Structure from Motion Using Structure-less Resection

Enliang Zheng
The University of North Carolina at Chapel Hill
ezheng@cs.unc.edu

Changchang Wu
Google
ccwu@google.com

Abstract

This paper proposes a new incremental structure from motion (SfM) algorithm based on a novel structure-less camera resection technique. Traditional methods rely on 2D-3D correspondences to compute the pose of candidate cameras using PnP. In this work, we take the collection of already reconstructed cameras as a generalized camera, and determine the absolute pose of a candidate pinhole camera from pure 2D correspondences, which we call it semi-generalized camera pose problem. We present the minimal solvers of the new problem for both calibrated and partially calibrated (unknown focal length) pinhole cameras. By integrating these new algorithms in an incremental SfM system, we go beyond the state-of-art methods with the capability of reconstructing cameras without 2D-3D correspondences. Large-scale real image experiments show that our new SfM system significantly improves the completeness of 3D reconstruction over the standard approach.

1. Introduction

The standard incremental structure from motion (SfM) is a widely used technique [16, 18, 15, 5]. During the incremental reconstruction, cameras with estimated poses are added to the 3D model repeatedly, which is a process called camera resection. Traditionally, the pose estimation step uses PnP algorithms, leveraging the correspondences between the 3D points and 2D features [9, 20, 19]. However, such structure-based resection method requires sufficient 3D points to be visible in the new cameras, which cannot be always satisfied even when there are enough feature matches. Figure 1 shows an extreme case of such a problem. Each of the three images captures two out of three objects in the scene, and no two-view reconstruction can be used to resect a third camera, because there are no three-view overlaps. In general, feature tracks are not always triangulated to 3D points due to pose inaccuracy, outlier feature matches, and threshold settings. This can easily lead to incomplete reconstructions with the standard SfM approach, even when there are sufficient feature matches.

Figure 1. There are 3 objects in the scene: sugar (O_1), blueberries (O_2) and vitamin (O_3). Each of the three images on the left can see only two out of the three objects, where the lack of three-view overlap prohibits standard resection. Note there are not reliable feature matches on the table due to the repeating patterns. The right image shows the dense reconstruction [3] from our reconstructed cameras using structure-less resection.

In this paper, we introduce a novel structure-less resection technique that exploits solely 2D matches for exact camera pose estimation that maximizes the number of potential 3D points. By taking the set of already reconstructed pinhole cameras as a single generalized camera [14], we register a new pinhole camera to the generalized camera using the 2D image correspondences (see Figure 2a) between the multiple cameras. Given the example in Figure 1, we may first compute the two-view reconstruction of the first two images (which reconstructs O_2 only), use the two cameras together as a generalized camera to resect the third camera from the 2D matches on object O_1 and O_3, and then reconstruct O_1 and O_3.

We name this new problem semi-generalized camera pose estimation since it involves one generalized camera and one pinhole camera. This paper presents the minimal solvers for the semi-generalized camera pose estimation problem, with the pinhole camera either calibrated or

partially calibrated (unknown focal length). The calibrated case has 6 degrees of freedom (3 in translation and 3 in rotation), and the case with unknown focal length accordingly has 7 degrees of freedom. The two cases respectively require 6 and 7 2D correspondences to solve the minimal problem. For convenience, we refer to the two problems as the 6-point problem and 7-point problem respectively.

These semi-generalized camera pose problems become more complicated than the fully generalized camera pose problem [17], when considering the concentric rays from the multiple pinhole cameras within the generalized camera (e.g. Figure 2). Although the 6-point solver by Stewenius et al. [17] works for the calibrated semi-generalized problem when there are no concentric rays among the generalized camera [7], its formulation leads to an infinite number of trivial solutions for many other configurations of our 6-point problems. Other non-minimal methods have been proposed specifically for moving multi-camera rigs such as [8] and [12], but cannot be directly applied as well. This paper handles the previously unsolved 6-point problems with a set of new polynomial constraints, and our solution to the 7-point problems goes beyond the state-of-the-art generalized camera pose methods by handling unknown focal lengths.

Our solutions to the semi-generalized pose estimation problems enable structure-less resection and accordingly a new incremental SfM system. Our method effectively deals with challenging camera poses that were difficult for the standard structure-based resection, and improves the completeness for incremental reconstructions.

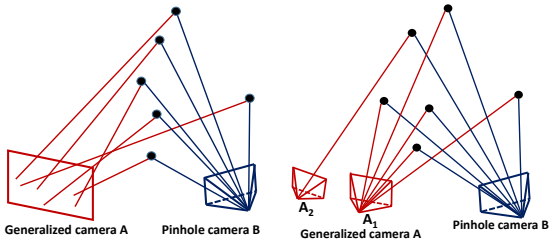
The main contributions of this paper include:

- Theoretical analysis for a set of new semi-generalized camera pose estimation problems.
- Minimal solutions to the new problems for calibrated or partially calibrated pinhole cameras.
- An improved incremental structure from motion algorithm that utilizes the new structure-less resection.

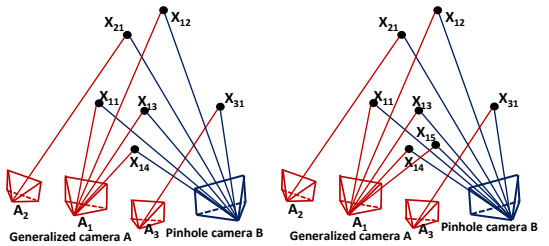
2. The Problem

We first introduce some notations for easy illustrations. As shown in Figure 2a, we denote the generalized camera and the pinhole camera as A and B respectively. Within camera A, each pinhole camera is denoted as A_i, where i is the camera index. Unlike previous methods that assume the generalized camera as a set of arbitrary rays, we explicitly model the number of viewing rays from A_i, denoted as $|A_i|$. For convenience, we let A_1 be the pinhole camera that has the **largest** number of viewing rays.

In the context of registering a new pinhole cameras to the existing camera system, it is important to deal with a group of concentric rays, rather than assuming each ray is from different cameras. Otherwise, a new camera need to have matches with about 6 images in order to be resected,



(a) 6-point problem, $|A_1| = 1$ (b) 6-point problem, $|A_1| = 5$



(c) 6-point problem, $|A_1| = 4$ (d) 7-point problem, $|A_1| = 5$

Figure 2. Illustration of several semi-generalized pose estimation problems. A_1 is the pinhole camera within A that has the largest number of viewing rays. X_{ij} is used to denote the unknown intersection of $j-$th viewing ray from camera A_i and its corresponding ray from B. This paper presents solutions to (b), (c), (d) and the 7-point problem with $|A_1| = 6$.

6-Point	$\|A_1\|$	6	5	4	3	≤ 2
	# of solutions	-	20	40	56	64
7-Point	$\|A_1\|$	6	5	4	3	≤ 2
	# of solutions	18	50	84	108	118

Table 1. The number of solutions for the 6-point and 7-point problem increases as $|A_1|$ decreases.

which significantly limits the applicability of the technique. For instance, when registering B to two cameras A_1 and A_2 within A, the generalized camera A must have $|A_1| \geq 3$.

We formulate the 6-point and 7-point semi-generalized pose problems in Macaulay2 [4] to investigate the number of solutions (Formulation details in Section 4 and 5). We discover that the number of solutions is determined by the largest number of concentric rays $|A_1|$. Table 1 shows how the number of solutions increases as the viewing rays from camera A_1 decreases. The semi-generalized camera pose problem can be considered as a transitional problem between the relative pinhole camera pose problem and the fully generalized camera pose problem. It can be seen that, as the number of concentric rays $|A_1|$ decreases, the problem becomes less 'pinhole' and more 'generalized'.

In the following sections, we present our solvers for the various semi-generalized pose problems. For convenience, we use $|A_1| + (|A| - |A_1|)$ to denote the $|A|$-point problem that has the largest number of $|A_1|$ rays from A_1. Section 3

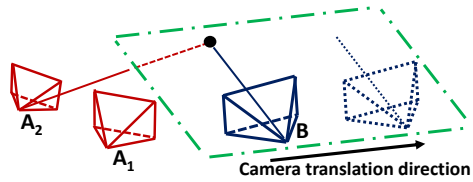


Figure 3. The viewing ray from camera B and the up-to-scale translation determine a 3D plane. The viewing ray from camera A_2 intersects the plane to determine the translation scale, but there exists degeneracy if the ray is in the plane.

first presents the solutions to the $5+1$ problem (Figure 2b) and the $6+1$ problem based on existing relative pose solvers [13, 1]. Afterwards, we present the solutions to the 6-point problems in section 4 and the 7-point problems in section 5.

3. The 5+1 and 6+1 solvers

The semi-generalized pose estimation becomes much more simplified when there is a single ray not from A_1, and we develop these solvers by exploiting two existing relative pose solvers.

For the calibrated $5+1$ problem shown in Figure 2b, the generalized camera A has five rays coming out of camera A_1 and one ray from camera A_2. We first compute the essential matrix for A_1 and B using the 5 correspondences [13] , which gives up to 10 solutions. Each essential matrix can then be decomposed into 2 rotations and 1 up-to-scale translation. For each rotation, the scale of the translation is then determined in general by intersecting the remaining rays from A_2 and B (see Figure 3).

Similarly for the $6+1$ problem with unknown focal length, we apply the solver by [1] to recover the essential matrix and focal length, which also has 10 solutions. The rotation and translation are then recovered similarly as the $5+1$ problem. Note that 1 out of 10 solutions from [1] is always trivial, which accounts for the difference with the minimal 18 solutions we show in Table 1.

It is clear from the algorithms and the number of solutions that the $5+1$ and $6+1$ problems are not much more than the pinhole camera pose problems, so in this case the semi-generalized camera pose problem is quite 'pinhole'.

4. The 6-point solvers for $|A_1| \leq 4$

This section first discuss our parameterization for 6-point problem, and then presents our polynomial system for the the minimal solvers.

4.1. Parameterization

For a calibrated generalized camera, each 2D measurement corresponds to a unique line in the camera coordinate system. This line can be represented as a Plucker line vector $L = (q^\top, q'^\top)^\top$, such that the set of points $X(\lambda)$ on the line can be parameterized as

$$X(\lambda) = q' \times q + \lambda q. \tag{1}$$

$q' = 0$ if and only if the line passes the origin of the coordinate system. More details can be found in [14].

In our problem, the generalized camera A is composed of multiple pinhole cameras $\{A_i\}$. The j-th 3D ray from camera A_i is denoted as $L_{ij} = [q_{ij}^\top, q_{ij}'^\top]^\top$. The k-th ray from camera B is denoted as $L_k = [q_k^\top, q_k'^\top]^\top$. The 3D point by the intersection of L_{ij} and L_k is denoted as X_{ij}. See Figure 2 for the illustration. The relationship between the 2D correspondences between A and B is given by the well-known generalized epipolar constraint [14, 17]:

$$\begin{aligned} q_k^\top R_B R_A^\top q'_{ij} + q_k^\top (R_B R_A^\top [t_A]_\times - [t_B]_\times R_B R_A^\top) q_{ij} \\ + q_{ij}^\top R_A R_B^\top q'_k = 0 \end{aligned} \tag{2}$$

where R_A, t_A, R_B, t_B are the rotation and translation of camera A and B.

Without loss of generality, we may assume identity rotation $R_A = I$ and origin position $t_A = 0$ for the generalized camera A. Since B is a pinhole camera, we may let the camera center be the origin of its local coordinate system, so that $q'_k = 0$ for B. The semi-generalized epipolar constraint can be written as

$$q_k^\top R_B \; q'_{ij} - q_k^\top [t_B]_\times R_B \; q_{ij} = 0. \tag{3}$$

Moreover, we may define the plucker coordinate of A such that $q'_{1j} = 0$ for viewing rays from A_1. The relationship between A_1 and B is basically an essential matrix:

$$q_k^\top [t_B]_\times R_B \; q_{1j} = 0. \tag{4}$$

To build the polynomial systems, we parameterize the rotation R_B using a homogeneous quaternion vector $[s, v_x, v_y, v_z]$ and set $s = 1$. Although this eliminates the possibility of $s = 0$, it is typically fine in real applications and has been widely used in minimal problems [6, 17, 8].

4.2. Polynomial system

Consider the j-th viewing ray from camera A_i intersects the k-th viewing ray from camera B at 3D point X_{ij}. Based on Eq. (1), we have the following in the two cameras:

$$\begin{aligned} q'_{ij} \times q_{ij} + \lambda_{ij} q_{ij} &= X_{ij} \\ \lambda_k q_k &= R_B \; X_{ij} + t_B, \end{aligned}$$

from which we can eliminate X_{ij} and obtain

$$t_B = \lambda_k \; q_k - R_B \; (q'_{ij} \times q_{ij} + \lambda_{ij} \; q_{ij}) \tag{5}$$

By substituting t_B into Eq. (3), we may obtain five equations in unknown parameters R_B, λ_{ij} and λ_k from the remaining 5 correspondences. These equations are linear in λ_{ij} and λ_k, and can be written as

$$\begin{bmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \\ F_{41} & F_{42} & F_{43} \\ F_{51} & F_{52} & F_{53} \end{bmatrix} \begin{bmatrix} \lambda_{ij} \\ \lambda_k \\ 1 \end{bmatrix} = 0. \tag{6}$$

Since Eq. (6) has non-trivial solutions, the left 5×3 matrix $F_{5\times3}$ has the rank constraint that $\text{rank}(F_{5\times3}) < 3$. Then the determinant of any 3×3 submatrix, which is composed of any three rows in $F_{5\times3}$, should equal to 0.

When $i \neq 1$. The determinant of each submatrix is a polynomial in unknown parameters $[v_x, v_y, v_z]$, so we get 10 polynomial equations for all the 10 submatrices. Note these polynomial are simplified version of the constraints in [17], which corresponds to the generalized epipolar constraint. To differentiate with other polynomial constraints that will be introduced later, we call these polynomial equations type $\mathbb{E}_1$ for convenience.

When $i = 1$. Since we choose the coordinate system such that $q'_{1j} = 0$, the relationship is further simplified to

$$t_B = \lambda_k \, q_k - \lambda_{1j} \, R_B \, q_{1j}, \tag{7}$$

which leads to a special form of Eq. (6). For example in the $4 + 2$ problem, the polynomial system is as follows:

$$\begin{bmatrix} F_{11} & F_{12} & 0 \\ F_{21} & F_{22} & 0 \\ F_{31} & F_{32} & 0 \\ F_{41} & F_{42} & F_{43} \\ F_{51} & F_{52} & F_{53} \end{bmatrix} \begin{bmatrix} \lambda_{1j} \\ \lambda_k \\ 1 \end{bmatrix} = 0. \tag{8}$$

The third elements of the first $|A_1| - 1$ rows are 0, because they correspond to the remaining rays from A_1, and the relative relationship between A_1 and B is up to scale. In the case of $4 + 2$, we may rewrite the first 3 equations as

$$\begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \\ F_{31} & F_{32} \end{bmatrix} \begin{bmatrix} \lambda_{1j} \\ \lambda_k \end{bmatrix} = 0. \tag{9}$$

Similar to $F_{5\times3}$ above, any 2×2 submatrix in the left matrix must be rank deficient and have a determinant of 0. We call these polynomial equations of type $\mathbb{E}_2$. These equations are basically constraints between the two pinhole cameras A_1 and B. Note these constraints do not exist in [17], which considers fully generalized cameras.

For semi-generalized pose problem with $|A_1| \leq 3$, we find that the type $\mathbb{E}_1$ equations are sufficient for solving the camera poses, with a caveat that there are $8 = 64 - 56$ trivial solutions when $|A_1| = 3$. The minimal solvers for the $|A_1| \leq 3$ 6-point problems can then be built using only $\mathbb{E}_1$ equations, which will be a simplified version of [17] thanks to the pinhole camera B on one side.

Problem	6-point		7-point			
Equation	$\mathbb{E}_1$	$\mathbb{E}_2$	$\mathbb{E}_3$	$\mathbb{E}_4$	$\mathbb{E}_5$	$\mathbb{E}_6$
Degree	6	4	10	7	11	8
Monomials	84	35	382	129	440	162
Lin. indep. eqs	14	4	30	10	30	15

Table 2. The degree and the number of monomials of our polynomials. The last row is the number of linearly independent polynomials in the 4+2 and 5+2 solvers.

Matrix	Multipliers
M_1	$1, v_x, v_y, v_z$
M_2	$1, v_x, v_y, v_z, v_x^2, v_y^2, v_z^2$
M_3, M_5	$1, w_c, u_x, u_y$
M_4, M_6	$1, w_c, w_s, u_x, u_y, w_c^2, w_s^2, u_x^2, u_y^2,$ $w_c w_s, w_c u_x, w_c u_y, w_s u_x, w_s u_y, u_x u_y$

Table 3. The coefficient matrices and their multipliers for the 6-point and 7-point solvers. Each matrix M_i corresponds to the echelon form of the raw coefficient matrix of all type $\mathbb{E}_i$ equations.

We will focus on the $4 + 2$ solver in this paper, which requires both the $\mathbb{E}_1$ and $\mathbb{E}_2$ equations. After enumerating all possible i, we have a collection of many equations of type $\mathbb{E}_1$ and $\mathbb{E}_2$, which are not all linearly independent. It turns out there are respectively 14 and 4 linearly independent equations of type $\mathbb{E}_1$ and $\mathbb{E}_2$. We also find that the equations of type $\mathbb{E}_1$ from Eq. (8) is linearly depedent and do not need to be considered. Table 2 shows the properties of the polynomial equations of different types. The $14 + 4$ equations of type $\mathbb{E}_1$ and $\mathbb{E}_2$ gives a polynomial equation system with exactly 40 solutions for our problem.

4.3. Gröbner basis solver

The standard approach for solving polynomial systems typically involves Gauss-Jordan (G-J) elimination on elimination template, action matrix construction, and eigenvector-based solution from the action matrix. Since a detailed description of the method is beyond the scope of this paper, we will briefly describe the key steps for constructing our action matrix, and refer the readers to [2, 10, 11] for more details.

By modeling the problem Macaulay2 [4], we first compute the bases of the quotient ring, and use them to design the following action matrix computation. For each type of equations $\mathbb{E}_i$, a coefficient matrix is first constructed, such that each row corresponds to one polynomial equation of type $\mathbb{E}_i$ and the columns are in the GRevLex monomial ordering. G-J elimination is then applied to the coefficient

matrix to get the echelon form matrix, denoted as M_i. The polynomial equations represented by $M_i, i = 1, 2$ are multiplied by each variable listed in Table 3, and stacked together to get the elimination template. For instance, the multipliers $\{1, v_x, v_y, v_z\}$ are used for M_1. The action matrix can be extracted from the echelon form of the elimination template. Here the elimination template size can be slightly reduced under the condition that the action matrix can still be constructed [11].

5. The 7-point solvers for $|A_1| \leq 5$

This section presents the minimal solver for the 7-point problems (e.g. Figure 2d). We will first extend the parameterization used in the 6-point problems to include the unknown focal length, then introduce the polynomial systems for the minimal solvers.

5.1. Parameterization

Let f be the unknown focal length of the pinhole camera B. To avoid trivial solutions of $f = 0$, we define the inverse of the calibration matrix K as

$$K^{-1} = \begin{bmatrix} w & 0 & 0 \\ 0 & w & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{10}$$

where $w = 1/f$. Let $\hat{q}_k$ be an observed image point in B, the corresponding ray direction is given by $K^{-1}\hat{q}_k$. By including this mapping in the generalized epipolar constraint in Eq. (3), we obtain the following

$$(K^{-1}\hat{q}_k)^\top R_B q'_{ij} - (K^{-1}\hat{q}_k)^\top [t_B]_\times R_B \; q_{ij} = 0. \tag{11}$$

Let $C_B = -R_B^\top t_B$ be the camera center of B, we have $-[t_B]_\times R_B = R_B[C_B]_\times$, with which we can derive

$$\hat{q}_k^\top (K^{-1}R_B) \; q'_{ij} + (\hat{q}_k)^\top (K^{-1}R_B)[C_B]_\times \; q_{ij} = 0. \tag{12}$$

As recently demonstrated by [19], it leads to a solution doubling effect when parameterizing both focal length and camera rotation. The redundancy is caused by mirrored solutions with negative focal lengths. In our problem, we find such a straightforward parameterization producing 100 solutions instead of the expected 50, which is unnecessarily complicated. Similar to the parametrization in [19], we decompose rotation parameter into two components $R_B = R_\theta R_\rho$, such that R_θ is a rotation around z axis, and R_ρ is a rotation around an axis in x-y plane. R_θ has one degree of freedom, and R_ρ, which is parameterized as $[1, u_x, u_y, 0]$, has two degrees of freedom. Now $K^{-1}R_B$ can be re-parameterized by combining K^{-1} and R_θ as the following:

$$K^{-1}R_B = K^{-1}R_\theta R_\rho = \begin{bmatrix} w\cos\theta & -w\sin\theta & \\ w\sin\theta & w\cos\theta & \\ & & 1 \end{bmatrix} R_\rho \tag{13}$$

Let $w_c = w\cos\theta$, and $w_s = w\sin\theta$, Eq. (13) has

$$K^{-1}R_B = \begin{bmatrix} w_c & -w_s & \\ w_s & w_c & \\ & & 1 \end{bmatrix} R_\rho. \tag{14}$$

Similar to the elimination of translation in the 6-point problem, we may first reduce the number of variables by eliminating the camera center. This results in a problem with four unknown parameters $\{w_c, w_s, u_x, u_y\}$, as opposed to $\{u_x, u_y, y_z\}$ in the 6-point problem. Once w_c and w_s are computed, the rotation angle θ and the focal length w can be easily extracted.

5.2. Polynomial system

Using the same formula as the 6-point problems, we generate two similar types of polynomial equations. By using a 3D point X_{ij} to eliminate C_B, the resulting polynomial system has the following form:

$$\begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{21} & Q_{22} & Q_{23} \\ Q_{31} & Q_{32} & Q_{33} \\ Q_{41} & Q_{42} & Q_{43} \\ Q_{51} & Q_{52} & Q_{53} \\ Q_{61} & Q_{62} & Q_{63} \end{bmatrix} \begin{bmatrix} \lambda_{ij} \\ \lambda_k \\ 1 \end{bmatrix} = 0. \tag{15}$$

The determinant of any 3×3 submatrix is 0, and generates a polynomial constraint in the four unknown parameters. We call these polynomial equation type $\mathbb{E}_3$.

For the first camera A_1, the third element of the first $|A_1| - 1$ rows would be 0, from which we can construct polynomial constraints similar to $\mathbb{E}_2$. For example, in the case of $5 + 2$ problem, we have

$$\begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \\ Q_{31} & Q_{32} \\ Q_{41} & Q_{42} \end{bmatrix} \begin{bmatrix} \lambda_{1j} \\ \lambda_k \end{bmatrix} = 0, \tag{16}$$

Similarly, the polynomial equations from any of the 2×2 submatrices is defined as equation of type $\mathbb{E}_4$.

Nevertheless, we find the polynomial equation system from the $\mathbb{E}_3$ and $\mathbb{E}_4$ equations has infinite number of trivial solutions. To explain this, we discover that the left 6×3 matrix in Eq. (15) has the following structure

$$\begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{21} & Q_{22} & Q_{23} \\ Q_{31} & Q_{32} & Q_{33} \\ Q_{41} & Q_{42} & Q_{43} \\ Q_{51} & Q_{52} & Q_{53} \\ Q_{61} & Q_{62} & Q_{63} \end{bmatrix} = \begin{bmatrix} Q_{11} & Q^1_{12}w_c + Q^2_{12}w_s & Q_{13} \\ Q_{21} & Q^1_{22}w_c + Q^2_{22}w_s & Q_{23} \\ Q_{31} & Q^1_{32}w_c + Q^2_{32}w_s & Q_{33} \\ Q_{41} & Q^1_{42}w_c + Q^2_{42}w_s & Q_{43} \\ Q_{51} & Q^1_{52}w_c + Q^2_{52}w_s & Q_{53} \\ Q_{61} & Q^1_{62}w_c + Q^2_{62}w_s & Q_{63} \end{bmatrix}. \tag{17}$$

Similar structure exits in the left matrix in Eq. (16),

$$\begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \\ Q_{31} & Q_{32} \\ Q_{41} & Q_{42} \end{bmatrix} = \begin{bmatrix} Q_{11} & Q^1_{12}w_c + Q^2_{12}w_s \\ Q_{21} & Q^1_{22}w_c + Q^2_{22}w_s \\ Q_{31} & Q^1_{32}w_c + Q^2_{32}w_s \\ Q_{41} & Q^1_{42}w_c + Q^2_{42}w_s \end{bmatrix}. \quad (18)$$

It can be seen that $w_c = w_s = 0$ is a trivial solution to the polynomial system by making one column all zeros, and u_x and u_y can be any values. To avoid such trivial solutions, we can rewrite Eq. (15) as

$$\begin{bmatrix} Q_{11} & Q^1_{12} & Q^2_{12} & Q_{13} \\ Q_{21} & Q^1_{22} & Q^2_{22} & Q_{23} \\ Q_{31} & Q^1_{32} & Q^2_{32} & Q_{33} \\ Q_{41} & Q^1_{42} & Q^2_{42} & Q_{43} \\ Q_{51} & Q^1_{52} & Q^2_{52} & Q_{53} \\ Q_{61} & Q^1_{62} & Q^2_{62} & Q_{63} \end{bmatrix} \begin{bmatrix} \lambda_{ij} \\ \lambda_k w_c \\ \lambda_k w_s \\ 1 \end{bmatrix} = 0, \quad (19)$$

from which the determinant of any 4×4 submatrices defines a new polynomial constraint, which we call type $\mathbb{E}_5$. Similarly, Eq. (16) can be rewritten as

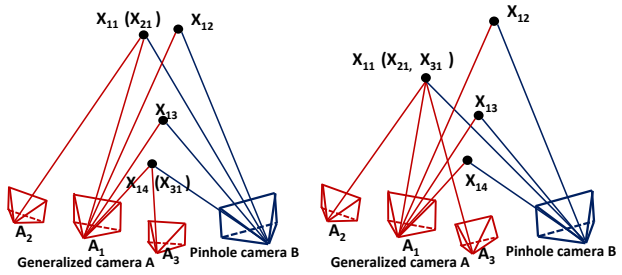
$$\begin{bmatrix} Q_{11} & Q^1_{12} & Q^2_{12} \\ Q_{21} & Q^1_{22} & Q^2_{22} \\ Q_{31} & Q^1_{32} & Q^2_{32} \\ Q_{41} & Q^1_{42} & Q^2_{42} \end{bmatrix} \begin{bmatrix} \lambda_{1j} \\ \lambda_k w_c \\ \lambda_k w_s \end{bmatrix} = 0. \quad (20)$$

where determinant from any of the 3×3 submatrices defines a new polynomial constraint, we call type $\mathbb{E}_6$.

As numerical stability decreases when the number of solutions increases, it is worth focusing on the problems with lower degrees. Here we will only detail for the $5 + 2$ problem, while other solvers can be built similarly. It can be verified with Macaulay2 that the polynomial system comprised of all the $\mathbb{E}_3$, $\mathbb{E}_4$, $\mathbb{E}_5$ and $\mathbb{E}_6$ type of equations gives exactly 50 solutions for the $5 + 2$ problem. The properties of these equations are shown in Table 2.

5.3. Gröbner basis solver

The solver for the 7-point problems is developed similar to the 6-point problems, but a special scheme is necessary due to its very high polynomial degrees. Notice that some of the polynomial equations represented by M_3 and M_5 are of degree 10 and 11 respectively, corresponding to $\mathbb{E}_3$ and $\mathbb{E}_5$ (Table 2). This high polynomial degree degrades the accuracy and efficiency of the solver, and we find it prohibits automatic solver generator [10] from finding the solutions. Using Macaulay2, we discover that polynomials of degrees 9 and higher in M_3 and M_5 can be safely removed. This is the key step to solve the polynomial equation system in 7-point solvers. With the reduced set of polynomials, we compute the elimination template and its echelon form, from which the action matrix is then extracted. After recovering the rotation and focal length, the translation vector t_B can be then computed according to Eq. (11) using SVD.



(a) (b)

Figure 4. Possible configurations with replicated rays in camera B, where 4a still has 40 solutions, but 4b is unsolvable.

6. Incremental structure from motion

We integrate the 6/7-point solvers to incremental SfM as a complementary camera resection method to the standard PnP-based one. Since the new scheme does not require any 3D point position, we refer to it as structure-less resection. Incremental SfM is effectively improved by the more chances of registering new cameras.

6.1. Integrating the structure-less resection

The selection of candidate camera for resection now needs to consider potential ray intersections in addition to visible 3D points. As structure-based resection is quite accurate and fast, we first try the normal selection scheme that selects the candidate camera if it sees sufficient 3D points. If no camera sees sufficient 3D points, we will pick the camera with the largest number of tracks that contain any reconstructed cameras. These tracks could be either seen by only one reconstructed camera (e.g. Figure 1), or not triangulated due to baseline thresholds. The number of potential new 3D points is basically the number of un-triangulated tracks containing previously reconstructed cameras.

For each selected camera candidate, we first try standard PnP-based resection, and if it fails, we use structure-less resection. Each track shared by the camera candidate and any previously reconstructed camera gives a 2D correspondence. Given a set of 2D correspondences, we use the 6/7-point solvers in a RANSAC framework to resect the candidate camera. Similar to PnP-based RANSAC, we recover the camera pose that yields the largest number of ray intersections. After each successful resection, we triangulate more tracks, run bundle adjustment, and move to the next camera candidate.

6.2. Our RANSAC

The sampling of 2D correspondences needs to provide ray pairs of the expected configurations for the minimal solver (e.g. 6+1). We also pay special attention to ray replication during the sampling. Since the generalized camera A contains multiple pinhole cameras, it is possible that one ray

from B intersects rays from multiple pinhole cameras in A (See Figure 4), which happens in practice when 3D structure is seen by multiple cameras. In this case we say that the viewing ray from camera B is (algebraically) replicated, since we can consider that multiple rays from B coincide.

Although our solvers were originally intended to work for ray correspondences without replication, we find them working fine if the rays from B are either unique or replicated at most **twice**. Figure 4 shows two possible cases of ray replication for the 6-point problem. In fact, we can determine if a problem is solvable by counting the number of constraints from triangulated 3D points and rays. For the problem in Figure 4a, the 3D point X_{11} (it coincides with X_{21}) can be triangulated using correspondences from A_1 and A_2, and similarly for point X_{14}. The two 3D points and two additional rays give exactly 6 constraints, which makes the problem solvable [7]. On the contrary, the problem in Figure 4b is equivalent to having one 3D-2D correspondence and 3 2D-2D correspondences, which only gives 5 constraints and is hence unsolvable.

7. Experiments

This section evaluates the performance of the 6-point and 7-point solvers on synthetic data and evaluates the effectiveness of our SfM system on internet photo collections.

7.1. Solver speed

We evaluate the speed of the solvers on a Linux machine with an Intel Xeon X5650 @2.67GHz CPU. The average running time for the four solvers are listed in Table 4. All the solvers have a reasonable speed for real applications, and it can be seen that the $5+1$ and $6+1$ solvers are significantly faster than the corresponding $4+2$ and $5+2$ solvers due to their simpler polynomial systems.

7.2. Stability and accuracy

We use synthetic data to quantitatively evaluate the numerical stability on noise-free data and the accuracy on noisy data. For the 6-point problem, 3D points and cameras are uniformly generated in the cube $[-2,2]\times[-2,2]\times[0,2]$ and $[-2,2]\times[-2,2]\times[-1,0]$. The 3D points are then projected into the cameras to produce the 2D image correspondences. Camera rotations are random but with the principal direction pointing to a random position in the cube $[-2,2]\times[-2,2]\times[0,2]$. The rotation, translation and focal length errors are evaluated according to Table 5 on 10K randomly generated testing samples.

Solver	5+1	4+2	6+1	5+2
Matrix	10×20	73×113	10×20	378×428
Time (ms)	0.048	1.2	0.046	13.6

Table 4. The comparison of speed for the four solvers, where the second row is the size of their elimination template matrix.

Rotation	Translation	Focal length
$\delta_R = \angle(R_g R^\top)$	$\delta_t = \frac{\|\|t-t_g\|\|}{\|\|t_g\|\|}$	$\delta_f = \frac{\|f-f_g\|}{f_g}$

Table 5. The error definitions for rotation, translation and focal length, where the subscript g means ground truth. Given multiple solutions, the solution with the smallest translation error is used.

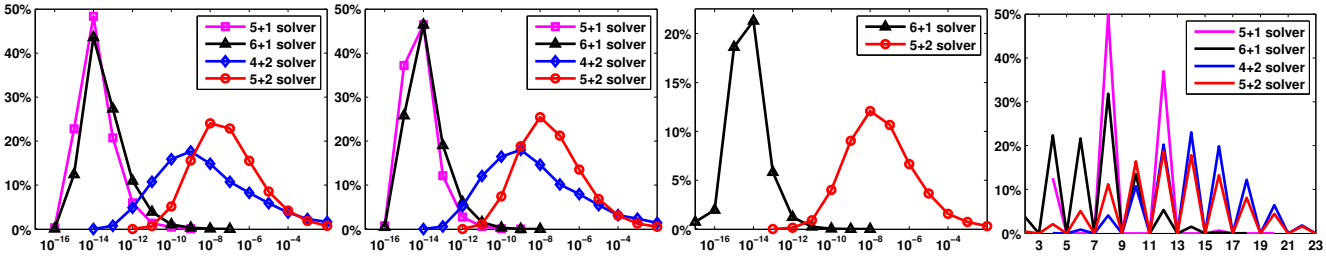
We first run noise-free random problems with the four solvers and evaluate their numerical stability. The focal length for pinhole camera is randomly drawn within the range of $[200, 2000]$. The resulting error distributions can be found in Figure 5. As expected, the solvers with simpler polynomial systems have better numerical stability since the G-J elimination on the larger elimination template are more likely to produce more numerical errors. Specifically, the $5+1$ and $6+1$ solvers have better stability compared to $4+2$ and $5+2$, and $4+2$ also has slightly better stability than $5+2$. Nevertheless, all the solvers are accurate enough for real applications with most errors less than 10^{-4}.

An opposite advantage in accuracy is however discovered for noisy data. We add zero-mean Gaussian noise with different standard deviations to 2D measures, and again run 10K random problems with the four solvers. To make the noise level corresponding to angular observation errors, we fix the ground-truth focal length to 1000 in this experiment. Figure 6 shows that the $4+2$ and $5+2$ solvers have higher accuracy than the corresponding $5+1$ and $6+1$ solvers for all the noise levels. This can be explained by the more balanced distribution of viewing rays within the generalized camera of configuration $4+2$ and $5+2$. Therefore, real applications should prefer the more accurate $4+2$ and $5+2$ solvers since the structure-less resection is used only sparely and speed is not a concern.

7.3. Real images

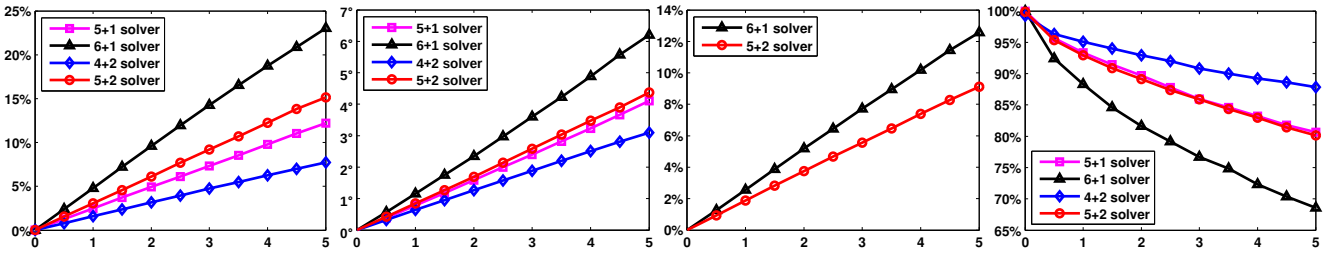
We have shown earlier in Figure 1 that our method is capable of incremental reconstruction without using 3-view overlaps. 3-view overlap is a requirement that many users often overlook when taking images for reconstruction, while our SfM allows to relax the capture requirement a little bit for 3D reconstruction.

To demonstrate its benefit on normal reconstruction problems, we randomly select 800 datasets from pubic Internet photos. Each dataset is a single connected component of image graph, with size ranging from 32 to 8K. Ideally, each connected component should produce a single model (except for outliers), but incremental SfM often have incomplete reconstructions due to accumulated errors and sometimes weak three-view overlaps in the connected compo-



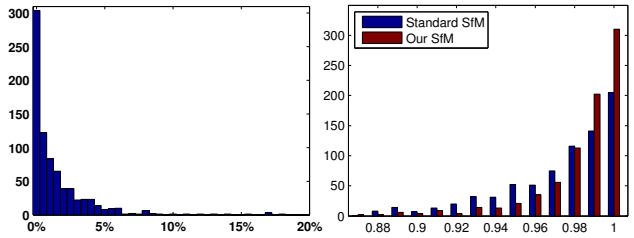
(a) Translation error δ_t (b) Rotation error δ_R (radian) (c) Focal length error δ_f (d) Number of solutions

Figure 5. The error distributions for noise-free data and the distribution for the number of solutions. To better handle large focal lengths, the same 2D normalization is applied to $6+1$ and $5+2$ to make the mean squared norm 2. All the solvers exhibit reasonable small errors, while the $5+1$ and $6+1$ solvers have lower errors because of their simpler polynomial systems.



(a) Median translation error δ_t (b) Median rotation error δ_R (degrees) (c) Median focal length error δ_f (d) Runs with $\delta_t < \frac{1}{2}$

Figure 6. The accuracy under different level of Gaussian noise, where the horizontal axes are the noise levels. The $4+2$ and $5+2$ solvers have better accuracy thanks to better distribution of viewing rays in A.



(a) Size increase by our SfM (b) Completeness of reconstruction

Figure 7. The improvements by our SfM over standard SfM on 800 datasets. The left is the histogram of model size increase with a bin size of 0.5%. The right is the histogram of completeness (model size over connected component size) with a bin size of 0.01.

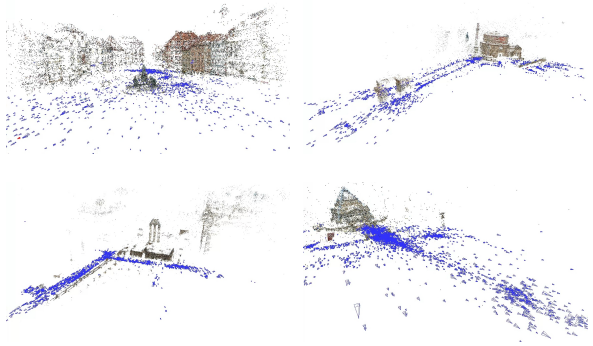


Figure 8. Four selected models from our experiments (The blue dots are the camera positions).

nents. Specifically for better accuracy, we use the $4+2$ and $5+2$ solvers for resection. Figure 7 shows that our new SfM system effectively improves the completeness for incremental reconstruction by providing an alternative method to register cameras when standard resection fails. Because standard SfM already produces reasonably large models, the increase is expected to be small in the experiments. A few examples of our reconstructions are shown in Figure 8.

We have observed a few bad cameras from the experiments, which are caused by registering to already inaccurate cameras from Standard SfM. Our future work includes robustness improvement for the structure-less resection.

8. Conclusion

Are 3-view overlap and 2D-3D correspondences indispensable for incremental SfM (of pinhole cameras)? Not anymore, with structure-less resection.

Acknowledgment We thank Henrik Stewénius for sharing the Macaulay2 code of their paper [17].

References

[1] M. Bujnak, Z. Kukelova, and T. Pajdla. 3d reconstruction from image collections with a single known focal length. In *ICCV*, 2009.

[2] D. A. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra, 3/e (Undergraduate Texts in Mathematics)*. Springer-Verlag New York, Inc., 2007.

[3] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *PAMI*, 2010.

[4] D. R. Grayson and M. E. Stillman. Macaulay2, a software system for research in algebraic geometry. Available at http://www.math.uiuc.edu/Macaulay2/.

[5] J. Heinly, J. Schönberger, E. Dunn, and J. Frahm. Reconstructing the World* in Six Days *(As Captured by the Yahoo 100 Million Image Dataset). In *CVPR*, 2015.

[6] K. Josephson and M. Byrd. Pose estimation with radial distortion and unknown focal length. In *CVPR*, 2009.

[7] K. Josephson, M. Byrd, F. Kahl, and K. strm. Image-based localization using hybrid feature correspondences. In *CVPR*, 2007.

[8] L. Kneip and H. Li. Efficient computation of relative pose for multi-camera systems. In *CVPR*, 2014.

[9] L. Kneip, D. Scaramuzza, and R. Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *CVPR*, 2011.

[10] Z. Kukelova, M. Bujnak, and T. Pajdla. Automatic generator of minimal problem solvers. In *ECCV*, 2008.

[11] Z. Kúkelová, T. Pajdla, and M. Bujnak. *Algebraic methods in computer vision*. PhD thesis, PhD thesis, Center for Machine Perception, Czech Technical University, Prague, Czech republic, 2012.

[12] H. Li, R. I. Hartley, and J. Kim. A linear approach to motion estimation using generalized camera models. In *CVPR*, 2008.

[13] D. Nistér. An efficient solution to the five-point relative pose problem. *IEEE Trans. PAMI*, 2004.

[14] R. Pless. Using many cameras as one. In *CVPR*, 2003.

[15] J. Schönberger, F. Radenović, O. Chum, and J. Frahm. From single image query to detailed 3d reconstruction. In *CVPR*, 2015.

[16] N. Snavely, S. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *IJCV*, 2008.

[17] H. Stewénius, D. Nistér, M. Oskarsson, and K. Åström. Solutions to minimal generalized relative pose problems. In *Workshop on Omnidirectional Vision*, 2005.

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

[19] C. Wu. P3.5P: Pose estimation with unknown focal length. In *CVPR*, 2015.

[20] Y. Zheng, S. Sugimoto, I. Sato, and M. Okutomi. A general and simple method for camera pose and focal length determination. In *CVPR2014*, 2014.