

Location estimation (section 2)	**8.** Extract the largest maximally parallel rigid component $G_t = (V_t, E_t)$ of $\bar{G}_R$ [35]
	9. If $|V_t|$ is small enough, compute the location estimates $\{\hat{t}_i\}_{i\in V_t}$ by the SDR (2.6)
	9'. If $|V_t|$ is large (with respect to the computational resources):
	  - Partition G_t into parallel rigid patches (section 3.1), form the patch graph G_P
	  - Compute camera location estimates for each patch, using the SDR (2.6)
	  - Compute pairwise patch signs using (3.2), synchronize patches in $\mathbb{Z}_2$ [15]
	  - Estimate patch scales, translations, and camera locations $\{\hat{t}_i\}_{i\in\bigcup P_k}$ by (3.3)

	Output: Camera orientations and translations: $\{\hat{R}_i, \hat{t}_i\}$

5. Experiments.

5.1. Synthetic data experiments. We first provide experimental evaluation of the performance of our SDR (2.6) in $\mathbb{R}^3$ with synthetic line data. The experiments present the SDR performance with respect to the underlying graph (e.g., number of nodes, number of edges), noise model, and noise level. Also, we provide comparisons to the LS method of [3, 9] and the ℓ_∞ method of [49], which also directly use pairwise line information in location estimation. Moreover, we compare the performance of our distributed algorithm to that of the SDR applied directly to the entire graph.

We used a noise model incorporating the effects of small disturbances and outliers. Given a set of locations $\{t_i\}_{i=1}^n$ and $G_t = (V_t, E_t)$, for each $(i,j) \in E_t$, we first let

$$(5.1) \qquad \gamma_{ij} = \begin{cases} \gamma_{ij}^U & \text{with probability } p\,, \\ (t_i - t_j)/\|t_i - t_j\|_2 + \sigma\gamma_{ij}^G & \text{with probability } 1-p \end{cases}$$

and normalized γ_{ij} to obtain $\{\tilde{\gamma}_{ij} = \Theta[\gamma_{ij}]\}_{(i,j)\in E_t}$ as our "directed" lines. Here, $\{\gamma_{ij}^U\}_{(i,j)\in E_t}$ and $\{\gamma_{ij}^G\}_{(i,j)\in E_t}$ are independent and identically distributed (i.i.d.) random variables drawn from uniform distribution on S^2 and standard normal distribution on $\mathbb{R}^3$, respectively. For

the SDR and the LS solvers, we used the (undirected) lines $\Gamma_{ij} = \tilde{\gamma}_{ij}(\tilde{\gamma}_{ij})^T$, while the ℓ_∞ solver required the directed lines $\tilde{\gamma}_{ij}$.

We evaluated the performance of each method in terms of the normalized root mean squared error (NRMSE) given by

$$(5.2) \qquad \text{NRMSE}(\{\hat{\mathbf{t}}_i\}) = \sqrt{\frac{\sum_i \|\hat{\mathbf{t}}_i - \mathbf{t}_i\|_2^2}{\sum_i \|\mathbf{t}_i - \mathbf{t}_c\|_2^2}},$$

where $\hat{\mathbf{t}}_i$ are the location estimates (after removal of the global scale, translation, and negation) and $\mathbf{t}_c$ is the center of mass of $\mathbf{t}_i$.

We performed experiments with fixed parallel rigid graphs with $n = 100$ and $n = 200$ nodes having average and minimum degrees of $n/4$ and $3n/100$, respectively. The original locations $\mathbf{t}_i$ are i.i.d. random variables drawn from standard normal distribution on $\mathbb{R}^3$. The NRMSE values (averaged over 10 random realizations of the noise model and the original locations) are summarized in Table 2. In order to observe the performance of the solvers with different noise structures (and since the LS and ℓ_∞ solvers were already very sensitive to outliers), we performed these experiments with pure small disturbances, i.e., with $p = 0$ (the first three rows of Table 2), and with pure outliers, i.e., $\sigma = 0$ (the last three rows of Table 2).

Table 2

NRMSE (5.2) performance of the SDR (2.6) versus LS [3, 9] and ℓ_∞ [49] solvers. Measurements are generated by the noise model (5.1), and NRMSE values are averaged over 10 trials.

	$n = 100$			$n = 200$		
σ	SDR	LS	ℓ_∞	SDR	LS	ℓ_∞
0.01	**0.0209**	0.0417	0.0619	**0.0178**	0.0525	0.0194
0.05	**0.0752**	1.1947	0.3742	**0.0368**	1.0760	0.7448
0.1	**0.1936**	1.2704	1.0247	**0.1453**	1.3870	0.8976
p	SDR	LS	ℓ_∞	SDR	LS	ℓ_∞
0.01	**0.1049**	1.1584	1.1350	**0.1189**	1.1063	0.8326
0.02	**0.1481**	1.1994	1.0876	**0.1333**	1.1226	1.0825
0.05	**0.2458**	1.2248	1.0689	**0.2064**	1.3848	1.1163

Table 2 indicates that the estimator given by our SDR is robust to both types of noise, and increasing the number of nodes and edge connectivity further improves its accuracy. However, the LS and the ℓ_∞ solvers are sensitive to both kinds of noise. This is mainly due to the collapse phenomenon for the LS solver, and due to the structure of the cost function, which is not robust to large errors, for the ℓ_∞ solver. A collapsing solution of the LS solver is compared to the SDR solution in Figure 9 (the ℓ_∞ solution is not included since it produces a "cloud" of locations unrelated to the ground truth and degrades the visibility of the figure).

We also compare the performance of our distributed algorithm to that of the SDR applied to the whole graph for $n = 200$. For the distributed algorithm, we fix a maximum patch size of $N_{max} = 70$ and divide the whole graph into eight patches. The NRMSE results, summarized in Table 3, demonstrate that the accuracy of the distributed approach is comparable to that of the nondistributed approach, perhaps even slightly better for higher levels of noise.

5.2. Real data experiments. We tested our location estimation algorithm on three sets of real images with different sizes. To solve the SDR (2.6), we used the SDPT3 package from [58]
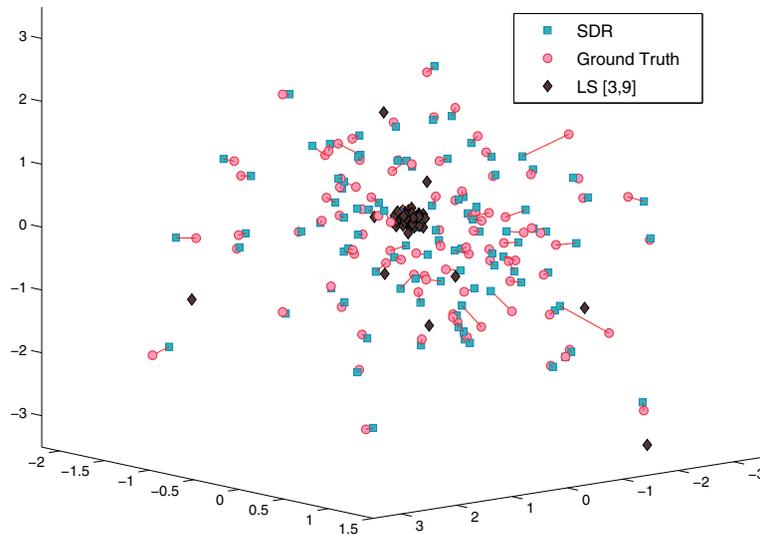
Figure 9. *A sample solution, with $n = 100$, $p = 0$, $\sigma = 0.05$, demonstrating the collapsing behavior of the LS solution. The line segments represent the error incurred by the SDR solution compared to the ground truth.*

Table 3
NRMSE (5.2) results of the SDR solver (2.6) (denoted by "Full") and the distributed SDR solver (denoted by "Dist."); see section 3 for the distributed SDR.

	$p = 0.01$		$p = 0.02$		$p = 0.05$	
σ	Full	Dist.	Full	Dist.	Full	Dist.
0.01	**0.1089**	0.1175	**0.1255**	0.1342	0.1957	**0.1852**
0.02	**0.1174**	0.1207	0.1381	**0.1364**	0.2064	**0.1960**
0.05	0.1426	**0.1385**	**0.1490**	0.1523	0.2137	**0.2058**

for small data sets (for its superlinear convergence with respect to the level of accuracy), and for the large data set, we used ADM (Algorithm 1). To construct a sparse 3D structure in our experiments, we used the parallel bundle adjustment (PBA) algorithm of [67]. We also used the patch-based multi-view stereo (PMVS) algorithm of [19] to evaluate a dense 3D reconstruction. We performed our computations on multiple workstations with Intel Xeon X7542 CPUs, each with six cores, running at 2.67 GHz. In order to directly compare the accuracy of the location estimation by SDR to that of LS [3, 9] and ℓ_∞ [49] solvers, we fed all three solvers with the orientation estimates produced by our iterative solver (section 4.1) and the robust subspace estimates (section 4.2), which produced more accurate estimates for all data sets.

5.3. Small data sets. First we provide our results for the small Fountain-P11 and HerzJesu-P25 data sets of [54], which include 11 and 25 images, respectively. For camera calibration and performance evaluation, we used the focal lengths and ground truth locations provided with the data. For these small data sets, we did not require a distributed approach. For both data

sets, the SDR (2.6) estimated the camera locations very accurately in less than a second, and the solutions of (2.6) had rank equal to 1, demonstrating the tightness of our relaxation. We also ran the PBA algorithm, resulting in an average reprojection error of 0.17 pixels for the Fountain-P11 data and 0.43 pixels for the HerzJesu-P25 data, to construct the 3D structures given in Figure 10. We summarize and compare our end results to previous works[5] in Table 4.

<div align="center">

Table 4

Location estimation errors for the Fountain-P11 and HerzJesu-P25 data sets.

	Error (in meters)	
Method	Fountain-P11	HerzJesu-P25
SDR (2.6)	**0.0002**	**0.0053**
LS [3, 9]	0.0048	0.0054
ℓ_∞ [49]	0.0064	0.0253
Linear method of [50]	0.1317	0.2538
Bundler [52]	0.0072	0.0308

</div>

5.4. Large data set. The next set of images is the larger Notre Dame Cathedral set from [51], composed of 715 images. This is an irregular collection of images, and hence estimating the camera locations for all of these images (or a large subset) is challenging. Even for orientation estimation on this data set, previous works usually discard a large subset (see, e.g., [25, 3]). In our experiment, we used the focal lengths and radial distortion corrections from [51]. We can accurately and efficiently estimate the camera locations for a subset of size 637. This reduction in size is due to our rotation estimation step (section 4.1) (we can robustly recover rotations for a subset of size 687) and due to the node removals during the distributed algorithm (section 3). We partitioned the whole graph into patches of sizes smaller than 150 (section 3.1), getting 20 rigid patches in less than a minute (where extraction of parallel rigid components took a total of 42 seconds). The related SDRs were solved in parallel, in about 21 minutes. Finally, the stitching of the patches (sections 3.2 and 3.3) took 57 seconds.

We assessed the performance of our algorithm in terms of the NRMSE measure (5.2), using the location estimates of [51] as the ground truth. We first got an NRMSE of 0.104, and then applied PBA once (with an initial 3D structure of about 204,000 points) to get an NRMSE of 0.054, and an average reprojection error of 0.43 pixels, in less than 2 minutes. The resulting 3D structure is provided in Figure 11.

We also compared our results to those of the LS and ℓ_∞ solvers applied to the whole graph and in a distributed fashion. In both cases, the LS and ℓ_∞ solvers resulted in very large errors[6] (also, because of the very low quality of initial 3D structures for these solutions, PBA produced no improvements). The NRMSE results are summarized in Table 5. In addition, a snapshot of the location estimates for our distributed algorithm and the LS solver, demonstrating the collapsing LS solution, is provided in Figure 12.

---

[5]Results of [50] are cited from [3].

[6]We note that (a slightly modified version of) the LS solver in [3] achieves reasonable accuracy for the Notre Dame data set when a significant number of the images are discarded.
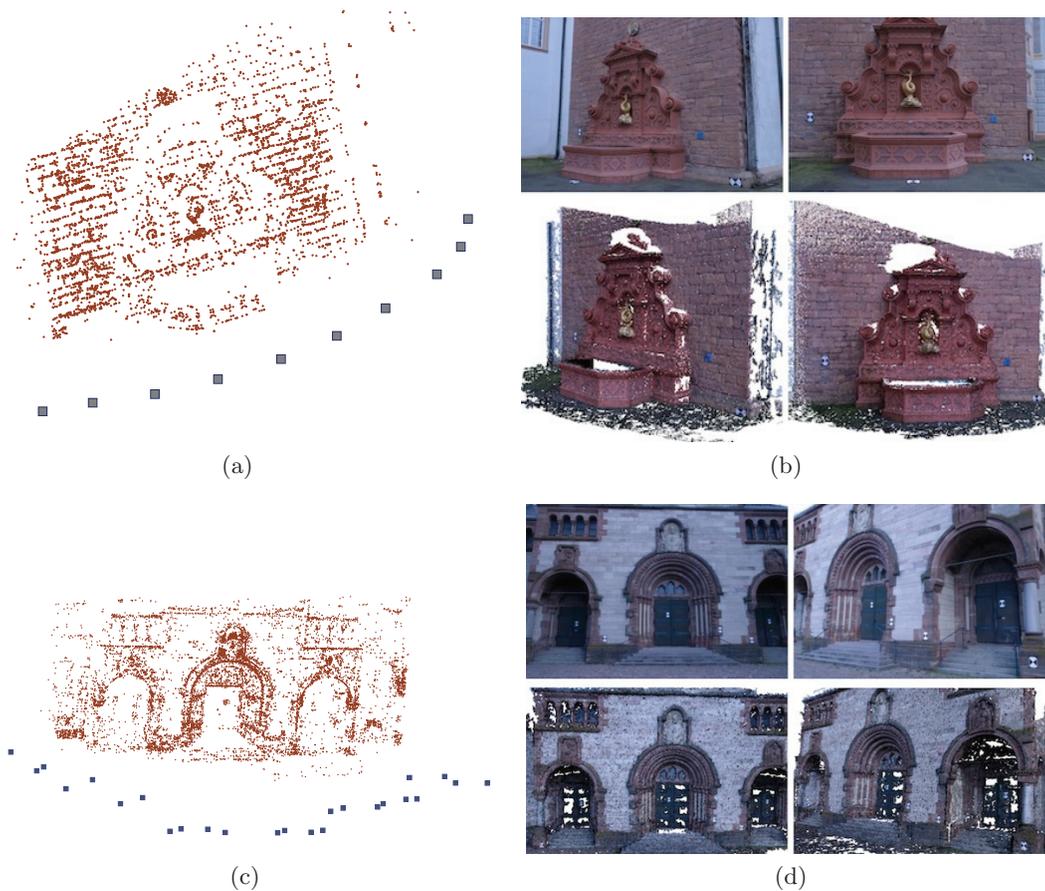
Figure 10. (a) *Sparse* 3D *structure and* 11 *estimated camera locations (in blue) for the Fountain-P*11 *data.* (b) *Sample images and snapshots of the dense* 3D *reconstruction.* (c) *Sparse* 3D *structure and* 25 *estimated camera locations (in blue) for the HerzJesu-P*25 *data.* (d) *Sample images and snapshots of the dense* 3D *reconstruction.*

Table 5
Location estimation errors for the Notre Dame data.

Method	NRMSE
Distributed SDR	0.104
Distributed SDR (followed by PBA)	0.054
Distributed LS [3, 9]	1.087
LS [3, 9]	1.392
Distributed ℓ_∞ [49]	1.125
ℓ_∞ [49]	1.273

6. Conclusion and future work. We formulated a semidefinite relaxation for estimating positions from pairwise line measurements and applied it to the problem of camera location estimation in structure from motion. We elucidated the importance of parallel rigidity in determining uniqueness and well-posedness of the estimation problem and provided rigorous analysis for stability of our SDR. Also, we introduced an alternating direction augmented
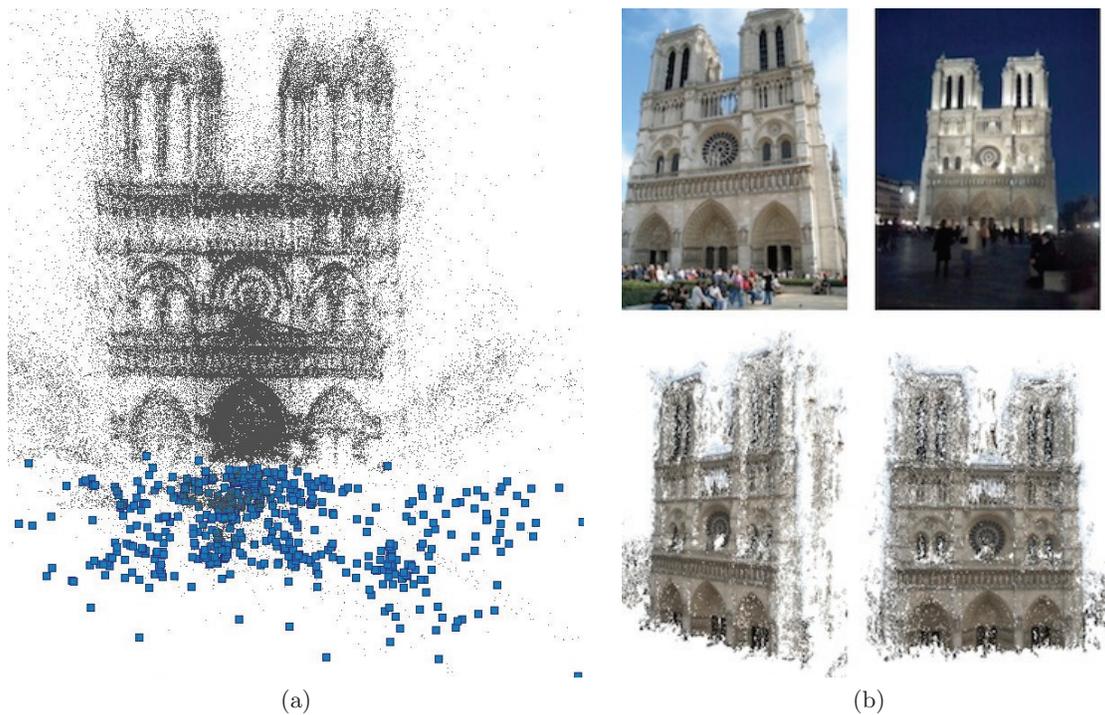
Figure 11. (a) *Sparse 3D structure of about* 204,000 *points and (some of)* 637 *estimated camera locations (in blue) for the Notre Dame data.* (b) *Sample images and snapshots of the dense 3D reconstruction.*

Lagrangian method (ADM) to solve the SDR and an efficient, distributed version of our SDR designed to handle very large sets of input images.

In the context of structure from motion (SfM), our algorithm can be used to efficiently and stably obtain the locations and orientations of the corresponding cameras, i.e., the camera motion, in order to produce a high-quality initial point for reprojection error minimization algorithms, as demonstrated by our experiments on real image sets. We also note that, for collections of images taken from an approximately planar surface (which is usually the case for images of large 3D structures taken by visitors), the pairwise lines can also be well approximated to lie within the same plane, and hence the SDR can take this prior knowledge to allow a more efficient (and, perhaps, more accurate) computation of camera locations rendered by the reduction of the problem from three to two dimensions. In addition, in SfM, since the sign information (i.e., the directions) of the pairwise lines between camera locations can be estimated, an optimization framework that uses this additional information is a topic for further study (see, e.g., [45, 56]).

As future work, we plan to investigate and explain the tightness of the SDR, i.e., to characterize the conditions under which it returns a rank-1 matrix as its solution. Also, we plan to apply the SDR to the problem of sensor network localization with bearing information.

Appendix A. Parallel rigidity. In this appendix, we review fundamental concepts and results in parallel rigidity theory (also see [16, 17, 64, 63] and the references therein). We begin with the concept of *point formation*. A d-dimensional point formation $\mathbb{F}_{\mathbf{p}}$ at $\mathbf{p} = [\, \mathbf{p}_1^T \; \mathbf{p}_2^T \; \cdots \; \mathbf{p}_n^T \,]^T$

Figure 12. 637 *camera locations of the Notre Dame data set (with* 715 *images) estimated using the distributed SDR. The solution of* [51] *is taken as the ground truth. The collapsing solution of the LS solver of* [3, 9] *is also provided.*

is a set of n points $\{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n\} \subseteq \mathbb{R}^d$ (assumed to be separate, i.e., $\mathbf{p}_i \neq \mathbf{p}_j$), together with a set E of *links*, satisfying $E \subseteq \{(i, j) : i \neq j, \, i, j \in \{1, 2, \ldots, n\}\}$. For the camera location estimation problem, we think of the points $\mathbf{p}_i$ as representing the camera locations, and the pairs $(i, j) \in E$ are used to denote the camera pairs, between which there is a pairwise measurement. Note that each formation $\mathbb{F}_\mathbf{p}$ uniquely determines a *graph* $G_{\mathbb{F}_\mathbf{p}} := (V, E)$, having the vertex set $V = \{1, 2, \ldots, n\}$ and the edge set E of $\mathbb{F}_\mathbf{p}$, and also a *measurement function* $\Gamma_{\mathbb{F}_\mathbf{p}}$, whose value at $(i, j) \in E$ is the measured quantity between $\mathbf{p}_i$ and $\mathbf{p}_j$ (to keep the notation simple in section 2.1, by abuse of notation, we refer to the set $\{\Gamma_{ij}\}_{(i,j) \in E_t}$ of measurements $\Gamma_{ij} := \Gamma_{\mathbb{F}_\mathbf{p}}(i, j)$, defined on E_t of $G_t = (V_t, E_t)$, as *the formation*). In order to represent the *pairwise line measurements*, we use a measurement function given by $\Gamma_{\mathbb{F}_\mathbf{p}}(i, j) = (\mathbf{p}_i - \mathbf{p}_j)(\mathbf{p}_i - \mathbf{p}_j)^T / \|\mathbf{p}_i - \mathbf{p}_j\|_2^2$. In the literature (see, e.g., [16, 17]), the pairwise measurements are considered only in terms of the *direction constraints* they imply. These constraints are used to define *parallel point formations* $\mathbb{F}_\mathbf{q}$ of a formation $\mathbb{F}_\mathbf{p}$ (explained below), and are homogeneous equations given by

$$(\mathbf{p}_i - \mathbf{p}_j)_{N_1}^T (\mathbf{q}_i - \mathbf{q}_j) = 0, \quad (i, j) \in E,$$

$$\vdots$$

$$(\mathbf{p}_i - \mathbf{p}_j)_{N_{d-1}}^T (\mathbf{q}_i - \mathbf{q}_j) = 0, \quad (i, j) \in E,$$

where $(\mathbf{p}_i - \mathbf{p}_j)_{N_i}$ for $i = 1, \ldots, d-1$ are (linearly independent) vectors that span the subspace

orthogonal to $p_i - p_j$ and, therefore, uniquely define the line between $\mathbf{p}_i$ and $\mathbf{p}_j$. We use the measurement function $\Gamma_{\mathbb{F}_\mathbf{p}}$ to compactly represent these equations by $(I_d - \Gamma_{\mathbb{F}_\mathbf{p}}(i,j))(\mathbf{q}_i - \mathbf{q}_j) = 0, (i,j) \in E$. Also, note that, for the camera location estimation problem, $\Gamma_{\mathbb{F}_\mathbf{p}}$ encapsulates the maximal information we can extract from the epipolar constraints; e.g., we cannot estimate the *signed pairwise lines* $(\mathbf{p}_i - \mathbf{p}_j)/\|\mathbf{p}_i - \mathbf{p}_j\|_2$ based solely on the epipolar constraints (see section 4 for further details).

Two point sets $\{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n\}$ and $\{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n\}$ in $\mathbb{R}^d$ are said to be *congruent* if there exist $\mathbf{x} \in \mathbb{R}^d$ and $c \in \mathbb{R}$ such that $c\mathbf{p}_i + \mathbf{x} = \mathbf{q}_i$ for all $i \in \{1, 2, \ldots, n\}$, i.e., congruent point sets can be obtained from one another by translation, scaling, or negation. A point formation $\mathbb{F}_\mathbf{p}$ that is uniquely determined, up to congruence, by its graph $G_{\mathbb{F}_\mathbf{p}}$ and its measurement function $\Gamma_{\mathbb{F}_\mathbf{p}}$ is called *globally parallel rigid.*

For a given formation $\mathbb{F}_\mathbf{p}$ in $\mathbb{R}^d$, a *parallel point formation* $\mathbb{F}_\mathbf{q}$ is a point formation (with the same graph $G_{\mathbb{F}_\mathbf{p}} = (V, E)$) in which $\mathbf{p}_i - \mathbf{p}_j$ is parallel to $\mathbf{q}_i - \mathbf{q}_j$ for all (i, j) in E (i.e., $\Gamma_{\mathbb{F}_\mathbf{p}}(i,j) = \Gamma_{\mathbb{F}_\mathbf{q}}(i,j)$ on E). It is clear that congruence transformations, i.e., translations, scalings, and negation, produce *trivial* parallel point formations of the original point formation; any other parallel formation is termed *nontrivial*. A point formation that does not admit any nontrivial parallel point formations is called a *parallel rigid* point formation; otherwise it is called *flexible* (see Figure 4 for a simple example). We note that, in contrast to the case of classical rigidity involving distance measurements, equivalence of global parallel rigidity and (simple) parallel rigidity turns out to be a rephrasing of definitions (also see [64, 31, 34, 16]).

The concept of parallel point formations allows us to obtain a linear algebraic characterization: Given a point formation $\mathbb{F}_\mathbf{p}$ with the graph $G_{\mathbb{F}_\mathbf{p}} = (V, E)$, $\mathbb{F}_\mathbf{q}$ is a parallel formation if and only if its point set satisfies

$$(A.1) \qquad \left(I_d - \Gamma_{\mathbb{F}_\mathbf{p}}(i,j)\right)(\mathbf{q}_i - \mathbf{q}_j) = 0 \, , \ (i,j) \in E \, ,$$

which can be rewritten in matrix form as

$$(A.2) \qquad R_{\mathbb{F}_\mathbf{p}} \mathbf{q} = 0 \, ,$$

where $R_{\mathbb{F}_\mathbf{p}} \in \mathbb{R}^{d|E| \times d|V|}$ is termed *the parallel rigidity matrix* of the formation $\mathbb{F}_\mathbf{p}$ (see, e.g., [16], for a slightly different, but equivalent, formulation). Here, point sets of the trivial parallel formations of $\mathbb{F}_\mathbf{p}$ span a $(d+1)$-dimensional subspace of the nullspace of $R_{\mathbb{F}_\mathbf{p}}$. As a result, $\mathbb{F}_\mathbf{p}$ is parallel rigid if and only if $\dim(\mathcal{N}(R_{\mathbb{F}_\mathbf{p}})) = d+1$, i.e., $\text{rank}(R_{\mathbb{F}_\mathbf{p}}) = d|V| - (d+1)$ (note that $R_{\mathbb{F}_\mathbf{p}}^T R_{\mathbb{F}_\mathbf{p}}$ is the matrix L of the linear cost function in (2.6) with noiseless measurements).

Now we consider the generic properties of formations. A point $\mathbf{p}$ in $\mathbb{R}^{dn}$ (or the point set $\{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n\}$ in $\mathbb{R}^d$) is said to be *generic* if its dn coordinates are algebraically independent, i.e., there is no nonzero polynomial ψ on $\mathbb{R}^{dn}$, with integer coefficients, satisfying $\psi(\mathbf{p}) = 0$ (for a more general definition see [24]). The set of generic points is an open dense subset of $\mathbb{R}^{dn}$. A graph $G = (V, E)$ is called *generically parallel rigid* (in $\mathbb{R}^d$) if, for a generic point $\mathbf{p} \in \mathbb{R}^{d|V|}$, the formation $\mathbb{F}_\mathbf{p}$ having the underlying graph G is parallel rigid (in fact, if G of $\mathbb{F}_\mathbf{p}$ is generically parallel rigid, then $\mathbb{F}_\mathbf{p}$ is parallel rigid for all generic $\mathbf{p}$). Also, a formation $\mathbb{F}_\mathbf{p}$ is called a *generically parallel rigid* formation (in $\mathbb{R}^d$) if its underlying graph is generically parallel rigid (in $\mathbb{R}^d$). Generic parallel rigidity is a combinatorial property of the underlying graph as characterized by Theorem 2.1 (also see [16, 17, 34, 64, 63, 31]). Combinatorial conditions of

Theorem 2.1 also translate to efficient combinatorial algorithms for generic parallel rigidity testing (see, e.g., [32]).

There is also a linear algebraic characterization of generically parallel rigid formations using the notion of *generic rank:* The generic rank of the parallel rigidity matrix R is given by $\mathrm{GenericRank}(R) = \max\{\mathrm{rank}(R_{\mathbb{F}_{\mathbf{p}}}), \ \mathbf{p} \in \mathbb{R}^{d|V|}\}$, which clearly depends only on the underlying graph. A formation $\mathbb{F}_{\mathbf{p}}$ (and hence its underlying graph $G = (V, E)$) is generically parallel rigid in $\mathbb{R}^d$ if and only if $\mathrm{GenericRank}(R) = d|V| - (d+1)$. Also, the set of points with maximal rank, i.e., $\{\mathbf{p} \in \mathbb{R}^{d|V|} : \mathrm{rank}(R_{\mathbb{F}_{\mathbf{p}}}) = \mathrm{GenericRank}(R)\}$, is an open dense subset of $\mathbb{R}^{d|V|}$. As a result, similar to the randomized algorithm of [29] for (classical) generic rigidity testing, we propose the following randomized test for generic parallel rigidity:

1. Given a formation on the graph $G = (V, E)$, randomly sample $\mathbf{p} \in \mathbb{R}^{d|V|}$ from an absolutely continuous (with respect to Lebesgue measure) probability measure (e.g., let $\mathbf{p}$ be sampled from i.i.d. Gaussian distribution), and centralize $\mathbf{p}$ such that $\sum_{i=1}^{|V|} \mathbf{p}_i = \mathbf{0}_d$.

2. Construct an orthogonal basis for the trivial nullspace of $R_{\mathbb{F}_{\mathbf{p}}}$: For $\mathbf{u}_i = \mathbf{1}_{|V|} \otimes \mathbf{e}_i$ ($\mathbf{e}_i$ denoting the ith canonical basis vector in $\mathbb{R}^d$), such a basis is given by $\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_d, \mathbf{p}\}$.

3. To check whether $\mathrm{rank}(R_{\mathbb{F}_{\mathbf{p}}}) = d|V| - (d+1)$ or not, compute the smallest eigenvalue $\lambda_1(W_{\mathbb{F}_{\mathbf{p}}})$ of $W_{\mathbb{F}_{\mathbf{p}}} = R_{\mathbb{F}_{\mathbf{p}}}^T R_{\mathbb{F}_{\mathbf{p}}} + U_{\mathbf{p}} U_{\mathbf{p}}^T$, where $U_{\mathbf{p}} = [\,\mathbf{u}_1 \ \mathbf{u}_2 \ \cdots \ \mathbf{u}_d \ \mathbf{p}\,]$. If $\lambda_1(W_{\mathbb{F}_{\mathbf{p}}}) > \epsilon$ (where ϵ is a small positive constant set to prevent numerical precision errors), declare G to be generically parallel rigid; otherwise declare G to be flexible.

This randomized test correctly decides (up to precision errors) the generic rigidity of G with probability 1 (i.e., up to precision errors, our procedure can produce a false negative with probability 0 and does not produce a false positive). Also note that $\lambda_1(W_{\mathbb{F}_{\mathbf{p}}}) = \lambda_{dn}(W_{\mathbb{F}_{\mathbf{p}}}) + \lambda_1(W_{\mathbb{F}_{\mathbf{p}}} - \lambda_{dn}(W_{\mathbb{F}_{\mathbf{p}}})I_{dn})$, and since $W_{\mathbb{F}_{\mathbf{p}}}$ is positive semidefinite, $\lambda_{dn}(W_{\mathbb{F}_{\mathbf{p}}})$ and $\lambda_1(W_{\mathbb{F}_{\mathbf{p}}} - \lambda_{dn}(W_{\mathbb{F}_{\mathbf{p}}})I_{dn})$ are largest magnitude eigenvalues, which can be computed (e.g., by the power method[7]) in $\mathcal{O}(m)$ time. For fixed d, this yields a time complexity of $\mathcal{O}(m)$ (dominated by the complexity of step 2) for the randomized test (compare to the time complexity $O(n^2)$ of the pebble game algorithm, which is, however, an integer algorithm, i.e., is free of numerical errors). Also, considering its simplicity, we choose to use this test for testing parallel rigidity.

Appendix B. Proof of Theorem 2.3. Let $\tilde{c}$ denote the suboptimal constant for (2.7) given by $\tilde{c} = (\mathrm{Tr}(T^*))^{-1}$. First we consider the decomposition of $\tilde{c}T^*$ in terms of the nullspace $\mathcal{S}$ of L^0 and its complement $\bar{\mathcal{S}}$; that is, we let

(B.1)
$$\tilde{c}T^* = X + Y + Z$$

where $X \in \mathcal{S} \otimes \mathcal{S}$, $Y \in \bar{\mathcal{S}} \otimes \bar{\mathcal{S}}$, and $Z \in (\bar{\mathcal{S}} \otimes \mathcal{S}) \oplus (\mathcal{S} \otimes \bar{\mathcal{S}})$.[8] Using this decomposition, we can bound $\delta(T^*, T_0)$ in terms of $\mathrm{Tr}(Y)$.

Lemma B.1.

(B.2)
$$\delta(T^*, T_0) \leq \|\tilde{c}T^* - T_0\|_F \leq \sqrt{2\,\mathrm{Tr}(Y)}\,.$$

---

[7]Although every iteration of the power method has time complexity $\mathcal{O}(m)$, the number of iterations is greater than $\mathcal{O}(1)$ as it depends on the spectral gap of $W_{\mathbb{F}_{\mathbf{p}}}$.

[8]For two subspaces $\mathcal{U}$ and $\mathcal{V}$, the tensor product $\mathcal{U} \otimes \mathcal{V}$ denotes the space of matrices spanned by rank-1 matrices $\{\mathbf{u}\mathbf{v}^T : \mathbf{u} \in \mathcal{U}, \ \mathbf{v} \in \mathcal{V}\}$.

Proof. First we note that the orthogonal decomposition (B.1) and $T_0 \in \mathcal{S} \otimes \mathcal{S}$ imply

$$(B.3) \qquad \delta^2(T^*, T_0) \leq \|\tilde{c}T^* - T_0\|_F^2 = \|X - T_0\|_F^2 + \|Y\|_F^2 + \|Z\|_F^2 .$$

Here, since the formation is parallel rigid, we get $\dim(\mathcal{S}) = d+1$, where $\mathcal{S}$ is spanned by the d eigenvectors of H (corresponding to its only nonzero eigenvalue n) and $\mathbf{t}^0$. Since $\mathrm{Tr}(\tilde{c}T^*H) = 0$, we get $X = a\mathbf{t}^0(\mathbf{t}^0)^T$ (with $1 \geq a = \tilde{c}(\mathbf{t^0})^T T^* \mathbf{t^0} \geq 0$). Also, since $\mathrm{Tr}(\tilde{c}T^*) = 1$ and $\mathrm{Tr}(Z) = 0$, we get

$$(B.4) \qquad \mathrm{Tr}(Y) = 1 - \mathrm{Tr}(X) = 1 - a = \mathrm{Tr}((1-a)\mathbf{t}^0(\mathbf{t}^0)^T) = \mathrm{Tr}(T_0 - X) = \|X - T_0\|_F .$$

We now consider the spectral decomposition of L^0 given by $L^0 = U_0 \Sigma_0 U_0^T$, where $U_0 = \begin{bmatrix} \mathbf{v}_1 \ \mathbf{v}_2 \ ... \ \mathbf{v}_d \ \mathbf{t}^0 \ \mathbf{s}_{d+2} \ ... \ \mathbf{s}_{dn} \end{bmatrix}$ for $\mathbf{v}_i$ denoting the d eigenvectors of H and $\{\mathbf{s}_{d+2}, \dots, \mathbf{s}_{dn}\}$ is an arbitrary (orthonormal) basis for $\bar{\mathcal{S}}$. The representations of X, Y, Z in this basis, given by $\tilde{X} = U_0^T X U_0$, $\tilde{Y} = U_0^T Y U_0$, and $\tilde{Z} = U_0^T Z U_0$, are as follows: $\tilde{X}$ has a single nonzero entry $\tilde{X}_{d+1,d+1} = a$, $\tilde{Y}$ is supported on its lower $(dn - (d+1)) \times (dn - (d+1))$ block, and $\tilde{Z}$ is supported on its $(d+1)$th row and column except the first $d+1$ entries. Hence we get

$$(B.5) \qquad \tilde{c}U_0^T T^* U_0 = \begin{bmatrix} 0_{d \times d} & & 0_{d \times (dn-d)} \\ & a & \tilde{\mathbf{z}}^T \\ 0_{(dn-d) \times d} & \tilde{\mathbf{z}} & \tilde{Y} \end{bmatrix} \succeq 0 ,$$

where $\tilde{\mathbf{z}}$ denotes the nonzero entries of $\tilde{Z}$. By a Schur complement argument, (B.5) implies

$$(B.6) \qquad \tilde{Y} - \frac{\tilde{\mathbf{z}}\tilde{\mathbf{z}}^T}{1 - \mathrm{Tr}(\tilde{Y})} \succeq 0 \ \Rightarrow \ 2\,\mathrm{Tr}(Y)(1 - \mathrm{Tr}(Y)) \geq 2\|\tilde{\mathbf{z}}\|_2^2 = \|\tilde{Z}\|_F^2 = \|Z\|_F^2 ,$$

where we use $a = 1 - \mathrm{Tr}(Y)$ and $\mathrm{Tr}(\tilde{Y}) = \mathrm{Tr}(Y)$. Combining (B.3), (B.4), (B.6) with the simple fact $\|Y\|_F \leq \mathrm{Tr}(Y)$, we get the assertion of Lemma B.1. $\blacksquare$

The next step is to bound $\mathrm{Tr}(Y)$. We provide the result in Lemma B.2, where, L_G denotes the Laplacian of the graph G_t and the parameter κ is given by $\kappa = (\min_{(i,j) \in E_t} \|\mathbf{t}_i^0 - \mathbf{t}_j^0\|_2^2)^{-1}$.

Lemma B.2.

$$(B.7) \qquad \mathrm{Tr}(Y) \leq \alpha_1 \epsilon \|\tilde{c}T^* - T_0\|_F + \alpha_2 \epsilon^2 ,$$

where $\alpha_1 = \frac{\sqrt{2}m}{\lambda_{d+2}(L^0)}$ and $\alpha_2 = (\frac{\kappa\sqrt{d}\|L_G\|_F}{m} + 1)\frac{\lambda_n(L_G)}{\lambda_{d+2}(L^0)}$.

Proof. We start with a (loose) bound on $\mathrm{Tr}(Y)$, given by

$$(B.8) \qquad \mathrm{Tr}(L^0(\tilde{c}T^*)) = \mathrm{Tr}(L^0 Y) \geq \mathrm{Tr}(\lambda_{d+2}(L^0)Y) = \lambda_{d+2}(L^0)\,\mathrm{Tr}(Y) .$$

Now, in order to bound $\mathrm{Tr}(L^0(\tilde{c}T^*))$, we consider the following partitioning:

$$(B.9a) \qquad \mathrm{Tr}(L^0(\tilde{c}T^*)) = \mathrm{Tr}((L^0 - L)(\tilde{c}T^*)) + \tilde{c}\,\mathrm{Tr}(LT^*)$$

$$(B.9b) \qquad = \mathrm{Tr}((L^0 - L)(\tilde{c}T^* - T_0)) - \mathrm{Tr}(LT_0) + \tilde{c}\,\mathrm{Tr}(LT^*)$$

$$(B.9c) \qquad \leq \mathrm{Tr}((L^0 - L)(\tilde{c}T^* - T_0)) + (\tilde{c}\kappa - 1)\,\mathrm{Tr}(LT_0) ,$$

where L is given by (2.4); (B.9b) follows from $\text{Tr}(L^0 T_0) = 0$; and (B.9c) holds by the feasibility of κT_0 and the optimality of T^* for the SDR. Now, we use the noise model to bound the terms in (B.9c). For the first term, we get

$$\text{(B.10a)} \qquad \text{Tr}((L^0 - L)(\tilde{c}T^* - T_0)) \leq \|L^0 - L\|_F \|\tilde{c}T^* - T_0\|_F$$

$$\text{(B.10b)} \qquad \leq \sum_{i \sim j} \|(\gamma_{ij}^0 + \epsilon_{ij})(\gamma_{ij}^0 + \epsilon_{ij})^T - \gamma_{ij}^0(\gamma_{ij}^0)^T\|_F \|\tilde{c}T^* - T_0\|_F$$

$$\text{(B.10c)} \qquad = \sum_{i \sim j} \left(\|\epsilon_{ij}\|_2^4 + 4\|\epsilon_{ij}\|_2^2 \epsilon_{ij}^T \gamma_{ij}^0 + 2(\epsilon_{ij}^T \gamma_{ij}^0)^2 + 2\|\epsilon_{ij}\|_2^2\right)^{\frac{1}{2}} \|\tilde{c}T^* - T_0\|_F$$

$$\text{(B.10d)} \qquad = \sum_{i \sim j} \left(2\|\epsilon_{ij}\|_2^2 - \|\epsilon_{ij}\|_2^4/2\right)^{\frac{1}{2}} \|\tilde{c}T^* - T_0\|_F$$

$$\text{(B.10e)} \qquad \leq \epsilon \sqrt{2}m \|\tilde{c}T^* - T_0\|_F \,,$$

where (B.10d) follows from $\|\gamma_{ij}\|_2^2 = \|\gamma_{ij}^0 + \epsilon_{ij}\|_2^2 = 1 + 2\epsilon_{ij}^T \gamma_{ij}^0 + \|\epsilon_{ij}\|_2^2 = 1$. For the second term in (B.9c), first we have

$$\text{(B.11a)} \qquad \text{Tr}(LT_0) = \text{Tr}((L - L^0)T_0)$$

$$\text{(B.11b)} \qquad = \sum_{i \sim j} \text{Tr}((\gamma_{ij}^0(\gamma_{ij}^0)^T - \gamma_{ij}\gamma_{ij}^T)\gamma_{ij}^0(\gamma_{ij}^0)^T)\|\mathbf{t}_i^0 - \mathbf{t}_j^0\|_2^2$$

$$\text{(B.11c)} \qquad = -\sum_{i \sim j} ((\epsilon_{ij}^T \gamma_{ij}^0)^2 + 2\epsilon_{ij}^T \gamma_{ij}^0)\|\mathbf{t}_i^0 - \mathbf{t}_j^0\|_2^2$$

$$\text{(B.11d)} \qquad = \sum_{i \sim j} \left(\|\epsilon_{ij}\|_2^2 - \frac{\|\epsilon_{ij}\|_2^4}{4}\right) \|\mathbf{t}_i^0 - \mathbf{t}_j^0\|_2^2$$

$$\text{(B.11e)} \qquad \leq \epsilon^2 \lambda_n(L_G) \,.$$

In order to bound the multiplier $\tilde{c}\kappa - 1$, we use the feasibility of T^*:

$$\text{(B.12a)} \qquad m \leq \sum_{i \sim j} \text{Tr}(C^{ij}T^*) = \text{Tr}((L_G \otimes I_d)T^*) \leq \sqrt{d}\|L_G\|_F \text{Tr}(T^*)$$

$$\text{(B.12b)} \qquad \Rightarrow \tilde{c}\kappa - 1 \leq \frac{\kappa}{\text{Tr}(T^*)} + 1 \leq \frac{\kappa\sqrt{d}\|L_G\|_F}{m} + 1 \,.$$

Finally, combining (B.8), (B.9c), (B.10e), (B.11e), and (B.12b), we obtain the claim (B.7) of Lemma B.2. ∎

We now use Lemmas B.1 and B.2 to obtain our *SDR stability* result.

Proof of Theorem 2.3. The (second) inequality in (B.2) and (B.7) provide a quadratic inequality for $\|\tilde{c}T^* - T_0\|_F$, given by

$$\text{(B.13)} \qquad \|\tilde{c}T^* - T_0\|_F^2 - 2\epsilon\alpha_1 \|\tilde{c}T^* - T_0\|_F - 2\epsilon^2\alpha_2 \leq 0 \,.$$

Examining the roots of this polynomial immediately yields

$$\text{(B.14)} \qquad \delta(T^*, T_0) \leq \|\tilde{c}T^* - T_0\|_F \leq \epsilon \left[\alpha_1 + \left(\alpha_1^2 + 2\alpha_2\right)^{\frac{1}{2}}\right] \,,$$

which was to be shown. ∎

Appendix C. Construction of ADM framework. We first introduce some notation: $\mathbf{1}_m$ and $\mathbf{0}_m$ denote all-ones and all-zeros vectors of dimension m. $\mathrm{S}^{m \times m}$ denotes the space of $m \times m$ symmetric matrices. For $X \in \mathbb{R}^{m \times m}$, $\mathrm{vec}(X)$ is the vector in $\mathbb{R}^{m^2}$ that contains the columns of X, each stacked on top of the next in the order in which they appear in the matrix, and $\mathrm{mat}(\mathbf{x})$ is the matrix X such that $\mathbf{x} = \mathrm{vec}(X)$. We also use 0 for the all-zeros matrix when its dimensions are obvious from the context.

In order to rewrite the SDR in standard form, consider the matrix variable X and the (nonnegative) slack variable $\nu \in \mathbb{R}^m$ satisfying

(C.1a)
$$X = \begin{bmatrix} T & 0 \\ 0 & \mathrm{Diag}(\nu) \end{bmatrix},$$

(C.1b)
$$1 = \mathrm{Tr}(C^{ij}T) - \nu_{ij} = \mathrm{Tr}(A_{ij}^{(1)}X), \ (i,j) \in E_t,$$

where $A_{ij}^{(1)}$ is the matrix corresponding to the linear functional $X \to \mathrm{Tr}(C^{ij}T) - \nu_{ij}$. Let M denote the number of (a priori) zero entries of X, i.e., $M = 2dnm + m(m-1)$. We can also replace (C.1a) by a set of linear equalities in X given by $\mathrm{Tr}(A_k^{(2)}X) = 0$ for $k = 1, \ldots, M/2$. Hence, for $X = \begin{bmatrix} T & \star \\ \star & \star \end{bmatrix}$, we get

$$\left\{ \begin{array}{c} T \succeq 0, \\ \mathrm{Tr}(C^{ij}T) \geq 1, \ (i,j) \in E_t \end{array} \right\} \iff \left\{ \begin{array}{c} X \succeq 0, \\ \mathrm{Tr}(A_{ij}^{(1)}X) = 1, \ (i,j) \in E_t, \\ \mathrm{Tr}(A_k^{(2)}X) = 0, \ k = 1, \ldots, M/2 \end{array} \right\}.$$

Now, we can rewrite (2.6) in standard form (ignoring the equality constraint $\mathrm{Tr}(HT) = 0$ for now)

(C.2a)
$$\underset{X}{\mathrm{minimize}} \quad \mathrm{Tr}(WX)$$

(C.2b)
$$\text{subject to} \quad \mathcal{A}(X) = \mathbf{b},$$

(C.2c)
$$X \succeq 0,$$

where $W = \begin{bmatrix} L & 0 \\ 0 & 0 \end{bmatrix}$ (for L given by (2.4)), $\mathbf{b} = \begin{bmatrix} \mathbf{1}_m \\ \mathbf{0}_{M/2} \end{bmatrix}$, and the linear operator $\mathcal{A}$ satisfies $\mathcal{A}(X) = \begin{bmatrix} \mathcal{A}^{(1)}(X) \\ \mathcal{A}^{(2)}(X) \end{bmatrix}$ for $\mathcal{A}^{(1)}$ and $\mathcal{A}^{(2)}$ given by

(C.3a)
$$\mathcal{A}^{(1)} : \mathrm{S}^{dn+m \times dn+m} \to \mathbb{R}^m; \quad \mathcal{A}^{(1)}(X) = [\mathrm{Tr}(A_1^{(1)}X) \ldots \mathrm{Tr}(A_m^{(1)}X)]^T,$$

(C.3b)
$$\mathcal{A}^{(2)} : \mathrm{S}^{dn+m \times dn+m} \to \mathbb{R}^{M/2}; \quad \mathcal{A}^{(2)}(X) = [\mathrm{Tr}(A_1^{(2)}X) \ldots \mathrm{Tr}(A_{M/2}^{(2)}X)]^T.$$

Here, we let $A^{(i)}$ ($i = 1, 2$) denote the matrices satisfying $\mathcal{A}^{(i)}(X) = A^{(i)}\mathrm{vec}(X)$ that are given by

(C.4a)
$$A^{(1)} := \begin{bmatrix} \mathrm{vec}(A_1^{(1)}) & \ldots & \mathrm{vec}(A_m^{(1)}) \end{bmatrix}^T \in \mathbb{R}^{m \times (dn+m)^2},$$

(C.4b)
$$A^{(2)} := \begin{bmatrix} \mathrm{vec}(A_1^{(2)}) & \ldots & \mathrm{vec}(A_{M/2}^{(2)}) \end{bmatrix}^T \in \mathbb{R}^{M/2 \times (dn+m)^2}.$$

We also let $A := [\begin{smallmatrix} A^{(1)} \\ A^{(2)} \end{smallmatrix}]$, which satisfies $\mathcal{A}(X) = A\mathrm{vec}(X)$. Note that $A^{(1)}(A^{(2)})^T = 0$ and $A^{(2)}(A^{(2)})^T = I_{M/2}$ (after scaling $A_k^{(2)}$). Now, the dual of (C.2) is

(C.5a)
$$\underset{\mathbf{y},S}{\text{minimize}} \quad -\mathbf{b}^T\mathbf{y}$$

(C.5b)
$$\text{subject to} \quad \mathcal{A}^*(\mathbf{y}) + S = W\,,$$

(C.5c)
$$S \succeq 0\,,$$

where the operator $\mathcal{A}^* : \mathbb{R}^{m+M/2} \to \mathrm{S}^{dn+m \times dn+m}$ is the adjoint of $\mathcal{A}$ and is defined by $\mathcal{A}^*(\mathbf{y}) := \mathrm{mat}(A^T\mathbf{y})$. Hence, the augmented Lagrangian function for the dual SDP (C.5) is given by

(C.6)
$$\mathcal{L}_\mu(\mathbf{y}, S, X) = -\mathbf{b}^T\mathbf{y} + \mathrm{Tr}(X(\mathcal{A}^*(\mathbf{y}) + S - W)) + \frac{\mu}{2}\|\mathcal{A}^*(\mathbf{y}) + S - W\|_F^2\,,$$

where $X \in \mathrm{S}^{dn+m \times dn+m}$ and $\mu > 0$ is a penalty parameter. Here, we can obtain an ADM to minimize $\mathcal{L}_\mu(\mathbf{y}, S, X)$ with respect to $\mathbf{y}, S, X$ in an alternating fashion: Starting from an initial point $(\mathbf{y}^0, S^0, X^0)$, ADM sequentially solves at each iteration [62] the problems

(C.7a)
$$\mathbf{y}^{k+1} := \underset{\mathbf{y}}{\mathrm{argmin}}\; \mathcal{L}_\mu(\mathbf{y}, S^k, X^k)\,,$$

(C.7b)
$$S^{k+1} := \underset{S \succeq 0}{\mathrm{argmin}}\; \mathcal{L}_\mu(\mathbf{y}^{k+1}, S, X^k)\,,$$

and finally updates (the dual Lagrange multiplier) X by

(C.8)
$$X^{k+1} := X^k + \mu(\mathcal{A}^*(\mathbf{y}^{k+1}) + S^{k+1} - W)\,.$$

We solve (C.7a) by setting $\nabla_{\mathbf{y}}\mathcal{L}_\mu = 0$ and obtain

(C.9)
$$\mathbf{y}^{k+1} = -(\mathcal{A}\mathcal{A}^*)^{-1}\left(\frac{1}{\mu}\left(\mathcal{A}(X^k) - \mathbf{b}\right) + \mathcal{A}\left(S^k - W\right)\right)\,,$$

where the operator $\mathcal{A}\mathcal{A}^*$ satisfies $\mathcal{A}\mathcal{A}^*(\mathbf{y}) = \begin{bmatrix} A^{(1)}(A^{(1)})^T & 0 \\ 0 & I_{M/2} \end{bmatrix}\mathbf{y}$. Letting $\mathcal{B} : \mathrm{S}^{dn+m \times dn+m} \to \mathbb{R}^m$ denote the linear operator

(C.10)
$$\mathcal{B}(X) = [\mathrm{Tr}(C^1 T)\ldots\mathrm{Tr}(C^m T)]^T \quad \text{for} \quad X = \begin{bmatrix} T & \star \\ \star & \star \end{bmatrix}\,,$$

where C^k is the constraint matrix of the kth edge in E_t (given by (2.5)), we get $\mathcal{A}^{(1)}(\mathcal{A}^{(1)})^* = \mathcal{B}\mathcal{B}^* + I$ (and hence, $A^{(1)}(A^{(1)})^T = BB^T + I_m$, where B satisfies $\mathcal{B}(X) = B\mathrm{vec}(X)$), making $\mathcal{A}\mathcal{A}^*$ invertible. Also, by rearranging the terms of $\mathcal{L}_\mu(\mathbf{y}^{k+1}, S, X^k)$, (C.7b) becomes equivalent to

(C.11)
$$S^{k+1} = \underset{S \succeq 0}{\mathrm{argmin}} \left\|S - \Theta^{k+1}\right\|_F\,,$$

where $\Theta^{k+1} := W - \mathcal{A}^*(\mathbf{y}^{k+1}) - \frac{1}{\mu}X^k$. Hence, we get the solution $S^{k+1} = U_+\Sigma_+U_+^T$, where

$$(\text{C.12}) \qquad \Theta^{k+1} = \begin{bmatrix} U_+ & U_- \end{bmatrix} \begin{bmatrix} \Sigma_+ & 0 \\ 0 & \Sigma_- \end{bmatrix} \begin{bmatrix} U_+^T \\ U_-^T \end{bmatrix}$$

is the spectral decomposition of Θ^{k+1} and Σ_+ denotes the diagonal matrix of positive eigenvalues of Θ^{k+1}. Finally, using (C.8), X^{k+1} is given by

$$(\text{C.13}) \qquad X^{k+1} = -\mu U_-\Sigma_-U_-^T.$$

When implemented as is, the $\mathbf{y}$, S, and X updates of each iteration of ADM are extremely costly. This computational cost stems from the increase in the dimensions of the variables involved (after stating the primal and dual SDPs in standard form) and manifests itself via the computation of the inverse of $\mathcal{A}\mathcal{A}^*$, the $\mathbf{y}$ update (C.9), and the decomposition (C.12). However, considering the convergence analysis of ADM given in [62], since $\mathcal{A}$ is full-rank and the Slater condition (clearly) holds for the SDP (C.2) (i.e., there exists $\hat{X} \succ 0$ such that $\mathcal{A}(\hat{X}) = \mathbf{b}$), ADM converges to a primal-dual solution $(X^*, \mathbf{y}^*, S^*)$ of (C.2) and (C.5) *irrespective of the initial point* $(X^0, \mathbf{y}^0, S^0)$. Here, we make the following crucial observation: Starting from X^0 and S^0 having the form

$$(\text{C.14}) \qquad X^0 = \begin{bmatrix} T^0 & 0 \\ 0 & \text{Diag}(\nu^0) \end{bmatrix}, \quad S^0 = \begin{bmatrix} R^0 & 0 \\ 0 & \text{Diag}(\eta^0) \end{bmatrix},$$

i.e., satisfying $\mathcal{A}^{(2)}(X^0) = \mathcal{A}^{(2)}(S^0) = \mathbf{0}_{M/2}$, it follows that $\mathbf{y}^k$ becomes essentially m-dimensional (i.e., $\mathbf{y}_i^k = 0$ for $i = m+1, \ldots, m+M/2$), and X^k, S^k preserve their form (i.e., $\mathcal{A}^{(2)}(X^k) = \mathcal{A}^{(2)}(S^k) = \mathbf{0}_{M/2}$) for each $k \geq 1$. To see this, consider X^k and S^k satisfying (C.14) for some $T^k \succeq 0$, $\nu^k \geq 0$, and $R^k \succeq 0$, η^k. Then, by (C.9), we have

$$(\text{C.15a}) \qquad \mathbf{y}^{k+1} = -\begin{bmatrix} \left(A^{(1)}(A^{(1)})^T\right)^{-1} & 0 \\ 0 & I_{M/2} \end{bmatrix} \left(\frac{1}{\mu} \begin{bmatrix} \mathcal{A}^{(1)}(X^k) - \mathbf{1}_m \\ \mathbf{0}_{M/2} \end{bmatrix} + \begin{bmatrix} \mathcal{A}^{(1)}\left(S^k - W\right) \\ \mathbf{0}_{M/2} \end{bmatrix} \right)$$

$$(\text{C.15b}) \qquad = -\begin{bmatrix} \left(\tilde{\mathcal{B}}\tilde{\mathcal{B}}^* + I\right)^{-1} \left(\frac{1}{\mu}(\tilde{\mathcal{B}}(T^k) - \nu^k - \mathbf{1}_m) + \tilde{\mathcal{B}}(R^k - L) - \eta^k\right) \\ \mathbf{0}_{M/2} \end{bmatrix}$$

$$(\text{C.15c}) \qquad = \begin{bmatrix} \mathbf{z}^{k+1} \\ \mathbf{0}_{M/2} \end{bmatrix},$$

where $\tilde{\mathcal{B}}$ is the (trivial) restriction of $\mathcal{B}$ to $\text{S}^{dn \times dn}$, i.e., $\tilde{\mathcal{B}}(T) = \mathcal{B}(X)$ for $X = \begin{bmatrix} T & \star \\ \star & \star \end{bmatrix}$. For $\Theta^{k+1} = W - \mathcal{A}^*(\mathbf{y}^{k+1}) - \frac{1}{\mu}X^k$, we obtain

$$(\text{C.16a}) \qquad \Theta^{k+1} = \begin{bmatrix} L & 0 \\ 0 & 0 \end{bmatrix} - \text{mat}\left((A^{(1)})^T\mathbf{z}^{k+1}\right) - \frac{1}{\mu}\begin{bmatrix} T^k & 0 \\ 0 & \text{Diag}(\nu^{\mathbf{k}}) \end{bmatrix}$$

$$(\text{C.16b}) \qquad = \begin{bmatrix} L - \frac{1}{\mu}T^k - \tilde{\mathcal{B}}^*(\mathbf{z}^{k+1}) & 0 \\ 0 & \text{Diag}(\mathbf{z}^{k+1} - \frac{1}{\mu}\nu^{\mathbf{k}}) \end{bmatrix}.$$

Here, we define $F^{k+1} := L - \frac{1}{\mu}T^k - \tilde{\mathcal{B}}^*(\mathbf{z}^{k+1})$, and consider its spectral decomposition given by

$$F^{k+1} = \begin{bmatrix} V_+ & V_- \end{bmatrix} \begin{bmatrix} D_+ & 0 \\ 0 & D_- \end{bmatrix} \begin{bmatrix} V_+^T \\ V_-^T \end{bmatrix},$$

where D_+ denotes the diagonal matrix of positive eigenvalues of F^{k+1}. Then, using (C.11), (C.12), and (C.13), the S^{k+1} and X^{k+1} updates are given by

$$(C.17a) \qquad S^{k+1} = \begin{bmatrix} V_+ D_+ V_+^T & 0 \\ 0 & \mathrm{Diag}\left(\max\left\{\mathbf{z}^{k+1} - \frac{1}{\mu}\nu^k, \mathbf{0}_m\right\}\right) \end{bmatrix},$$

$$(C.17b) \qquad X^{k+1} = \begin{bmatrix} -\mu V_- D_- V_-^T & 0 \\ 0 & -\mu\,\mathrm{Diag}\left(\min\left\{\mathbf{z}^{k+1} - \frac{1}{\mu}\nu^k, \mathbf{0}_m\right\}\right) \end{bmatrix}.$$

As a result, we can rewrite the updates (C.7a), (C.7b), and (C.8) (or, equivalently, (C.9), (C.11), and (C.13)), in terms of the new variables $\mathbf{z}, \nu, \eta \in \mathbb{R}^m$ and $R, T \in \mathbb{R}^{dn \times dn}$, as

$$(C.18a) \qquad \mathbf{z}^{k+1} := -\left(\tilde{\mathcal{B}}\tilde{\mathcal{B}}^* + I\right)^{-1}\left(\frac{1}{\mu}(\tilde{\mathcal{B}}(T^k) - \nu^k - \mathbf{1}_m) + \tilde{\mathcal{B}}(R^k - L) - \eta^k\right),$$

$$(C.18b) \qquad R^{k+1} := V_+ D_+ V_+^T,$$

$$(C.18c) \qquad \eta^{k+1} := \max\left\{\mathbf{z}^{k+1} - \frac{1}{\mu}\nu^k, \mathbf{0}_m\right\},$$

$$(C.18d) \qquad T^{k+1} := -\mu V_- D_- V_-^T,$$

$$(C.18e) \qquad \nu^{k+1} := -\mu\min\left\{\mathbf{z}^{k+1} - \frac{1}{\mu}\nu^k, \mathbf{0}_m\right\}.$$

Hence, we are able to overcome the excess computational cost induced by rewriting (2.6) in standard form and maintain convergence. Also, the linear constraint $\mathrm{Tr}(HT) = 0$ of (2.6) can be trivially incorporated into the ADM framework in the following way: Let $\tilde{B} \in \mathbb{R}^{m \times d^2 n^2}$ denote the matrix corresponding to the linear operator $\tilde{\mathcal{B}}$, i.e., $\tilde{\mathcal{B}}(T) = \tilde{B}\mathrm{vec}(T)$, and let $\tilde{B}_l$ denote the lth row of $\tilde{B}$ given by $\tilde{B}_l = \mathrm{vec}(C^l)^T$ for $l \in E_t$. Then, since $\tilde{\mathcal{B}}^*(\mathbf{z}^{k+1}) = \mathrm{mat}(\tilde{B}^T\mathbf{z}^{k+1}) = \sum_l \mathbf{z}_l^{k+1} C^l$, we get

$$(C.19a) \qquad \mathrm{Tr}(HF^{k+1}) = \mathrm{Tr}(HL) - \frac{1}{\mu}\mathrm{Tr}(HT^k) - \sum_{l=1}^{m} \mathbf{z}_l^{k+1}\,\mathrm{Tr}(HC^l)$$

$$(C.19b) \qquad\qquad\qquad = -\frac{1}{\mu}\mathrm{Tr}(HT^k).$$

Here, if we have $\mathrm{Tr}(HT^k) = 0$, we get $\mathrm{Tr}(HF^{k+1}) = 0$, which, together with $T^k \succeq 0$, implies that the eigenvectors of H are in the nullspace of F^{k+1}. Hence, we obtain $\mathrm{Tr}(HT^{k+1}) = 0$. As a result, if we choose T^0 such that $\mathrm{Tr}(HT^0) = 0$, by induction we get $\mathrm{Tr}(HT^k) = 0$ for all k, hence for the solution T^* of ADM (as requested in (2.6)).

Moreover, we now show that the operator $\tilde{\mathcal{B}}\tilde{\mathcal{B}}^*$ has a simple structure allowing efficient computation of the inverse $(\tilde{\mathcal{B}}\tilde{\mathcal{B}}^* + I)^{-1}$.

Lemma C.1. *Let $\tilde{B} \in \mathbb{R}^{m \times d^2 n^2}$ denote the matrix corresponding to the linear operator $\tilde{\mathcal{B}}$, i.e., $\tilde{\mathcal{B}}(T) = \tilde{B}\mathrm{vec}(T)$. Then the inverse $(\tilde{\mathcal{B}}\tilde{\mathcal{B}}^* + I)^{-1}$ is given by*

$$(\text{C.20a}) \qquad \left(\tilde{\mathcal{B}}\tilde{\mathcal{B}}^* + I\right)^{-1}(\mathbf{z}) = \left(\tilde{B}\tilde{B}^T + I_m\right)^{-1}\mathbf{z}$$

$$(\text{C.20b}) \qquad = \left(V_{\tilde{B}}\left(D_{\tilde{B}} + I_n\right)^{-1}V_{\tilde{B}}^T + \frac{1}{2d+1}\left(I_m - V_{\tilde{B}}V_{\tilde{B}}^T\right)\right)\mathbf{z},$$

where $V_{\tilde{B}}$ and $D_{\tilde{B}}$ denote the matrix of eigenvectors and the diagonal matrix of eigenvalues of $\tilde{B}\tilde{B}^T$, corresponding to its n largest eigenvalues.

Proof. Let $k = (i_k, j_k)$ denote the kth edge in E_t, and let $\tilde{B}_k$ denote the kth row of $\tilde{B}$ given by $\tilde{B}_k = \mathrm{vec}(C^k)^T$, where C^k is given by (2.5). Then, the (k,l)th entry of $\tilde{B}\tilde{B}^T$ is given by

$$(\text{C.21}) \qquad \tilde{B}\tilde{B}_{k,l}^T = \begin{cases} 0, & \{(i_k,i_l),(i_k,j_l),(j_k,i_l),(j_k,j_l)\} \cap E_t = \emptyset, \\ d, & \{(i_k,i_l),(i_k,j_l),(j_k,i_l),(j_k,j_l)\} \cap E_t \neq \emptyset \ \text{and} \ k \neq l, \\ 4d, & k = l, \end{cases}$$

i.e., $\tilde{B}\tilde{B}^T = dM_G^T M_G + 2dI_m$, where $M_G \in \mathbb{R}^{n \times m}$ is the (vertex-edge) incidence matrix of G_t (see, e.g., [10]). Hence, the spectral decomposition of $\tilde{B}\tilde{B}^T$ is given by

$$(\text{C.22}) \qquad \tilde{B}\tilde{B}^T = \begin{bmatrix} V_{\tilde{B}} & U_{\tilde{B}} \end{bmatrix} \begin{bmatrix} D_{\tilde{B}} & 0 \\ 0 & 2dI_{m-n} \end{bmatrix} \begin{bmatrix} V_{\tilde{B}}^T \\ U_{\tilde{B}}^T \end{bmatrix},$$

which clearly implies the claim of the lemma. ∎

As a result, in order to compute $(\tilde{\mathcal{B}}\tilde{\mathcal{B}}^* + I)^{-1}$, we need to compute only the n largest eigenvalues and the corresponding eigenvectors of the sparse matrix $\tilde{B}\tilde{B}^T$.

Appendix D. Proof of Proposition 3.1. We prove the result in two steps: First we prove exact recovery of patch signs by (3.2) and EVM; then we prove exact recovery of global locations by (3.3) (given the noiseless signs).

Consider a pair of patches P_i and P_j, $(i,j) \in E_P$. Using (3.1), we obtain (where we tamely assume $|c^i| > 0$ for all $i \in V_P$, and $\mathbf{t}_k = \mathbf{t}_l \iff k = l$ for all $k,l \in \bigcup P_i$)

$$(\text{D.1}) \qquad \mathbf{t}_k^i = \frac{c^j}{c^i}\mathbf{t}_k^j + \frac{\mathbf{t}^j - \mathbf{t}^i}{c^i}, \quad k \in P_i \cap P_j.$$

Observe that (D.1) implies that $c^{ij} = c^j/c^i$ and $\mathbf{t}^{ij} = (\mathbf{t}^j - \mathbf{t}^i)/c^i$ comprise a minimizer of the pairwise patch registration step (3.2), with a cost equal to 0. Hence, every minimizer of (3.2) has a cost equal to 0, i.e., is a solution of the linear equations

$$(\text{D.2}) \qquad \mathbf{t}_k^i = c^{ij}\mathbf{t}_k^j + \mathbf{t}^{ij}, \quad k \in P_i \cap P_j.$$

However, since $|P_i \cap P_j| \geq 2$, c^{ij} can be uniquely determined by simply selecting two equations from (D.2), subtracting one equation from the other (to eliminate $\mathbf{t}^{ij}$) and solving for c^{ij} (note that, since we assume $\mathbf{t}_k \neq \mathbf{t}_l$ for $k \neq l$ and $|c^i| > 0$ for all $i \in V_P$, we get $\mathbf{t}_k^i \neq \mathbf{t}_l^i$ for $k \neq l$,

which allows us to solve for c^{ij}). Also, $\mathbf{t}^{ij}$ is uniquely determined by substituting the solution for c^{ij} in one of the equations in (D.2). As a result, $c^{ij} = c^j/c^i$ and $\mathbf{t}^{ij} = (\mathbf{t}^j - \mathbf{t}^i)/c^i$ is the unique solution of (3.2) for each $(i,j) \in E_P$, which yields the (unique) pairwise sign solution $z^{ij} = \mathrm{sign}(c^j/c^i) = z^i z^j$ (i.e., (3.2) recovers the pairwise signs exactly for all $(i,j) \in E_P$). Given the noiseless pairwise signs $\{z^{ij}\}_{(i,j) \in E_P}$ defined on the connected graph G_P, EVM [15] recovers the exact patch signs z^i (up to a global sign z); i.e., we obtain $\hat{z}^i = zz^i$ (for arbitrary $z \in \{-1,+1\}$).

Now we define $\beta := (\min_i |c^i|)^{-1}$, $\mathbf{x}_k := z\beta\mathbf{t}_k$, $s^i := \beta|c^i|$, and $\mathbf{u}^i := z\beta\mathbf{t}^i$ for each $i \in V_P$ and $k \in \bigcup P_i$. Here, $\{\mathbf{x}_k, s^i, \mathbf{u}^i\}$ satisfy

$$\text{(D.3)} \qquad \mathbf{x}_k - (s^i \hat{z}^i \mathbf{t}_k^i + \mathbf{u}^i) = z\beta(\mathbf{t}_k - (c^i \mathbf{t}_k^i + \mathbf{t}^i)) = \mathbf{0} \quad \forall k \in P_i,\ i \in V_P\,,$$

where the last equality follows from (3.1). We note that (D.3), together with the feasibility $s^i \geq 1$, implies that $\{\mathbf{x}_k, s^i, \mathbf{u}^i\}_{k \in P_i, i \in V_P}$ is a minimizer of (3.3), with a cost equal to 0. We now show that, for any minimizer $\{\mathbf{y}_k, r^i, \mathbf{w}^i\}_{k \in P_i, i \in V_P}$ of (3.3), $\{\mathbf{y}_k\}_{k \in \bigcup P_i}$ is congruent to $\{\mathbf{x}_k\}_{k \in \bigcup P_i}$, which is itself congruent to $\{\mathbf{t}_k\}_{k \in \bigcup P_i}$. Since $\{\mathbf{y}_k, r^i, \mathbf{w}^i\}_{k \in P_i, i \in V_P}$ (with $r^i \geq 1$ for all $i \in V_P$) must have the cost value 0, for each $i \in V_P$ we get

$$\text{(D.4)} \qquad \mathbf{t}_k^i = \frac{\mathbf{y}_k - \mathbf{w}^i}{zz^i r^i} = \frac{\mathbf{x}_k - \mathbf{u}^i}{zz^i s^i} \ \Rightarrow\ \mathbf{y}_k = \frac{r^i}{s^i}\mathbf{x}_k + \mathbf{w}^i - \frac{r^i}{s^i}\mathbf{u}^i \quad \forall k \in P_i\,.$$

Here, for each $(i,j) \in E_P$, we can find at least two separate $k \in P_i \cap P_j$ (say k_1 and k_2) for which (D.4) implies

$$\text{(D.5a)} \qquad \mathbf{y}_{k_1} - \mathbf{y}_{k_2} = \frac{r^i}{s^i}(\mathbf{x}_{k_1} - \mathbf{x}_{k_2}) = \frac{r^j}{s^j}(\mathbf{x}_{k_1} - \mathbf{x}_{k_2})$$

$$\text{(D.5b)} \qquad \Rightarrow \frac{r^i}{s^i} = c \ \forall i \in V_P \ \Rightarrow\ \mathbf{w}^i - \frac{r^i}{s^i}\mathbf{u}^i = \mathbf{t} \ \forall i \in V_P\,,$$

where, in (D.5b), the first implication follows from $r^i \geq 1$, $\mathbf{x}_{k_1} \neq \mathbf{x}_{k_2}$ (since $\mathbf{x}_k$ are congruent to $\mathbf{t}_k$), and the connectivity of G_P (i.e., since each pair results in a constant ratio of scales r^i/s^i, we can propagate this information to all the patches to conclude that this constant is the same for all patches), and also the second implication follows from substituting $r^i/s^i = c$ in (D.4) and the connectivity of G_P (again, allowing us to conclude that $\mathbf{w}^i - (r^i/s^i)\mathbf{u}^i$ is fixed for all patches). This completes the proof, since we just showed that, for any minimizer $\{\mathbf{y}_k, r^i, \mathbf{w}^i\}_{k \in P_i, i \in V_P}$ of (3.3), we have

$$\text{(D.6)} \qquad \mathbf{y}_k = z(c\beta)\mathbf{t}_k + \mathbf{t} \ ;\ z \in \{-1,+1\},\ \beta > 0,\ c \geq 1,\ \mathbf{t} \in \mathbb{R}^d\,,$$

i.e., $\{\mathbf{y}_k\}_{k \in \bigcup P_i}$ is congruent to $\{\mathbf{t}_k\}_{k \in \bigcup P_i}$. ∎

REFERENCES

[1] S. AGARWAL, N. SNAVELY, I. SIMON, S. M. SEITZ, AND R. SZELISKI, *Building Rome in a day*, in Proceedings of the 2009 IEEE 12th International Conference on Computer Vision (ICCV), 2009, pp. 72–79.

[2] E. ANDERSON, Z. BAI, C. BISCHOF, S. BLACKFORD, J. DEMMEL, J. DONGARRA, J. DU CROZ, A. GREENBAUM, S. HAMMARLING, A. MCKENNEY, AND D. SORENSEN, *LAPACK Users' Guide*, 3rd ed., SIAM, Philadelphia, 1999.

[3] M. ARIE-NACHIMSON, S. KOVALSKY, I. KEMELMACHER-SHLIZERMAN, A. SINGER, AND R. BASRI, *Global motion estimation from point matches*, in Proceedings of the 2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission (3DIMPVT '12), Zurich, IEEE Computer Society, Washington, DC, 2012, pp. 81–88.

[4] 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 Trans. Mobile Comput., 5 (2006), pp. 1663–1678.

[5] A. S. BANDEIRA, Y. KHOO, AND A. SINGER, *Open problem: Tightness of maximum likelihood semidefinite relaxations*, in Proceedings of the 27th Conference on Learning Theory (COLT), 2014, pp. 1265–1267.

[6] R. BHATIA, *Matrix Analysis*, Grad. Texts in Math. 169, Springer-Verlag, New York, 1997.

[7] P. BISWAS, T.-C. LIANG, K.-C. TOH, Y. YE, AND T.-C. WANG, *Semidefinite programming approaches for sensor network localization with noisy distance measurements*, IEEE Trans. Automat. Sci. Eng., 3 (2006), pp. 360–371.

[8] P. BISWAS, T.-C. LIANG, T.-C. WANG, AND Y. YE, *Semidefinite programming based algorithms for sensor network localization*, ACM Trans. Sen. Netw., 2 (2006), pp. 188–220.

[9] M. BRAND, M. ANTONE, AND S. TELLER, *Spectral solution of large-scale extrinsic camera calibration as a graph embedding problem*, in Computer Vision-ECCV 2004, Springer-Verlag, Berlin, Heidelberg, 2004, pp. 262–273.

[10] A. E. BROUWER AND W. H. HAEMERS, *Spectra of Graphs*, Springer, New York, 2012.

[11] E. J. CANDÈS, X. LI, Y. MA, AND J. WRIGHT, *Robust principal component analysis?*, J. ACM, 58 (2011), 11.

[12] E. J. CANDÈS AND B. RECHT, *Exact matrix completion via convex optimization*, Found. Comput. Math., 9 (2009), pp. 717–772.

[13] E. J. CANDÈS, T. STROHMER, AND V. VORONINSKI, *Exact and stable signal recovery from magnitude measurements via convex programming*, Comm. Pure Appl. Math., 66 (2013), pp. 1241–1274.

[14] K. N. CHAUDHURY, Y. KHOO, AND A. SINGER, *Global registration of multiple point clouds using semidefinite programming*, SIAM J. Optim., 25 (2015), pp. 468–501.

[15] M. CUCURINGU, Y. LIPMAN, AND A. SINGER, *Sensor network localization by eigenvector synchronization over the Euclidean group*, ACM Trans. Sen. Netw., 8 (2012), 19.

[16] T. EREN, W. WHITELEY, AND P. N. BELHUMEUR, *Using angle of arrival (bearing) information in network localization*, in Proceedings of the 45th IEEE Conference on Decision and Control, San Diego, CA, 2006, pp. 4676–4681.

[17] 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 Proceedings of the 42nd IEEE Conference on Decision and Control, Maui, HI, 2003, pp. 3064–3069.

[18] Y. FURUKAWA, B. CURLESS, S. M. SEITZ, AND R. SZELISKI, *Towards Internet-scale multi-view stereo*, in Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 1434–1441.

[19] Y. FURUKAWA AND J. PONCE, *Accurate, dense, and robust multiview stereopsis*, IEEE Trans. Pattern Anal. Mach. Intell., 32 (2010), pp. 1362–1376.

[20] M. X. GOEMANS AND D. P. WILLIAMSON, *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming*, J. ACM, 42 (1995), pp. 1115–1145.

[21] S. J. GORTLER, C. GOTSMAN, L. LIU, AND D. THURSTON, *On affine rigidity*, J. Comput. Geom., 4 (2013), pp. 160–181.

[22] V. M. GOVINDU, *Combining two-view constraints for motion estimation*, in Proceedings of the 2001 IEEE

Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Vol. 2, 2001, pp. 218–225.

[23] V. M. GOVINDU, *Lie-algebraic averaging for globally consistent motion estimation*, in Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Vol. 1, 2004, pp. 684–691.

[24] J. E. GRAVER, B. SERVATIUS, AND H. SERVATIUS, *Combinatorial Rigidity*, Grad. Stud. Math. 2, AMS, Providence, RI, 1993.

[25] R. HARTLEY, K. AFTAB, AND J. TRUMPF, L_1 *rotation averaging using the Weiszfeld algorithm*, in Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011, pp. 3041–3048.

[26] R. HARTLEY AND A. ZISSERMAN, *Multiple View Geometry in Computer Vision*, Cambridge University Press, Cambridge, UK, 2000.

[27] M. HAVLENA, A. TORII, J. KNOPP, AND T. PAJDLA, *Randomized structure from motion based on atomic 3D models from camera triplets*, in Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009, pp. 2874–2881.

[28] B. HE AND X. YUAN, *On the $O(1/n)$ convergence rate of the Douglas–Rachford alternating direction method*, SIAM J. Numer. Anal., 50 (2012), pp. 700–709.

[29] B. HENDRICKSON, *Conditions for unique graph realizations*, SIAM J. Comput., 21 (1992), pp. 65–84.

[30] B. HENDRICKSON, *The molecule problem: Exploiting structure in global optimization*, SIAM J. Optim., 5 (1995), pp. 835–857.

[31] B. JACKSON AND T. JORDÁN, *Graph theoretic techniques in the analysis of uniquely localizable sensor networks*, in Localization Algorithms and Strategies for Wireless Sensor Networks: Monitoring and Surveillance Techniques for Target Tracking, G. Mao and B. Fidan, eds., IGI Global, Hershey, PA, 2009, pp. 146–173.

[32] D. JACOBS AND B. HENDRICKSON, *An algorithm for two dimensional rigidity percolation: The pebble game*, J. Comput. Phys., 137 (1997), pp. 346–365.

[33] F. KAHL AND R. I. HARTLEY, *Multiple-view geometry under the L_∞ norm*, IEEE Trans. Pattern Anal. Mach. Intell., 30 (2008), pp. 1603–1617.

[34] B. KATZ, M. GAERTLER, AND D. WAGNER, *Maximum rigid components as means for direction-based localization in sensor networks*, in SOFSEM 2007: Theory and Practice of Computer Science, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 330–341.

[35] R. KENNEDY, K. DANIILIDIS, O. NARODITSKY, AND C. J. TAYLOR, *Identifying maximal rigid components in bearing-based localization*, in Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, Portugal, 2012, pp. 194–201.

[36] R. KENNEDY AND C. J. TAYLOR, *Network localization from relative bearing measurements*, in Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014), 2014, pp. 149–156.

[37] G. LERMAN, M. B. MCCOY, J. A. TROPP, AND T. ZHANG, *Robust computation of linear models by convex relaxation*, Found. Comput. Math., 15 (2015), pp. 363–410.

[38] N.-H. Z. LEUNG AND K.-C. TOH, *An SDP-based divide-and-conquer algorithm for large-scale noisy anchor-free graph realization*, SIAM J. Sci. Comput., 31 (2009), pp. 4351–4372.

[39] H. LI, *Multi-view structure computation without explicitly estimating motion*, in Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, 2010, pp. 2777–2784.

[40] D. G. LOWE, *Distinctive image features from scale-invariant keypoints*, Int. J. Comput. Vis., 60 (2004), pp. 91–110.

[41] D. MARTINEC AND T. PAJDLA, *Robust rotation and translation estimation in multiview reconstruction*, in Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2007.

[42] A. MOBASHER, M. TAHERZADEH, R. SOTIROV, AND A. K. KHANDANI, *A near-maximum-likelihood decoding algorithm for MIMO systems based on semi-definite programming*, IEEE Trans. Inform. Theory, 53 (2007), pp. 3869–3886.

[43] A. Y. NG, M. I. JORDAN, AND Y. WEISS, *On spectral clustering: Analysis and an algorithm*, in Advances in Neural Information Processing Systems, 14, MIT Press, Cambridge, MA, 2002, pp. 849–856.

[44] C. OLSSON, A. P. ERIKSSON, AND F. KAHL, *Efficient optimization for L_∞-problems using pseudoconvexity*, in Proceedings of the 11th IEEE International Conference on Computer Vision (ICCV), 2007.

[45] O. ÖZYEŞIL AND A. SINGER, *Robust camera location estimation by convex programming*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015), to appear; available online at http://arxiv.org/abs/1412.0165.

[46] G. PATAKI, *On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues*, Math. Oper. Res., 23 (1998), pp. 339–358.

[47] I. J. SCHOENBERG, *Metric spaces and completely monotone functions*, Ann. of Math. (2), 39 (1938), pp. 811–841.

[48] B. SERVATIUS AND W. WHITELEY, *Constraining plane configurations in computer-aided design: Combinatorics of directions and lengths*, SIAM J. Discrete Math., 12 (1999), pp. 136–153.

[49] K. SIM AND R. HARTLEY, *Recovering camera motion using L_∞ minimization*, in Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Vol. 1, 2006, pp. 1230–1237.

[50] S. N. SINHA, D. STEEDLY, AND R. SZELISKI, *A multi-stage linear approach to structure from motion*, in Trends and Topics in Computer Vision, Springer-Verlag, Berlin, Heidelberg, 2012, pp. 267–281.

[51] N. SNAVELY, S. M. SEITZ, AND R. SZELISKI, *Photo tourism: Exploring photo collections in* 3D, ACM Trans. Graphics, 25 (2006), pp. 835–846.

[52] N. SNAVELY, S. M. SEITZ, AND R. SZELISKI, *Skeletal graphs for efficient structure from motion*, in Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2008.

[53] A. M.-C. SO AND Y. YE, *Theory of semidefinite programming for sensor network localization*, Math. Program., 109 (2007), pp. 367–384.

[54] 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 Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2008.

[55] B. TRIGGS, P. F. MCLAUCHLAN, R. I. HARTLEY, AND A. W. FITZGIBBON, *Bundle adjustment—A modern synthesis*, in Vision Algorithms: Theory and Practice, Lecture Notes in Comput. Sci. 1883, Springer-Verlag, Berlin, Heidelberg, 2000, pp. 298–372.

[56] 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, 2009, pp. 901–908.

[57] M. TUBAISHAT AND S. MADRIA, *Sensor networks: An overview*, IEEE Potentials, 22 (2003), pp. 20–23.

[58] R. H. TÜTÜNCÜ, K. C. TOH, AND M. J. TODD, *Solving semidefinite-quadratic-linear programs using SDPT*3, Math. Program., 95 (2003), pp. 189–217.

[59] D. E. TYLER, *A distribution-free M-estimator of multivariate scatter*, Ann. Statist., 15 (1987), pp. 234–251.

[60] L. VANDENBERGHE AND S. BOYD, *Semidefinite programming*, SIAM Rev., 38 (1996), pp. 49–95.

[61] L. WANG AND A. SINGER, *Exact and stable recovery of rotations for robust synchronization*, Inf. Inference, 2 (2013), pp. 145–193.

[62] Z. WEN, D. GOLDFARB, AND W. YIN, *Alternating direction augmented Lagrangian methods for semidefinite programming*, Math. Program. Comput., 2 (2010), pp. 203–230.

[63] W. WHITELEY, *A matroid on hypergraphs, with applications to scene analysis and geometry*, Discrete Comput. Geom., 4 (1989), pp. 75–95.

[64] W. WHITELEY, *Some matroids from discrete geometry*, in Matroid Theory, Contemp. Math. 197, J. G. Oxley, J. E. Bonin, and B. Servatius, eds., American Mathematical Society, Providence, RI, 1996, pp. 171–311.

[65] A. WIESEL, Y. C. ELDAR, AND S. SHAMAI, *Semidefinite relaxation for detection of* 16*-QAM signaling in MIMO channels*, IEEE Signal Process. Lett., 12 (2005), pp. 653–656.

[66] C. WU, *VisualSFM: A Visual Structure from Motion System*, http://homes.cs.washington.edu/~ccwu/vsfm/ (2011).

[67] C. WU, S. AGARWAL, B. CURLESS, AND S. M. SEITZ, *Multicore bundle adjustment*, in Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011, pp. 3057–3064.

[68] H. XU, C. CARAMANIS, AND S. SANGHAVI, *Robust PCA via outlier pursuit*, IEEE Trans. Inform. Theory, 58 (2012), pp. 3047–3067.

[69] H. ZHA AND Z. ZHANG, *Spectral properties of the alignment matrices in manifold learning*, SIAM Rev.,
    51 (2009), pp. 545–566.
[70] Z. ZHANG AND Y. SHAN, *Incremental Motion Estimation through Local Bundle Adjustment*, Technical
    report MSR-TR-01-54, Microsoft Research, Redmond, WA, 2001.
[71] L. ZHI-QUAN, M. WING-KIN, A. M.-C. SO, Y. YE, AND S. ZHANG, *Semidefinite relaxation of quadratic
    optimization problems*, IEEE Signal Processing Mag., 27 (2010), pp. 20–34.