A New Rank Constraint on Multi-view Fundamental Matrices, and its Application to Camera Location Recovery

Soumyadip Sengupta[1], Tal Amir[2], Meirav Galun[2], Tom Goldstein[1], David W. Jacobs[1], Amit Singer[3], and Ronen Basri[2]

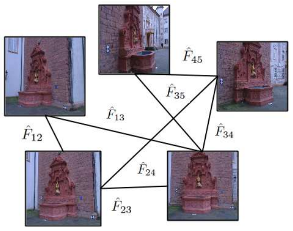
[1]University of Maryland, College Park, [2]Weizmann Institute of Science, [3]Princeton University.

Abstract

Accurate estimation of camera matrices is an important step in structure from motion algorithms. In this paper we introduce a novel rank constraint on collections of fundamental matrices in multi-view settings. We show that in general, with the selection of proper scale factors, a matrix formed by stacking fundamental matrices between pairs of images has rank 6. Moreover, this matrix forms the symmetric part of a rank 3 matrix whose factors relate directly to the corresponding camera matrices. We use this new characterization to produce better estimations of fundamental matrices by optimizing an L1-cost function using Iterative Re-weighted Least Squares and Alternate Direction Method of Multiplier. We further show that this procedure can improve the recovery of camera locations, particularly in multi-view settings in which fewer images are available.

1. Introduction

Accurate reconstruction of 3D scenes from multiview stereo images is one of the primary goals of computer vision. Current techniques use point correspondences to estimate either the essential or fundamental matrices between pairs of images, and then use the estimated matrices to recover the camera matrices and structure. Notable success was achieved when sequential methods were introduced [1, 21]. These methods first recover camera matrices and structure from two images. Then, adding one image at a time, they apply bundle adjustment to estimate the camera matrix (and structure) of the new image. Recent work attempts to further improve recovery by simultaneously considering subsets of images and recovering camera matrices that are consistent over each entire subset. In-addition a number of papers have focused on the consistent recovery of either camera orientation or location [2, 20, 19, 25, 26, 17].



$$\underbrace{\begin{bmatrix} \mathbf{0} & \hat{F}_{12} & \hat{F}_{13} & - & - \\ \hat{F}_{21} & \mathbf{0} & \hat{F}_{23} & \hat{F}_{24} & - \\ \hat{F}_{31} & \hat{F}_{32} & \mathbf{0} & \hat{F}_{34} & \hat{F}_{35} \\ - & \hat{F}_{42} & \hat{F}_{43} & \mathbf{0} & \hat{F}_{45} \\ - & - & \hat{F}_{53} & \hat{F}_{54} & \mathbf{0} \end{bmatrix}}_{\hat{F}} \approx \underbrace{\begin{bmatrix} \mathbf{0} & \lambda_{12} & \lambda_{13} & \lambda_{14} & \lambda_{15} \\ \lambda_{21} & \mathbf{0} & \lambda_{23} & \lambda_{24} & \lambda_{25} \\ \lambda_{31} & \lambda_{32} & \mathbf{0} & \lambda_{34} & \lambda_{35} \\ \lambda_{41} & \lambda_{42} & \lambda_{43} & \mathbf{0} & \lambda_{45} \\ \lambda_{51} & \lambda_{52} & \lambda_{53} & \lambda_{54} & \mathbf{0} \end{bmatrix}}_{\Lambda} \odot \underbrace{\begin{bmatrix} \mathbf{0} & F_{12} & F_{13} & F_{14} & F_{15} \\ F_{21} & \mathbf{0} & F_{23} & F_{24} & F_{25} \\ F_{31} & F_{32} & \mathbf{0} & F_{34} & F_{35} \\ F_{41} & F_{42} & F_{43} & \mathbf{0} & F_{45} \\ F_{51} & F_{52} & F_{53} & F_{54} & \mathbf{0} \end{bmatrix}}_{F}$$

$$\underline{\text{with } F = A + A^T \text{ and } \operatorname{rank}(A) = 3}.$$

Figure 1: Illustration of our rank constraint. Collections of fundamental matrices $\{\hat{F}_{ij}\}$ estimated for pairs of images (top) are arranged in a matrix $\hat{F}$ (bottom). This matrix should be equal (up to noise) to a matrix F or properly scaled collection of fundamental matrices, which in turn forms the symmetric part of a rank 3 matrix A.

This paper introduces new constraints to enable the consistent recovery of fundamental and essential matrices. This is potentially advantageous since those matrices capture simultaneously the location and orientation of the cameras, along (in the case of fundamental matrices) with their internal calibration parameters. For configurations of cameras that are not all collinear, our main result establishes that, when scaled properly, the matrix formed by appending all pairwise fundamental matrices in a multiview setting is of rank 6. More tightly, this matrix forms the symmetric part of a rank 3 matrix whose factors relate directly to the entries of the corresponding camera matrices. We further show that collinear cameras yield a matrix of rank 4 or less.

We use this characterization to develop an optimization formulation for estimating consistent sets of fundamental matrices. Our formulation can accept sets of estimated fundamental matrices in which some are noisy, some are outliers, and some cannot be estimated at all from image pairs (i.e., missing data). In solving this optimization we seek a set of scaled fundamental matrices that satisfy our constraints and fit the estimated fundamental matrices. Our formulation uses an L1 cost function, which is optimized with Iterative Re-weighted Least Squares (IRLS) [12], to remove outliers, and uses Alternate Direction Method of Multipliers (ADMM) [4] to incorporate rank constraints.

Our work is related to a variety of approaches to structure from motion (SfM) that utilize rank constraints. Tomasi and Kanade [23] showed that under an orthographic projection, and after centering, projected points form a rank 3 matrix. Sturm and Triggs [22, 24] extended this to perspective projection by showing that projected points, when scaled properly, form a rank 4 matrix. Unlike their work, which uses rank constraints on tracks of points in images, our work only considers fundamental matrices, and so in multiview settings it gives rise to systems with many fewer variables. Our approach, which seeks to recover a consistent set of fundamental matrices, is analogous to rotation or translation averaging and to loop closure [10, 6, 7]. In fact, obtaining consistent fundamental matrices can be regarded as simultaneous averaging of rotation, translation and camera calibration and as a way to close all loops. Our experiments indicate that such joint averaging performs better than a separate averaging of rotation and translation. [14] developed algebraic constraints that can be used to prove that, for cameras in general positions, certain graph configurations of fundamental matrices consistently predict the remaining fundamentals.

A number of algorithms have recently been proposed for solving unconstrained, low rank systems with outliers and missing data (e.g., [5, 13, 18]) with remarkable success. Extending such techniques to incorporate SfM constraints is an important next step.

When thousands of images are available, existing methods that use pairwise epipolar constraints or tri-focal tensors can exploit highly over-determined systems to handle noise and outliers quite accurately. However, when fewer images are available the importance of rank constraints grows, and their introduction can potentially yield more accurate estimation of camera parameters. Indeed, we provide experiments that show that using our characterization, essential matrices can be estimated more accurately than with current state-of-the-art methods, and these in turn can be translated to better estimates of camera locations.

2. Low-Rank Characterization of Fundamental Matrices in Multiview Settings

2.1. Background

We first introduce notations and give a short summary of the relevant concepts in multi-view geometry. An extensive discussion of this topic can be found in [11]. Let $I_1, ..., I_n$ denote a collection of n images of a scene and let $\mathbf{t}_i \in \mathbb{R}^3$ and $R_i \in SO(3)$ denote the location and orientation of the i'th camera in a global coordinate system. Let the 3×3 K_i denote the intrinsic camera calibration matrix for I_i. K_i is nonsingular and is typically specified in the form $K_i = \begin{bmatrix} f_x & \alpha & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}$, where f_x and f_y respectively are the focal lengths in the x and y direction, (u_0, v_0) form the principal point and α represents the skew coefficient. Let $P = (X, Y, Z)^T$ be a scene point in the global coordinate system. Its projection onto I_i (expressed in homogeneous coordinates) is given by $\mathbf{p}_i = P_i / Z_i$, where $P_i = (X_i, Y_i, Z_i)^T = K_i R_i^T (P - \mathbf{t}_i)$. We therefore associate with I_i the 3×4 camera matrix $C_i = K_i R_i^T \left[I, -\mathbf{t}_i \right]$, where I is a 3×3 identity matrix, noting that scaling C_i does not affect projection.

Next, we consider the relations between pairs of images, I_i and I_j. We can express the camera rotation and translation relating two images by $R_{ij} = R_i^T R_j$ and $\mathbf{t}_{ij} = R_i^T(\mathbf{t}_i - \mathbf{t}_j)$. Clearly, $R_{ji} = R_{ij}^T$ and $\mathbf{t}_{ji} = -R_{ij}^T \mathbf{t}_{ij}$. Two images are further related by epipolar line constraints, which are expressed by $\mathbf{p}_i^T F_{ij} \mathbf{p}_j = 0$, where F_{ij} denotes the fundamental matrix relating I_i to I_j. F_{ij} can be estimated up to scale from point correspondences. F_{ij} is related to the rotation and translation between I_i and I_j and to their respective calibration matrices by $F_{ij} = K_i^{-T} [\mathbf{t}_{ij}]_\times R_{ij} K_j^{-1}$, where $[\mathbf{t}_{ij}]_\times$ denotes the skew-symmetric matrix corresponding to cross-product with $\mathbf{t}_{ij}$. In cases in which the cameras are calibrated we set $K_i = K_j = I$ and replace the fundamental matrix with the essential matrix $E_{ij} = [\mathbf{t}_{ij}]_\times R_{ij}$. Therefore, $F_{ij} = K_i^{-T} E_{ij} K_j^{-1}$.

To derive our rank constraint we will need to express the essential and fundamental matrices relative to a global coordinate system. [27] derived an expression in terms of the camera matrices C_i and C_j. Here we will use the more recent derivation of [2] that, as we shall see below, is amenable to factorization:

$$E_{ij} = R_i^T (T_i - T_j) R_j, \tag{1}$$

$$F_{ij} = K_i^{-T} R_i^T (T_i - T_j) R_j K_j^{-1}, \tag{2}$$

where $T_i = [\mathbf{t}_i]_\times$.

2.2. Low-rank Construction

We next introduce our main result, which includes a low rank characterization of the collection of fundamental matrices in multiview settings. For our result we will construct a matrix of size $3n \times 3n$, denoted F, in which each of the 3×3 blocks includes a fundamental matrix F_{ij} (see Figure 1), where we assume that each of the pairwise fundamental matrices in F is scaled properly. We further define

$F_{ii} = 0$ for all $1 \leq i \leq n$, and note that this is consistent with (2). Likewise we define the $3n \times 3n$ matrix E from the essential matrices E_{ij}. We refer to F (resp. E) as the *multiview matrix of fundamentals (essentials)*.

Claim 1: F (and likewise E) is symmetric and $\text{rank}(F) \leq 6$. Moreover,

1. If F is produced by n cameras whose centers are not all collinear then $\text{rank}(F) = 6$ and there exists a $3n \times 3n$ matrix A with $\text{rank}(A) = 3$ such that $F = A + A^T$.

2. If F is produced by n cameras whose centers are all collinear then $\text{rank}(F) \leq 4$ and there exists a matrix A with $\text{rank}(A) \leq 2$ such that $F = A + A^T$.

Proof: To prove the claim we begin by defining the matrix A as follows. Let $U_i = K_i^{-T} R_i^T T_i$, $V_i = K_i^{-T} R_i^T$, and $A_{ij} = U_i V_j^T$. U_i, V_i, and A_{ij} are 3×3 matrices. Observing (2) and recalling that T_i is skew-symmetric we see that $F_{ij} = A_{ij} + A_{ji}^T$.

Next we construct the $3n \times 3$ matrices U and V as : $U = \begin{bmatrix} U_1 \\ \vdots \\ U_n \end{bmatrix}$ and $V = \begin{bmatrix} V_1 \\ \vdots \\ V_n \end{bmatrix}$ and set $A = UV^T$. Clearly, by construction, $\text{rank}(A) \leq 3$. Moreover, $F = A + A^T$, and so F is symmetric and $\text{rank}(F) \leq 6$.

<u>**Case 1**</u>: We show next that unless the cameras are all collinear $\text{rank}(A) = 3$. Clearly $\text{rank}(V) = 3$. Therefore we need to show that also $\text{rank}(U) = 3$. We prove this by contradiction. Assume $\text{rank}(U) < 3$. Then $\exists\, \mathbf{t} \in \mathbb{R}^3$, $\mathbf{t} \neq \mathbf{0}$, s.t. $U\mathbf{t} = \mathbf{0}$. This implies that $\mathbf{t}_i \times \mathbf{t} = \mathbf{0}$ for all $1 \leq i \leq n$. Thus, all the $\mathbf{t}_i$'s are parallel to $\mathbf{t}$, violating our assumption that not all camera locations are collinear. Consequently $\text{rank}(U) = 3$ and therefore also $\text{rank}(A) = 3$.

Next we show that when the cameras are not all collinear $\text{rank}(F) = 6$. We recall that $F_{ij} = K_i^{-T} E_{ij} K_j^{-1}$ where K_i and K_j are non-singular. We can therefore write $F = K^T E K$ where the $3n \times 3n$ matrix K is block diagonal with blocks formed by $\{K_i^{-1}\}_{i=1}^n$ and so has full rank. This implies that $\text{rank}(F) = \text{rank}(E)$, and so we are left to show that $\text{rank}(E) = 6$.

We assume WLOG that the camera locations are centered at the origin, i.e., $\sum_{i=1}^n \mathbf{t}_i = 0$ (since E is invariant to global translation of the cameras). We further argue that each column of U is orthogonal to each column of V. This is evident from the following identities

$$V^T U = \sum_{i=1}^{n} V_i^T U_i = \sum_{i=1}^{n} T_i = \left[\sum_{i=1}^{n} \mathbf{t}_i \right]_\times = 0_{3\times 3}. \quad (3)$$

Let $\tilde{A}$ denote the matrix A where we substitute $K_i = I, \forall i$ (so that $E = \tilde{A} + \tilde{A}^T$.) Denote by $\tilde{A} = \hat{U}\Sigma\hat{V}^T$ the SVD of $\tilde{A}$ ($\hat{U}$ and $\hat{V}$ are $3n \times 3$ and Σ is 3×3). Since $\tilde{A} = UV^T$ we have that $\text{span}(U) = \text{span}(\hat{U})$ and $\text{span}(V) = \text{span}(\hat{V})$. Now we can decompose E as :

$$E = \tilde{A} + \tilde{A}^T = \hat{U}\Sigma\hat{V}^T + \hat{V}\Sigma\hat{U}^T = [\, \hat{U} \;\; \hat{V} \,][\begin{smallmatrix} \Sigma & \\ & \Sigma \end{smallmatrix}] \begin{bmatrix} \hat{V}^T \\ \hat{U}^T \end{bmatrix} \quad (4)$$

Since the columns of U are orthogonal to those of V, the matrix $\begin{bmatrix} \hat{U} & \hat{V} \end{bmatrix}$ is column orthogonal. Thus, (4) is the SVD of E. And since $\tilde{A}$ is rank 3, Σ is full rank. Consequently, $\text{rank}(F) = \text{rank}(E) = 6$.

<u>**Case 2**</u>: Suppose all camera centers are collinear. WLOG assume that the origin of the global coordinate system is also collinear with the n cameras (since F is unaffected by global translation), and so we can write $\mathbf{t}_i = \alpha_i \mathbf{t}$ for $1 \leq i \leq n$ where $\alpha_i \in \mathbb{R}$ and $\mathbf{t} \in \mathbb{R}^3$. Let $T = [\mathbf{t}]_\times$, then clearly $U_i = \alpha_i K_i^{-T} R_i^T T$. Define $\tilde{U}_i = \alpha_i K_i^{-T} R_i^T$ (so $U_i = \tilde{U}_i T$) and let the $3n \times 3$ matrix $\tilde{U}$ be formed by stacking $U_1, U2, ...$ on top of each other. Then

$$A = UV^T = \tilde{U} T V^T.$$

Since T is skew-symmetric its rank is at most 2 and so is $\text{rank}(A)$. It follows that $\text{rank}(F) \leq 4$. ■

2.3. Tightness of our constraints

Claim 1 provides two constraints on the $3n \times 3n$ matrix F : (1) $F = A + A^T$ and $\text{rank}(A) = 3$. (2) The diagonal block of F vanishes, i.e., $F_{ii} = 0$.

We now investigate how tight these constraints are . We show that the number of degrees of freedom allowed by these constraints is equal to the number of degrees of freedom in the camera matrices. However, we find that there exist matrices that are allowed by these constraints, but do not produce valid fundamental matrices.

Counting arguments show that our constraints allow $12n - 15$ degrees of freedom (DOFs) in defining F. Specifically, since A is rank 3 it can be written as $A = UV^T$ where U and V are $3n \times 3$, so together they have $18n$ entries. The constraint $F = A + A^T$, however, gives rise to a 15 DOF ambiguity that should be subtracted from the number of entries of U and V, as we explain in the next paragraph. The constraint that $F_{ii} = 0$ requires $U_i V_i^T$ to be skew symmetric, yielding $6n$ more constraints on the entries of U and V, yielding together $12n - 15$ DOFs.

To calculate the DOFs in the ambiguity of $F = A + A^T$ note that we can write F as $F = [U, V] J [U, V]^T$, where J is a 6×6 permutation matrix defined as $J = [\begin{smallmatrix} \mathbf{0} & I \\ I & \mathbf{0} \end{smallmatrix}]$ (so $J[U, V]^T = [V, U]^T$). With this notation the ambiguity in factorizing F is obtained by introducing a 6×6 matrix Q such that $QJQ^T = J$ so that $[U, V] QJQ^T [U, V]^T = [U, V] J [U, V]^T = F$. Q has 36 entries, but the constraints $QJQ^T = J$ reduce its degrees of freedom to 15. Denote $Q = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix}$ these constraints restrict the products $Q_{11} Q_{12}$ and $Q_{21} Q_{22}$ to be skew symmetric and the sum

$Q_{11}Q_{22}+Q_{12}Q_{21}=I$, providing altogether 21 constraints on the 36 entries of Q, leaving 15 DOFs.

The number of DOFs in factoring F is equal to the DOFs in defining n cameras. In general, the number of DOFs in defining n perspective cameras is $11n-15$. However, each camera matrix can be scaled arbitrarily and each choice of scale will (inversely) scale the respective row and column of F. In other words, n camera matrices, $C_1, ..., C_n$, scaled arbitrarily by non zeros $1/s_1, ..., 1/s_n$, produce a collection of equivalent multiview fundamental matrices defined by SFS with $S = \text{diag}\{s_1, , s_2, ..., s_n\}, s_i \neq 0$. The freedom in choosing the entries of S accounts for the n missing DOFs.

We note however that although the DOFs in factoring F with our constraints are equal to the DOFs in defining n camera matrices there exist matrices that satisfy our constraints but cannot be realized with n cameras. Specifically, these constraints do not guarantee that all the pairwise fundamental matrices F_{ij} are rank deficient. The constraint $F_{ii} = 0$ restricts $U_i V_i^T$ to be skew-symmetric, implying that either U_i or V_i is rank deficient. If all U_i's (or equivalently all V_i's) are chosen to be rank deficient then so are all the F_{ij}. If however some of the U_i's and some of the V_i's are chosen to be full rank then they may produce F_{ij} blocks that are rank 3 and so they are not legal fundamental matrices. Note that the skew-symmetry of $U_i V_i^T$ guarantees that no more than 1/4 of the F_{ij}'s can be of full rank. Indeed, our experiments (in Section 4) often produce F_{ij}'s that are near rank 2; in a typical run the average ratio of the third to second largest singular value $\approx 7 \times 10^{-8}$, presumably because the problem is so over-constrained.

In conclusion, while our constraints provide a necessary but not sufficient conditions for consistency, counting considerations indicate that our constraints are nearly tight. Below we develop an optimization scheme that utilizes these constraints to infer the missing scale factors for collections of estimated pairwise fundamental matrices, to recover missing fundamentals and to correct noisy ones.

3. Low-rank Constrained Optimization to Recover Fundamental Matrices

In this section we formulate an optimization problem that uses the constraints derived in Section 2 to achieve a better recovery of pairwise fundamental matrices. Assume we are given a set of fundamental matrices $\hat{F}_{ij}$, where $(i,j) \in \Omega$ and Ω denotes the subset of image pairs for which fundamental matrices have been estimated. (We will further assume $(i,j) \in \Omega \implies (j,i) \in \Omega$.) We use these matrices to construct our measurement matrix $\hat{F}$ whose (i,j)'th 3×3 block contains $\hat{F}_{ij}$ if $(i,j) \in \Omega$ and is zero otherwise. Note that in the absence of errors each non-zero block is related by an unknown scale factor λ_{ij} to the corresponding block in the sought multiview matrix of fundamentals F, where λ_{ij} depends on the distance between the i'th and j'th cameras. Recovering these scale factors is essential in order to apply our constraints. Our task therefore can be expressed as:

$$\min_{F,\{\lambda_{ij}\}} \sum_{(i,j)\in\Omega} \|\hat{F}_{ij} - \lambda_{ij} F_{ij}\|_F, \tag{5}$$

where F is constrained to fulfill the constraints in Claim 1. Here we have chosen to minimize over the sum of Frobenius norms of each 3×3 block. Such mixed L1-L2 norm minimization is expected to be robust to outliers.

We note that the formulation (5) is bilinear in F and the scale factors. We could avoid this bilinearity by minimizing instead $\|\lambda_{ij}\hat{F}_{ij} - F_{ij}\|_F$. Such minimization, however, is subject to a zero trivial solution and so it requires an additional constraint such as $\sum_{ij} \lambda_{ij}^2 = 1$. Our experience with such a formulation is that it is quite sensitive to errors.

Expressing (5) with the constraints results in the following problem:

$$\min_{A,\{\lambda_{ij}\}} \frac{1}{2} \sum_{(i,j)\in\Omega} \|\hat{F}_{ij} - \lambda_{ij}(A_{ij} + A_{ji}^T)\|_F$$
$$\text{s.t.} \quad \text{rank}(A) = 3,\ A_{ii} + A_{ii}^T = 0,\ \lambda_{ij} = \lambda_{ji} \tag{6}$$

where A_{ij} denotes each 3×3 sub-block of A. Our solution for F then is $F = A + A^T$.

(6) introduces a number of challenges, including the mixed L1-Frobenius norms, the bilinearity, and the rank constraint. This problem is non-convex due to the latter two challenges. Below we describe how we approach these challenges with IRLS and ADMM. Our algorithm is summarized in Algorithm 1.

3.1. Handling Outliers with IRLS

We begin by addressing the mixed L1-Frobenius norm in the cost function. We approach this with Iterative Reweighted Least Squares (IRLS) [12]. IRLS converts the problem to weighted least squares where the weights are updated from one iteration to the next. At each iteration t of the IRLS we replace the cost function in (6) with

$$\min_{A,\{\lambda_{ij}\}} \frac{1}{2} \sum_{(i,j)\in\Omega} w_{ij}^t \|\hat{F}_{ij} - (A_{ij} + A_{ji}^T)\lambda_{ij}\|_F^2, \tag{7}$$

where

$$w_{ij}^t = \begin{cases} 1/\max(\delta, \|\hat{F}_{ij} - \lambda_{ij}^{t-1}(A_{ij}^{t-1} + (A_{ji}^{t-1})^T)\|_F), \\ \qquad\qquad \text{if } (i,j) \in \Omega \\ 0 \qquad\qquad \text{otherwise.} \end{cases}$$

δ is a regularization parameters (we use $\delta = 10^{-3}$).

To clarify presentation we simplify our notations as follows. Let W and Λ be $3n \times 3n$ matrices. Denoting their

3×3 sub-blocks by W_{ij} and Λ_{ij}, we set $W_{ij} = w_{ij}\mathbf{1}$ and $\Lambda_{ij} = \lambda_{ij}\mathbf{1}$, where $\mathbf{1}$ is a 3×3 matrix with all 1's. We further use the subscript WF to denote the weighted Frobenius norm, i.e., $\|\mathbf{v}\|^2_{WF} = \mathrm{trace}(\mathbf{v}^T W \mathbf{v})$ and use $\odot$ to denote element-wise product of matrices. Therefore, in each IRLS iteration we seek to solve

$$\begin{aligned} &\min_{A,\Lambda} \quad \frac{1}{2}\|\hat{F} - \Lambda \odot (A + A^T)\|^2_{WF} \qquad (8) \\ &\text{s.t.} \quad \mathrm{rank}(A) = 3,\ A_{ii} + A^T_{ii} = 0,\ \Lambda_{ij} = \lambda_{ij}\mathbf{1},\ \lambda_{ij} = \lambda_{ji}. \end{aligned}$$

3.2. Optimization using ADMM

Next, we wish to solve the non-convex optimization problem in (8), including the bilinearity and the rank constraint. To this end we will use a scaled version of Alternate Direction Method of Multiplier (ADMM) [4, 9]. We maintain a second copy of A, which we denote as B, and form the augmented Lagrangian of (8) as:

$$\begin{aligned} &\max_{\Gamma} \min_{A,B,\Lambda} \quad \frac{1}{2}\|\hat{F} - \Lambda \odot (A + A^T)\|^2_{WF} + \frac{\tau}{2}\|B - A + \Gamma\|^2_F \\ &\text{s.t. } \mathrm{rank}(B) = 3,\ A_{ii} + A^T_{ii} = 0,\ \Lambda_{ij} = \lambda_{ij}\mathbf{1},\ \lambda_{ij} = \lambda_{ji}. \\ &\qquad (9) \end{aligned}$$

The last term in this objective, $\frac{\tau}{2}\|B - A + \Gamma\|^2_F$ denotes the Lagrangian penalty; τ is a constant, and Γ is a matrix of Lagrange multipliers of the same size as A that is updated in the ADMM steps. We next describe the ADMM steps, which are applied iteratively.

Step 1: Solving for (A, Λ).
In each iteration, k, we solve the following sub-problems:

$$\begin{aligned} &\min_{A,\Lambda} \quad \frac{1}{2}\|\hat{F} - \Lambda \odot (A + A^T)\|^2_{WF} + \frac{\tau}{2}\|A - (B + \Gamma)\|^2_F \\ &\text{s.t. } A_{ii} + A^T_{ii} = 0,\ \Lambda_{ij} = \lambda_{ij}\mathbf{1},\ \lambda_{ij} = \lambda_{ji}. \qquad (10) \end{aligned}$$

Since (10) is non-convex we will solve it by alternative minimization of A and Λ

1. Optimize w.r.t. A:
   Because of the form of (10) it is useful to separate A into its symmetric and anti-symmetric parts, A_s and A_n, so that $A = \frac{1}{2}(A_s + A_n)$ with $A_s = A + A^T$ and $A_n = A - A^T$. Let $G = B + \Gamma$; G_s and G_n respectively denote its symmetric and anti-symmetric part. We can approximate (10) in terms of A_s and A_n and separately solve for them as follows:

$$\begin{aligned} A_s^{(k+1)} &= \operatorname*{argmin}_{A_s} \frac{1}{2}\|\hat{F} - \Lambda^{(k)} \odot A_s\|^2_{WF} \\ &\quad + \frac{\tau}{8}\|A_s - G_s^{(k)}\|^2_F \ \text{ s.t. } (A_s)_{ii} = 0, \qquad (11) \\ A_n^{(k+1)} &= \operatorname*{argmin}_{A_n} \frac{\tau}{8}\|A_n - G_n^{(k)}\|^2_F = G_n^{(k)}. \qquad (12) \end{aligned}$$

   To solve (11) we take the derivative w.r.t. A_s and equate to 0. Thus we update A_s according to

$$\begin{aligned} A_s^{(k+1)} &= W \odot \Lambda^{(k)} \odot \hat{F} + \frac{\tau}{4}G_s^{(k)} \qquad (13) \\ &\quad \oslash (W \odot \Lambda^{(k)} \odot \Lambda^{(k)} + \frac{\tau}{4}) \\ (A_s^{(k+1)})_{ii} &= 0 \qquad (14) \end{aligned}$$

   where $\oslash$ denotes element-wise division.

2. Optimize w.r.t. Λ: We minimize the following sub-problem

$$\begin{aligned} \Lambda^{(k+1)} &= \operatorname*{argmin}_{\Lambda} \|\hat{F} - \Lambda \odot A_s^{(k+1)}\|^2_{WF} \\ &\text{s.t. } \Lambda_{ij} = \lambda_{ij}\mathbf{1},\ \lambda_{ij} = \lambda_{ji}. \qquad (15) \end{aligned}$$

   We can solve (15) separately for each block as follows,

$$\begin{aligned} \lambda_{ij}^{(k+1)} &= \operatorname*{argmin}_{\lambda_{ij}} \|\hat{F}_{ij} - \lambda_{ij}(A_s^{(k+1)})_{ij}\|^2_{WF},\ i < j \\ &= \mathrm{trace}(\hat{F}^T_{ij}(A_s^{(k+1)})_{ij}) / \|(A_s^{(k+1)})_{ij}\|^2_F \qquad (16) \end{aligned}$$

   Note that $\lambda_{ii}^{(k+1)} = 0$, $\lambda_{ji}^{(k+1)} = \lambda_{ij}^{(k+1)}$ and $\Lambda_{ij}^{(k+1)} = \lambda_{ij}^{(k+1)}\mathbf{1}$.

Step 2: Solving for B.
This part of the ADMM deals with the rank constraint. It requires a solution to

$$B^{(k+1)} = \operatorname*{argmin}_{B} \frac{\tau}{2}\|B - A^{(k+1)} + \Gamma^{(k)}\|^2_F \text{ s.t. } \mathrm{rank}(B) = 3.$$

This is solved by:

$$B^{(k+1)} = SVP(A^{(k+1)} - \Gamma^{(k)}, 3), \qquad (17)$$

where $SVP(X, r)$ denotes the Singular Value Projection (SVP) of X into space the of rank-r matrices. To perform $SVP(X, r)$ we compute the SVD of X and keep its top r singular values and the corresponding singular vectors.

Step 3: Update of Γ**.**

$$\Gamma^{(k+1)} = \Gamma^{(k)} + (B^{(k+1)} - A^{(k+1)}). \qquad (18)$$

The three steps above form one ADMM iteration. To optimize (9), for every iteration of the IRLS we run these ADMM steps repeatedly till convergence. In experiments we observe monotonic convergence of the cost function defined in (6) with each IRLS iteration, see an example in Figure 2.

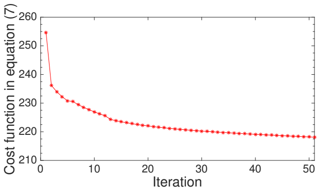


Figure 2: Convergence of our optimization algorithm.

Algorithm 1 IRLS-ADMM solver

```
Input: Estimated fundamentals in $\hat{F}$ and $\Omega$.
Output: Recovered $F$.
# IRLS: Solve (6).
Initialize $\Lambda$ and $A$.
Create weights for IRLS, $w_{ij}^0 = 1$ if $(i,j) \in \Omega$ and $w_{ij}^0 = 0$ otherwise. Set $t = 1$.
while not converged do
    # Solve (7) using ADMM formulation (9).
        Set $k = 0$, $\tau = \sum w_{ij}$, $\Gamma^0 = 0$. $B = A$.
        while not converged do
            # Alternative minimization of (10).
                Update $A$ using (12) and (14).
                Update $\Lambda$ using (16).
            Update $B$ using (17) .
            Update $\Gamma$ using (18) .
            $k = k + 1$.
        end while
    Update Weights $w_{ij}^t$ using (7).
    $t = t + 1$.
end while
$F = A + A^T$.
```

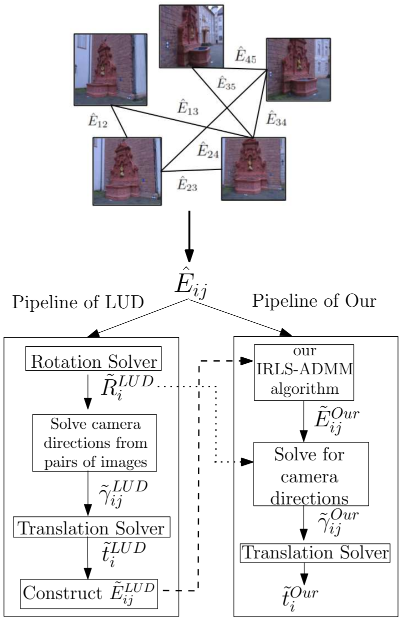


Figure 3: SfM pipelines for LUD (left) and our method (right).

4. Experiments

To demonstrate the utility of our method we tested it in the problem of estimating essential matrices and camera locations from multiple images. Current iterative and global approaches to Structure from Motion (SfM) are often tested on large datasets when many pairwise essential matrices can be estimated, achieving outstanding performance. We argue that imposing rank constraints can be useful particularly when the number of images is relatively small. To demonstrate this we run our method on subsets of images of different sizes showing improved performance relative to the existing methods particularly with smaller subsets.

In many common SfM pipelines the intrinsic calibration parameters are recovered separately. Therefore, in our main experiment below we assume that the cameras are calibrated and so we apply our optimization algorithm to essential matrices. Note that our derivations in Sections 2 and 3 hold also for essential matrices by setting $K_i = I$. Later in this section we also show the results of a smaller experiment with uncalibrated cameras.

We next describe the tested methods:

LUD [19]: Figure 3 shows the pipeline used by LUD to estimate camera locations and orientations from pairs of images. Starting from pairwise essential matrices estimated with SIFT [16] and RANSAC [3], this method first solves for camera orientations, denoted by $\tilde{R}_i^{\mathrm{LUD}}$ in Figure 3, by iteratively applying [6] while rejecting outliers. Using camera orientations it then returns to the image keypoints to estimate pairwise camera directions, denoted by $\tilde{\gamma}_{ij}^{\mathrm{LUD}}$. Using these pairwise directions it applies IRLS to solve for camera locations ($\tilde{t}_i^{\mathrm{LUD}}$), which we compare to our method. In addition, we use the estimated camera locations and orientations to reconstruct the pairwise essential matrices $\tilde{E}_{ij}^{\mathrm{LUD}}$.

ShapeKick [8]: For this method we use the same pipeline as used with LUD, except that we replace the translation recovery part of LUD with ShapeKick. ShapeKick formulates the location recovery problem as a convex optimization and solves it with ADMM. They achieved comparable performance to LUD on the dataset of [26].

1DSfM [26]: This method uses a pre-processing technique, based on projection in many random directions, to remove outliers in the original pairwise direction measurements. We use their software, which uses the pipeline described in [26] and only provides camera locations.

Our method: Figure 3 shows the pipeline used by our method. From the pairwise essential matrices we minimize (6) using the IRLS-ADMM summarized in Algorithm 1. Since our method is not convex it requires a good initialization. We initialize it with essential matrices produced by the LUD method of Ozyesil *et al.* [19], denoted $\tilde{E}_{ij}^{\mathrm{LUD}}$. Specifically $\tilde{E}_{ij}^{LUD}$ is used to initialize Λ and A in Algorithm 1. Our algorithm improves these essential matrix estimates, producing a collection of new pairwise estimates in E, denoted $\tilde{E}_{ij}^{\mathrm{Our}}$. To further produce camera locations we first use $\tilde{E}_{ij}^{\mathrm{Our}}$ and the rotations obtained by the LUD pipeline, $\tilde{R}_i^{\mathrm{LUD}}$, to solve for the pairwise camera directions $\tilde{\gamma}_{ij}^{\mathrm{Our}}$. Then we apply the translation solver of LUD to $\tilde{\gamma}_{ij}^{\mathrm{Our}}$ with $(i,j) \in \Omega$ to produce camera locations $\tilde{t}_i^{\mathrm{Our}}$. As is shown below, our improved estimates of essential matrices lead in turn to improved estimates of camera locations compared to the LUD pipeline.

We tested these methods on real image collections from [26], which come with 'ground truth' estimates of camera locations and essential matrices produced with a sequential method similar to [21]. (These ground truth estimates are used also in [26, 19, 8].) For our experiments we used 14 different scenes from the dataset. For each scene we randomly selected 5 different sub-samples of N images from the dataset. We used $N = 50$, 100, and 150 images, resulting in 70 different trials for each N. In each trial we compared the quality of the essential matrix recovered by our method to that recovered by LUD and ShapeKick. Likewise, we compared the quality of our recovered camera locations to those obtained by the three competing methods.

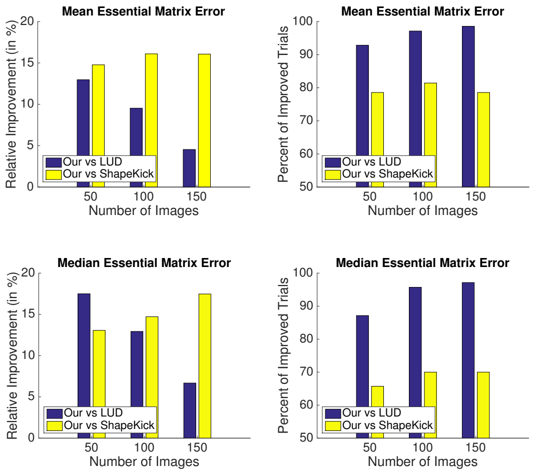


Figure 4: These graphs show a comparison of the recovery error of essential matrices achieved with our method compared to LUD (in blue) and ShapeKick (in yellow), for collections of 50, 100, and 150 images from [26], The graphs on the left show the amount of relative improvement and the ones on the right show the fraction of improved trials.

Figures 4-5 show our results. Each graph summarizes the results of 70 trials with each value of N. Figure 4 shows the quality of our essential matrix estimates compared to those obtained with LUD and ShapeKick, and Figure 5 shows the quality of our camera location estimates compared to those achieved by the three competing algorithms. We measure these as follows. In each experiment k we consider the collection of pairwise essential matrices produced by our method. We first normalize each matrix and measure its error to the respective (normalized) ground truth matrix. We then take the mean (or median) of this error over all essential matrices. Denote this error by e_k^{Our}. We then produce similar error measures for each competing algorithm, denoted e_k^{Other}. We then report:

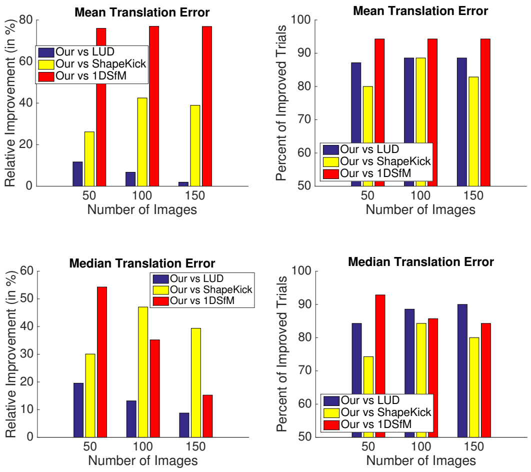


Figure 5: A comparison of the recovery error of camera locations achieved with our method compared to LUD (in blue) and ShapeKick (in yellow), and 1DSfM (in red) for collections of 50, 100, and 150 images from [26].

Relative Improvement (in %**)**: Here we report for each N and competing algorithm the average of $(e_k^{\text{Other}} - e_k^{\text{Our}})/e_k^{\text{Other}}$ over all experiments.
Percent of Improved Trials: This provides the percentage of trials in which our algorithm achieved more accurate results than a competing algorithm, i.e., $\frac{1}{K}\sum_{k=1}^{K}\mathbb{I}(e_k^{\text{Our}} < e_k^{\text{Other}})$, where $\mathbb{I}(.)$ is the indicator function and K denotes the total number of trials.

We provide similar measures to assess the quality of our camera location estimates. In Figure 6 we further show the median error of camera location estimates for all methods in all trials for $N = 50$.

It can be seen overall that our method leads to improved estimation of essential matrices and of camera locations. With 50 images, compared to, e.g., LUD, our algorithm improves the median essential matrix estimates by 17.69%. With 150 images a smaller overall improvement of 6.68% is achieved. This suggests that our constraints are more effective when smaller numbers of images are used. Interestingly, however, despite this reduction the fraction of trials in which our method achieved more accurate estimates compared to LUD in fact increased slightly from 87% with 50 images to 98% with 150 images, indicating that our method remains effective also with larger number of images (albeit yielding smaller improvement). Similar results are observed for camera location estimation. With 50 and 150 images our algorithms improves the median camera location error by 19.73% and 8.77% respectively, while the fraction of trials in which our method achieved more accurate estimates than LUD increased slightly from 84% with 50 images to 90% with 150 images.

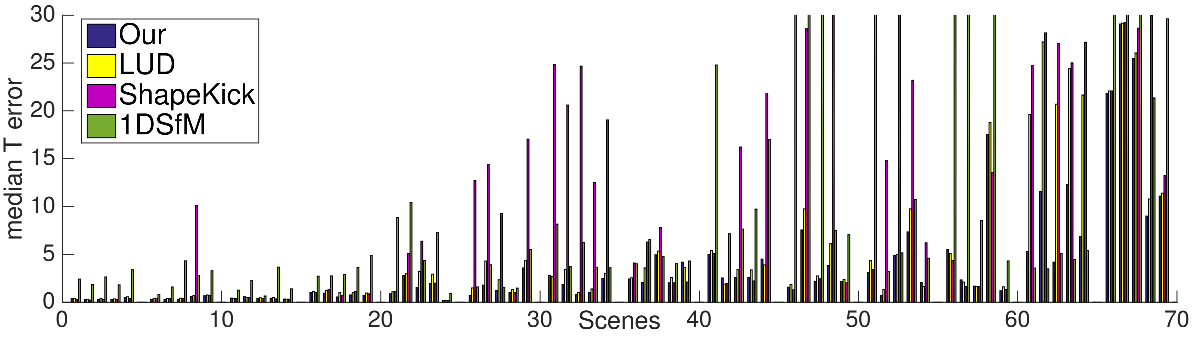


Figure 6: Median camera location error obtained by the four algorithms for 5 subsets of 50 images for 14 different scenes ('Notre Dame', 'Montreal Notre Dame', 'Alamo', 'Piazza del Popolo', 'Piccadilly', 'NYC Library', 'Yorkminster', 'Union Square', 'Madrid Metropolis', 'Tower of London', 'Vienna Cathedral', 'Roman Forum' and 'Ellis Island', 'Gendarmenmarkt'). For clarity we terminate the median T error axis at 30.

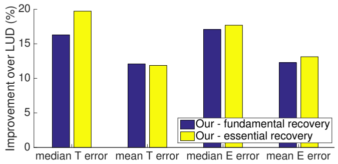


Figure 7: Improvement of our method over LUD using fundamental matrix (in blue) and essential matrix (yellow) for 50 images.

In our previous experiments we applied our optimization algorithm to essential matrices, assuming calibration is given. Below we further apply our algorithm to fundamental matrices in an uncalibrated setting. Since not all the entries of a 3 × 3 fundamental matrix are of same orders of magnitude, we normalize each of the input pairwise fundamental matrices by centering all the images and scaling their widths and heights uniformly to within the $[1, 1]$ square and then compute a normalized fundamental matrix. This does not affect our rank constraint and can be inverted at the end of the process. We tested our method on 5 subsamples of 50 images for 14 different scenes and compared it to LUD. To evaluate the quality of the recovered fundamental matrices we convert them to essential matrices by applying the known calibration matrices and further use these to recover camera locations. The results can be seen in Figure 6. Using our method to recover fundamentals (in blue) yielded comparable accuracies to our results for essential matrix recovery (yellow) and both our approaches improve significantly (10-20%) over LUD as shown in Figure 7.

We further performed bundle adjustment (using [15]) initialized by the camera parameters obtained with our method and LUD. After bundle adjustment compared to LUD our method improved camera location estimates on average by 11.52%, 3.13% and 5.43%, improving in 70.59%, 64.29% and 63.77% of all trials for 50, 100 and 150 images respectively in terms of median translation error. These results indicate that our method maintains improved accuracies over LUD also after bundle adjustment.

With 50 images the recovery of essential matrices with our method requires roughly 20 iterations of IRLS and 1000 iterations of ADMM. These take overall about 2 minutes on a 2.7 GHz Intel Core i5 computer.

To conclude, these experiments indicate that our characterization of fundamental matrices in multiview settings can be used to improve fundamental and essential matrix as well as camera location estimates. The advantage of these constraints appear to be particularly pronounced when fewer images are available.

5. Conclusion

We have introduced in this paper novel rank constraints on fundamental matrices in multiview settings. We have shown in particular that with non-collinear cameras the matrix that depicts the pairwise fundamentals is of rank 6 and forms the symmetric part of a rank 3 matrix whose factors are related directly to the entries of the respective camera matrices. We have used these constraints to develop an optimization framework to efficiently recover fundamental matrices for all pairs of images and to estimate their proper scale factors. Our experiments indicate that our method is able to provide improved estimates of essential matrices and camera locations in global SfM settings. Moreover, these experiments suggest that our constraints are particularly useful when fewer images are available.

Acknowledgment This material is based upon work supported by the National Science Foundation under grant no. IIS-1526234, by the Israel Science Foundation grant no. 1265/14 and by the Minerva Foundation with funding from the Federal German Ministry of Education and research. TG was supported by the US Office of Naval Research (N00014-17-1-2078), and the National Science Foundation (CCF-1535902). AS was supported by AFOSR FA9550-12-1-0317, Simons Investigator Award, and the Moore Foundation Data-Driven Discovery Investigator Award.

References

[1] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. Building rome in a day. In *Int. Conf. on Computer Vision*, pages 72–79. IEEE, 2009. 1

[2] M. Arie-Nachimson, S. Z. Kovalsky, I. Kemelmacher-Shlizerman, A. Singer, and R. Basri. Global motion estimation from point matches. In *Int. Conf. on 3D Imaging, Modeling, Processing, Visualization & Transmission*, pages 81–88. IEEE, 2012. 1, 2

[3] R. C. Bolles and M. A. Fischler. A ransac-based approach to model fitting and its application to finding cylinders in range data. In *IJCAI*, volume 1981, pages 637–643, 1981. 6

[4] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011. 2, 5

[5] E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009. 2

[6] A. Chatterjee and V. M. Govindu. Efficient and robust large-scale rotation averaging. In *Proc. of the IEEE Int. Conf. on Computer Vision*, pages 521–528, 2013. 2, 6

[7] Z. Cui and P. Tan. Global structure-from-motion by similarity averaging. In *Proc. of the IEEE Int. Conf. on Computer Vision*, pages 864–872, 2015. 2

[8] T. Goldstein, P. Hand, C. Lee, V. Voroninski, and S. Soatto. Shapefit and shapekick for robust, scalable structure from motion. In *European Conf. on Computer Vision*, pages 289–304. Springer, 2016. 6, 7

[9] T. Goldstein, B. O'Donoghue, S. Setzer, and R. Baraniuk. Fast alternating direction optimization methods. *SIAM Jour. on Imaging Sciences*, 7(3):1588–1623, 2014. 5

[10] R. Hartley, J. Trumpf, Y. Dai, and H. Li. Rotation averaging. *Int. Journal of Computer Vision*, 103(3):267–305, 2013. 2

[11] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge uni. press, 2003. 2

[12] P. W. Holland and R. E. Welsch. Robust regression using iteratively reweighted least-squares. *Communications in Statistics-theory and Methods*, 1977. 2, 4

[13] Y. Hu, D. Zhang, J. Ye, X. Li, and X. He. Fast and accurate matrix completion via truncated nuclear norm regularization. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 35(9):2117–2130, 2013. 2

[14] N. Levi and M. Werman. The viewing graph. In *Computer Vision and Pattern Recognition, 2003. Proc. 2003 IEEE Computer Society Conf. on*, volume 1, pages I–I. IEEE, 2003. 2

[15] M. I. Lourakis and A. A. Argyros. Sba: A software package for generic sparse bundle adjustment. *ACM Trans. on Mathematical Software*, 36(1):2, 2009. 8

[16] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 6

[17] D. Martinec and T. Pajdla. Robust rotation and translation estimation in multiview reconstruction. In *2007 IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. 1

[18] T. Okatani and K. Deguchi. On the wiberg algorithm for matrix factorization in the presence of missing components. *International Journal of Computer Vision*, 72(3):329–337, 2007. 2

[19] O. Ozyesil and A. Singer. Robust camera location estimation by convex programming. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2674–2683, 2015. 1, 6, 7

[20] O. Ozyesil, A. Singer, and R. Basri. Stable camera motion estimation using convex programming. *SIAM Jour. on Imaging Sciences*, 8(2):1220–1262, 2015. 1

[21] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM transactions on graphics (TOG)*, volume 25, pages 835–846. ACM, 2006. 1, 7

[22] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *European Conf. on Computer Vision*, pages 709–720. Springer, 1996. 2

[23] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *Int. Journal of Computer Vision*, 9(2):137–154, 1992. 2

[24] B. Triggs. Factorization methods for projective structure and motion. In *Computer Vision and Pattern Recognition, IEEE Computer Society Conf. on*, pages 845–851, 1996. 2

[25] R. Tron and R. Vidal. Distributed 3-d localization of camera sensor networks from 2-d image measurements. *IEEE Trans. on Automatic Control*, 59(12):3325–3340, 2014. 1

[26] K. Wilson and N. Snavely. Robust global translations with 1dsfm. In *European Conf. on Computer Vision*, pages 61–75. Springer, 2014. 1, 6, 7

[27] Z. Zhang and G. Xu. A general expression of the fundamental matrix for both perspective and affine cameras. In *Proc. of the 15th IJCAI*, pages 1502–1507. Morgan Kaufmann Publishers Inc., 1997. 2