Variable Elimination for 3D from 2D.

Ji Zhang[a], Mireille Boutin[b] and Daniel G. Aliaga[c]

[a]School of Mathematics, Purdue University, 150 N. University St., West Lafayette, IN, USA ;
[b] School of ECE, Purdue University, 465 Northwestern Av., West Lafayette, IN, USA;
[c]Dept. of Computer Sc., Purdue University, 250 N. University St., West Lafayette, IN, USA.

ABSTRACT

Accurately reconstructing the 3D geometry of a scene or object observed on 2D images is a difficult problem: there are many unknowns involved (camera pose, scene structure, depth factors) and solving for all these unknowns simultaneously is computationally intensive and suffers from numerical instability. In this paper, we algebraically decouple some of the unknowns so that they can be solved for independently. Decoupling the pose from the other variables has been previously discussed in the literature. Unfortunately, pose estimation is an ill-conditioned problem. In this paper, we algebraically eliminate all the camera pose parameters (i.e., position and orientation) from the structure-from-motion equations for an internally calibrated camera. We then also fully eliminate the structure coordinates from the equations. This yields a very simple set of homogeneous polynomial equations of low degree involving only the depths of the observed points. When considering a small number of tracked points and pictures (e.g., five points on two pictures), these equations can be solved using the sparse resultant method.

Keywords: Structure from Motion, pose-free structure from motion, Depth from motion.

1. INTRODUCTION

Suppose a set of pictures of a 3D scene (or 3D object) is given. If the pictures were taken along a generic camera path, it is possible to use them to reconstruct an approximation of the 3D shape of the scene. For example, this can be done by tracking some distinguished scene points on several consecutive pictures and solving the set of equations describing how the position of the tracked 3D points relates to their positions on the pictures. There are several applications where this can be useful. For example, this can be applied to the automatic creation of 3D virtual models in computer graphics. Another application is the generation of depth maps for 3DTV.

In general, the numerical solution of equations is a challenging problem. In many cases, iterative methods (e.g., gradient descent and Newton's method) are the only solution tools available. But in the last ten years or so, a number of effective non-iterative methods based on the concept of resultant (recently revived by Gelfand et al.[1]) have been developed for the special case of polynomial equations. Some of these methods have even been implemented in symbolic computation packages such as Maple[2,3] or Singular.[4] However, most polynomial equations encountered in practice are still solved using iterative numerical methods. This is mostly due to the computational complexity and the storage requirements of the aforementioned symbolic methods. For example, the upper limit on the number of variables that can currently be handled using the sparse resultant method is believed to be around ten. Unfortunately, most problems encountered in practice involve a lot more than ten variables.[5] This is the case for the equations that concern us, the so-called *structure-from-motion equations*, as each 3D feature point introduces 3 variables, while each camera position introduces 6. The sparse resultant method is thus not practical for the structure-from-motion problem in its original form.

In order to modify the form of this problem, we use elimination. Elimination is the problem of taking given a set of polynomial equations, say

$$\begin{aligned} p_1(x,y) &= 0, \\ p_2(x,y) &= 0, \\ &\vdots \\ p_n(x,y) &= 0, \end{aligned}$$

and finding a set of equations

$$\begin{aligned} g_1(x) &= 0, \\ g_2(x) &= 0, \\ &\vdots \\ g_k(x) &= 0, \end{aligned}$$

which are satisfied for a given x if and only there exists a y such that (x, y) is a solution of the original system. For linear equations, this can be done by simple Gaussian elimination. The case of non-linear polynomial equations can be treated using different methods including a resultant based method initially proposed by Bézout[6] in the 18[th] century, as well as the more recent Groebner basis methods.[7, 8] In our case, we performed elimination on the set structure-from motion equations but without using any sophisticated method: we merely used a few tricks from invariant theory. By successively eliminating more and more variables from the structure from motion equations, we are able to reach a problem size that can be handled by some of the current result-based methods, as we will demonstrate in the following.

It is well known that one can eliminate all the variables except for the pose parameters.[9] This yields a simple system of quadratic equations directly relating the camera pose parameters to the tracked points seen on the pictures. This system of equations can be solved using the sparse resultant approach,[10] among other methods. However, it has been proved[11] that pose estimation is an ill-conditioned the problem. This is due to the fact that, from the pictures alone, a change in camera orientation is sometimes very difficult to distinguish from a change in camera position. Thus attacking the structure-from motion problem by first reconstructing the pose does not yield an accurate reconstruction. We thus use another elimination route: first eliminate the camera pose completely. We do this in two steps. First, we eliminate the camera orientation parameters in Section 4. Second, we further eliminate the camera pose parameters in Section 5. Finally, in Section 6, we eliminate the structure parameters as well. This yields a simple set of (3N-5)(J-1) equations in N variables, where N is the number of feature points and J is the number of pictures. For example, with two pictures and five feature points, we only have ten (homogeneous) equations and ten variables. Surprisingly, each step of this elimination sequence can be achieved without a significant increase in the degrees of the polynomial equations: all equations obtained are of degree two or three.

Notation and Assumptions:

For the remained of the paper, we will assume that we are working with an internally calibrated camera and that the camera's focal length is equal to one. We will assume that feature points have been tracked on a sequence of pictures taken by this camera. The total number of feature points tracked will be denoted by N and the 3D feature points will be denoted by P_i, for $i = 1, \ldots, N$. The total number of picture taken will be denoted by J and the 2D coordinates of point P_i as seen on picture j will be written as (x_{ij}, y_{ij}). We will embed each picture point (x_{ij}, y_{ij}) in 3D by setting $p_{ij} = (x_{ij}, y_{ij}, 1)$. The camera center position used to take picture j will be written as C_j. The variable c_{ij} will be used to represent the ratio $c_{ij} = \frac{\|p_{ij}\|}{\|P_i - C_j\|}$. To simplify the notation, we will later introduce a new variable $\gamma_{ij} = \frac{1}{c_{ij}}$.

2. STRUCTURE-(AND POSE)-FROM-MOTION EQUATIONS

The equations relating the position of the 3D feature points (so-called *structure*) and the camera pose (i.e. camera orientation and position) to the 2D position of the points tracked on the pictures can be written as[9]:

$$p_{ij} = c_{ij} F_j \begin{pmatrix} P_i \\ 1 \end{pmatrix}, \text{ for all } i = 1, \ldots, N \text{ and all } j = 1, \ldots, J, \tag{1}$$

where c_{ij} is an unknown positive constant, and F_j is an unknown 3-by-4 matrix containing the camera parameters corresponding to picture j. These are the well-known *structure-from-motion* equations, which have been studied in depth in the literature. When the camera is internally calibrated, one can assume that the fundamental matrix takes the form

$$F_j = \begin{pmatrix} R_j & T_j \end{pmatrix},$$

where R_j is a 3D rotation matrix and T_j is a 3D translation vector, both unknown. The solution of this set of equation is not unique. Indeed, Equations 1 only define the feature points P_i up to a rotation, a translation, and a rescaling. In other words, Equations 1 define the shape of the scene but not its absolute placement in space or its size.

3. POSE-FROM-MOTION EQUATIONS

Eliminating the feature points from Equations 1 is quite simple: just multiply both sides by the inverse of the matrix F_j and $\frac{1}{c_{ij}}$. This yields an equivalent system of equations:

$$\frac{1}{c_{ij}} F_j^{-1} p_{ij} = \begin{pmatrix} P_i \\ 1 \end{pmatrix}, \text{ for all } i = 1, \ldots, N \text{ and all } j = 1, \ldots, J.$$

Observe that the right-hand side is independent of the index j. So we can replace this system of equations by

$$\frac{1}{c_{ij}} F_j^{-1} p_{ij} = \frac{1}{c_{i\bar{j}}} F_{\bar{j}}^{-1} p_{i\bar{j}} \text{ for all } i = 1, \ldots, N \text{ and all } j, \bar{j} = 1, \ldots, J.$$

This can be rewritten as

$$\frac{c_{i\bar{j}}}{c_{ij}} F_{\bar{j}} F_j^{-1} p_{ij} = p_{i\bar{j}} \text{ for all } i = 1, \ldots, N \text{ and all } j, \bar{j} = 1, \ldots, J. \tag{2}$$

The above forms a complete set of *Pose-from-Motion* equations, which can be solved for the relative pose between two pictures. This set of equations is well known in the literature, but we reproduce it here for completeness. However, as we stated earlier, Fermueller and Aloimonos have shown[11] that the problem of pose estimation is ill-conditioned. So first solving this system for the pose will not yield an accurate result for the structure because even a small error in the pose can yield a big error in the 3D points positions P_i's. From our point of view, it is thus more urgent to eliminate the camera pose, rather than the other variables. This is what we accomplish in the two following sections.

4. STRUCTURE-AND-CAMERA-POSITION-FROM-MOTION EQUATIONS

In this section, we eliminate the camera orientation parameters from the structure-from-motion equations. As mentioned earlier, these equations have previously appeared in an earlier publication,[12] though in a slightly modified version and with a more complicated justification

In order to explicitly express every equation as a polynomial equation, we first let $\gamma_{ij} = \frac{1}{c_{ij}}$. To eliminate the rotation matrix R from the equation, we first observe that the vectors $P_i - C_j$ and $\gamma_{ij} p_{ij}$ are related by an orientation preserving rigid motion. Since the invariants of this group action are well known, it is easy to obtain a complete set of camera-orientation-free equations. Alternatively, we can obtain this set using some basic algebraic manipulations. We first pick any i and $\bar{i}$ among $1, \ldots, N$. We then have

$$p_{ij} = c_{ij} F_j \begin{pmatrix} P_i \\ 1 \end{pmatrix},$$

$$p_{\bar{i}j} = c_{\bar{i}j} F_j \begin{pmatrix} P_{\bar{i}} \\ 1 \end{pmatrix}.$$

Taking the dot product of the left-hand-sides and right-hand-sides of these two equations, respectively, we obtain

$$p_{ij} \cdot p_{\bar{i}j} = c_{ij} F_j \begin{pmatrix} P_i \\ 1 \end{pmatrix} \cdot c_{\bar{i}j} F_j \begin{pmatrix} P_{\bar{i}} \\ 1 \end{pmatrix}.$$

Writing $F_j = \begin{pmatrix} R_j & T_j \end{pmatrix}$, we have

$$\begin{aligned} \gamma_{ij}\gamma_{\bar{i}j} p_{ij} \cdot p_{\bar{i}j} &= \begin{pmatrix} R_j & T_j \end{pmatrix} \begin{pmatrix} P_i \\ 1 \end{pmatrix} \cdot \begin{pmatrix} R_j & T_j \end{pmatrix} \begin{pmatrix} P_{\bar{i}} \\ 1 \end{pmatrix}, \\ &= (R_j P_i + T_j) \cdot (R_j P_{\bar{i}} + T_j) \\ &= \left(P_i + R_j^T T_j\right) \cdot \left(P_{\bar{i}} + R_j^T T_j\right). \end{aligned}$$

Letting $C_j = -R_j^T T_j$, we get a first type of camera-orientation-free equation:

$$\gamma_{ij}\gamma_{\bar{i}j} p_{ij} \cdot p_{\bar{i}j} = (P_i - C_j) \cdot (P_{\bar{i}} - C_j).$$

Another way to remove the rotation matrices from the equations is to take three pictures and to consider the quantity

$$p_{ij} \cdot p_{\bar{i}j} \times p_{\tilde{i}j} = c_{ij} F_j \begin{pmatrix} P_i \\ 1 \end{pmatrix} \cdot c_{ij} F_j \begin{pmatrix} P_{\bar{i}} \\ 1 \end{pmatrix} \times c_{ij} F_j \begin{pmatrix} P_{\tilde{i}} \\ 1 \end{pmatrix}.$$

Again, writing $F_j = \begin{pmatrix} R_j & T_j \end{pmatrix}$, we have

$$\begin{aligned} \gamma_{ij}\gamma_{\bar{i}j}\gamma_{\tilde{i}j} p_{ij} \cdot p_{\bar{i}j} \times p_{\tilde{i}j} &= \begin{pmatrix} R_j & T_j \end{pmatrix} \begin{pmatrix} P_i \\ 1 \end{pmatrix} \cdot \begin{pmatrix} R_j & T_j \end{pmatrix} \begin{pmatrix} P_{\bar{i}} \\ 1 \end{pmatrix} \times \begin{pmatrix} R_j & T_j \end{pmatrix} \begin{pmatrix} P_{\tilde{i}} \\ 1 \end{pmatrix}, \\ &= (R_j P_i + T_j) \cdot (R_j P_{\bar{i}} + T_j) \times (R_j P_{\tilde{i}} + T_j), \\ &= \left(P_i + R_j^T T_j\right) \cdot \left(P_{\bar{i}} + R_j^T T_j\right) \times \left(P_{\tilde{i}} + R_j^T T_j\right). \end{aligned}$$

Letting $C_j = -R_j^T T_j$, we obtain a second type of camera-orientation-free equations:

$$\gamma_{ij}\gamma_{\bar{i}j}\gamma_{\tilde{i}j} p_{ij} \cdot p_{\bar{i}j} \times p_{\tilde{i}j} = (P_i - C_j) \cdot (P_{\bar{i}} - C_j) \times (P_{\tilde{i}} - C_j).$$

Thus, the following equations, which relate the 3D feature points to the picture points without involving the camera orientation, are all consequences of Equations 1:

$$\begin{aligned} \gamma_{ij}\gamma_{\bar{i}j} p_{ij} \cdot p_{\bar{i}j} &= (P_i - C_j) \cdot (P_{\bar{i}} - C_j), \\ \gamma_{ij}\gamma_{\bar{i}j}\gamma_{\tilde{i}j} p_{ij} \cdot p_{\bar{i}j} \times p_{\tilde{i}j} &= (P_i - C_j) \cdot (P_{\bar{i}} - C_j) \times (P_{\tilde{i}} - C_j), \end{aligned}$$

for all $i, \bar{i}, \tilde{i} = 1, \ldots, N$ and all $j = 1, \ldots, J$. The above system is slightly redundant. This is because, for any $v_1, v_2 \in \mathbb{R}^3$ which are not collinear, the set $\{v_1, v_2, v_1 \times v_2\}$ forms a basis for $\mathbb{R}^3$. Thus, assuming that $P_1 - C_j$ and $P_2 - C_j$ are not collinear, all equations written above can be obtained from the following (smaller) set of equations.

$$\begin{aligned} \gamma_{ij}\gamma_{1j} p_{ij} \cdot p_{1j} &= (P_i - C_j) \cdot (P_1 - C_j), \\ \gamma_{ij}\gamma_{2j} p_{ij} \cdot p_{2j} &= (P_i - C_j) \cdot (P_2 - C_j), \\ \gamma_{ij}\gamma_{1j}\gamma_{2j} p_{ij} \cdot p_{1j} \times p_{2j} &= (P_i - C_j) \cdot (P_1 - C_j) \times (P_2 - C_j), \end{aligned} \tag{3}$$

for all $i, \bar{i} = 1, \ldots, N$ and all $j = 1, \ldots, J$.

We can show that this system forms a complete set of camera-orientation free equations, i.e. that solving the above equations for all P_i's and all C_j's is equivalent to solving Equations 1 for all P_i's, all C_j's and all R_j's and forgetting the actual values of the R_j's. To do this, we simply show that Equations 1 are a consequence of Equations 3. We begin by using a fact from invariant theory which states that if some vectors $v_1, \ldots, v_N$ and $w + 1, \ldots, w_N$ satisfy

$$v_i \cdot v_k = w_i \cdot w_k, \text{ for all } i, k = 1, \ldots, N,$$

then there exists an orthogonal matrix A such that $v_i = Aw_i$, for all $i = 1, \ldots, N$. Thus, for every index j, there exists an orthogonal matrix A_j such that

$$(\gamma_j p_{ij}) = A_j(P_i - C_j), \text{ for all } i = 1, \ldots, N.$$

But the determinant of A_j cannot be negative, otherwise we would have

$$\gamma_{ij}\gamma_{1j}\gamma_{2j} p_{ij} \cdot p_{1j} \times p_{2j} = -(P_i - C_j) \cdot (P_1 - C_j) \times (P_2 - C_j),$$

which contradicts the third equation (unless $P_i - C_j$,$P_1 - C_j$ and $P_2 - C_j$ are co-planar.) Hence, each A_j is a rotation matrix and we thus we obtain Equations 1.

The camera orientation are known to be the cause of numerical instability when attempting to solve Equations 1. It is thus expected that Equations 3 will yield a more stable solution for the structure than when attempting to solve for all variables using Equations 1. Indeed, we have shown, experimentally,[13] that numerically minimizing the total square error of these new equations yields a more accurate and stable solution than minimizing the total square error in the standard structure-from-motion formulation as in Lourakis et al.[14] (the so-called *bundle adjustment* method.[15])

Note that other authors have exploited the idea of using a camera-orientation-free formulation For example, the cosine of the angles used in the equations of the so-called *pyramid method*[16] can be obtained by taking the ratio of some of Equations contained in our system. However, the degree-three equations contained in our system cannot be fully recovered from the pyramid method equations.

5. (PURELY)-STRUCTURE-FROM-MOTION EQUATIONS

In this section, we eliminate the camera center coordinates C_j from Equations 3. To do this, we can simply observe that the fact that for every j, there exists R_j and T_j such that

$$\gamma_{ij} p_{ij} = R_j P_{ij} + T_j, \text{ for all} i = 1, \ldots, N,$$

means that the j^{th} point configuration

$$(\gamma_{1j} p_{1j}, \gamma_{2j} p_{2j}, \ldots, \gamma_{Nj} p_{Nj})$$

has the same *shape* and *orientation* as the point configuration

$$P_1, P_2, \ldots, P_N.$$

Obtaining pose-free equations is thus easy using some standard results of invariant theory since the fundamental invariants of the diagonal action of the group of rotations and translations in $\mathbb{R}^3$ are well known.[17] However, we can also obtain the same result using basic algebraic manipulation. For example, the following equations are all contained in the system of equations 3:

$$\begin{aligned}
\gamma_{1j}\gamma_{1j} p_{1j} \cdot p_{1j} &= (P_1 - C_j) \cdot (P_1 - C_j) && (4)\\
\gamma_{2j}\gamma_{2j} p_{2j} \cdot p_{2j} &= (P_2 - C_j) \cdot (P_2 - C_j) && (5)\\
\gamma_{3j}\gamma_{1j}\gamma_{2j} p_{3j} \cdot p_{1j} \times p_{2j} &= (P_3 - C_j) \cdot (P_1 - C_j) \times (P_2 - C_j) && (6)\\
\gamma_{ij}\gamma_{1j} p_{ij} \cdot p_{1j} &= (P_i - C_j) \cdot (P_1 - C_j) \text{ for } i = 2, 3, ...n && (7)\\
\gamma_{ij}\gamma_{2j} p_{ij} \cdot p_{2j} &= (P_i - C_j) \cdot (P_2 - C_j) \text{ for } i = 3, 4, ...n && (8)\\
\gamma_{ij}\gamma_{1j}\gamma_{2j} p_{ij} \cdot p_{1j} \times p_{2j} &= (P_i - C_j) \cdot (P_1 - C_j) \times (P_2 - C_j) \text{ for } i = 4, 5, ...n && (9)
\end{aligned}$$

Observe that

$$(P_i - P_1) \cdot (P_i - P_1)$$

$$= ((P_i - C_j) - (P_1 - C_j)) \cdot ((P_i - C_j) - (P_1 - C_j))$$
$$= (P_i - C_j) \cdot (P_i - C_j) - 2(P_i - C_j) \cdot (P_1 - C_j) + (P_1 - C_j) \cdot (P_1 - C_j), \ for\ i = 2, 3, ...n.$$

The expressions $(P_i - C_j) \cdot (P_1 - C_j)$ and $(P_1 - C_j) \cdot (P_1 - C_j)$ are already contained in the equation set, and the value of $(P_i - C_j) \cdot (P_i - C_j)$ is given by $(P_i - C_j) \cdot (P_1 - C_j) \times (P_2 - C_j)$. So we can replace Equation 7 with

$$(\gamma_{ij} p_{ij} - \gamma_{1j} p_{1j}) \cdot (\gamma_{ij} p_{ij} - \gamma_{1j} p_{1j}) = (P_i - P_1) \cdot (P_i - P_1) \text{ for } i = 2, 3, ...n$$

Similarly we can replace Equation 8 and Equation 9 with

$$\begin{aligned} \|\gamma_{ij} p_{ij} - \gamma_{2j} p_{2j}\|^2 &= \|P_i - P_2\|^2 \text{ for } i = 3, 4, ...n \\ (\gamma_{ij} p_{ij} - gamma_{3j} p_{3j}) \cdot (\gamma_{1j} p_{1j} - \gamma_{3j} p_3 j) \times (\gamma_{2j} p_2 j - \gamma_{3j} p_{3j}) &= (P_i - P_3) \cdot (P_1 - P_3) \times (P_2 - P_3) \text{ for } i = 4, 5, ...n. \end{aligned}$$

Thus we get the following equations, which relate the 3D feature points to the picture points, without referring to the camera pose:

$$\begin{aligned} \|\gamma_{ij} p_{ij} - \gamma_{1j} p_{1j}\|^2 &= \|P_i - P_1\|^2 \\ \|\gamma_{ij} p_{ij} - \gamma_{2j} p_{2j}\|^2 &= \|P_i - P_2\|^2 \\ (\gamma_{ij} p_{ij} - \gamma_{3j} p_{3j}) \cdot (\gamma_{1j} p_{1j} - \gamma_{3j} p_{3j}) \times (\gamma_{2j} p_{2j} - \gamma_{3j} p_{3j}) &= (P_i - P_3) \cdot (P_1 - P_3) \times (P_2 - P_3) \end{aligned} \tag{10}$$

This system forms a complete set of pose-free equations. Indeed we can show that Equations 10 imply Equations 1 by proceeding in a similar fashion as in the previous section. We observe that, for any fixed j, we have

$$\|\gamma_{ij} p_{ij} - \gamma_{\bar{i}j} p_{\bar{i}j}\|^2 = \|P_i - P_{\bar{i}}\|^2, \text{ for any } i, \bar{i} \in 1, \dots, N.$$

We use a fact from invariant theory which states that if vectors $v_1, \dots, v_N$ and $w_1, \dots, w_N$ satisfy

$$\|v_i - v_k\| = \|w_i - w_k\|, \text{ for all } i, k = 1, \dots, N,$$

then there exists an orthogonal matrix A and a translation vector T such that $v_i = Aw_i + T$, for all $i = 1, \dots, N$. Thus, for every index j, there exists an orthogonal matrix A_j and a translation vector T_j such that

$$\gamma_j p_{ij} = A_j P_i + T_j, \text{ for all } i = 1, \dots, N.$$

But the determinant of A_j cannot be negative, otherwise we would have

$$\gamma_{ij} \gamma_{1j} \gamma_{2j} (p_{ij} - p_{3j}) \cdot (p_{1j} - p_{3j}) \times (p_{2j} - p_{3j}) = -(P_i - P_3) \cdot (P_1 - P_3) \times (P_2 - P_3)$$

which contradicts the third equation (unless $P_i - P_3$,$P_1 - P_2$ and $P_i - P_1$ are not co-planar.) Hence, each A_j is a rotation matrix.

6. DEPTH-FROM-MOTION EQUATIONS

The depth of a 3D point P_i with respect to the camera center of picture j is given by the value of γ_{ij}. Thus, in order to obtain a set of *depth-from-motion* equations, we need to eliminate all the variables except the γ_{ij}'s from Equations 1. To do this, we observe that the right-hand sides of all the equations contained in Equations 10 are all independent of j. The the left-hand-sides for different j's must be equal. We thus obtain the following structure-and-camera-pose-free system of equations:

$$\begin{aligned} \|\gamma_{ij} p_{ij} - \gamma_{1j} p_{1j})\|^2 &= \|\gamma_{i\bar{j}} p_{i\bar{j}} - \gamma_{1\bar{j}} p_{1\bar{j}})\|^2 \\ \|(\gamma_{i\bar{j}} p_{i\bar{j}} - \gamma_{2\bar{j}} p_{2\bar{j}})\|^2 &= \|(\gamma_{i\bar{j}} p_{i\bar{j}} - \gamma_{2\bar{j}} p_{2\bar{j}})\|^2 \\ (\gamma_{ij} p_{ij} - \gamma_{3j} p_{3j}) \cdot (\gamma_{1j} p_{1j} - \gamma_{3j} p_{3j}) \times (\gamma_{2j} p_{2j} - \gamma_{3j} p_{3j}) &= (\gamma_{i\bar{j}} p_{i\bar{j}} - \gamma_{3\bar{j}} p_{3\bar{j}}) \cdot (\gamma_{1\bar{j}} p_{1\bar{j}} - \gamma_{3\bar{j}} p_{3\bar{j}}) \times (\gamma_{2\bar{j}} p_{2\bar{j}} - \gamma_{3\bar{j}} p_{3\bar{j}}) \end{aligned} \tag{11}$$

To show that this is a complete set, i.e. that Equations 1 is a consequence of these equations, we can proceed exactly as in the previous section.

This system contains (3N-5)(J-1) equations in N unknowns. However, it is homogeneous, so its solution is only defined up to a global scale factor. Since all γ_{ij}'s are strictly positive, we can set one of them, say γ_{11}, equal to one and solve for the remaining ones.

To obtain the smallest possible system of equations, we can consider J=2 picture (say j and $\bar{j}$) and N=5 points (say $i = 1, 2, 3, 4, 5$) on each picture. We then have the following set of equations

$$\begin{aligned}
\|\gamma_{1j}p_{1j} - \gamma_{2j}p_{2j}\|^2 &= \|\gamma_{1\bar{j}}p_{1\bar{j}} - \gamma_{2\bar{j}}p_{2\bar{j}}\|^2 \\
\|\gamma_{1j}p_{1j} - \gamma_{3j}p_{3j}\|^2 &= \|\gamma_{1\bar{j}}p_{1\bar{j}} - \gamma_{3\bar{j}}p_{2\bar{j}}\|^2 \\
\|\gamma_{3j}p_{3j} - \gamma_{2j}p_{2j}\|^2 &= \|\gamma_{3\bar{j}}p_{3\bar{j}} - \gamma_{2\bar{j}}p_{2\bar{j}}\|^2 \\
\|\gamma_{4j}p_{4j} - \gamma_{1j}p_{1j}\|^2 &= \|\gamma_{4\bar{j}}p_{4\bar{j}} - \gamma_{1\bar{j}}p_{1\bar{j}}\|^2 \\
\|\gamma_{4j}p_{4j} - \gamma_{2j}p_{2j}\|^2 &= \|\gamma_{4\bar{j}}p_{4\bar{j}} - \gamma_{2\bar{j}}p_{2\bar{j}}\|^2 \\
\|\gamma_{5j}p_{5j} - \gamma_{1j}p_{1j}\|^2 &= \|\gamma_{5\bar{j}}p_{5\bar{j}} - \gamma_{1\bar{j}}p_{1\bar{j}}\|^2 \\
\|\gamma_{5j}p_{5j} - \gamma_{2j}p_{2j}\|^2 &= \|\gamma_{5\bar{j}}p_{5\bar{j}} - \gamma_{2\bar{j}}p_{2\bar{j}}\|^2 \\
(\gamma_{4j}p_{4j} - \gamma_{3j}p_{3j}) \cdot (\gamma_{1j}p_{1j} - \gamma_{3j}p_{3j}) \times (\gamma_{2j}p_{2j} - \gamma_{3j}p_{3j}) &= (\gamma_{4\bar{j}}p_{4\bar{j}} - \gamma_{3\bar{j}}p_{3\bar{j}}) \cdot (\gamma_{1\bar{j}}p_{1\bar{j}} - \gamma_{3\bar{j}}p_{3\bar{j}}) \times (\gamma_{2\bar{j}}p_{2\bar{j}} - \gamma_{3\bar{j}}p_{3\bar{j}}) \\
(\gamma_{5j}p_{5j} - \gamma_{3j}p_{3j}) \cdot (\gamma_{1j}p_{1j} - \gamma_{3j}p_{3j}) \times (\gamma_{2j}p_{2j} - \gamma_{3j}p_{3j}) &= (\gamma_{5\bar{j}}p_{5\bar{j}} - \gamma_{3\bar{j}}p_{3\bar{j}}) \cdot (\gamma_{1\bar{j}}p_{1\bar{j}} - \gamma_{3\bar{j}}p_{3\bar{j}}) \times (\gamma_{2\bar{j}}p_{2\bar{j}} - \gamma_{3\bar{j}}p_{3\bar{j}})
\end{aligned}$$

Alternatively, we can write a slightly simpler system, such as

$$\begin{aligned}
\|\gamma_{1j}p_{1j} - \gamma_{2j}p_{2j}\|^2 &= \|\gamma_{1\bar{j}}p_{1\bar{j}} - \gamma_{2\bar{j}}p_{2\bar{j}}\|^2 \\
\|\gamma_{1j}p_{1j} - \gamma_{3j}p_{3j}\|^2 &= \|\gamma_{1\bar{j}}p_{1\bar{j}} - \gamma_{3\bar{j}}p_{3\bar{j}}\|^2 \\
\|\gamma_{2j}p_{2j} - \gamma_{3j}p_{3j}\| &= \|\gamma_{2\bar{j}}p_{2\bar{j}} - \gamma_{3\bar{j}}p_{3\bar{j}}\|^2 \\
\|\gamma_{1j}p_{1j} - \gamma_{4j}p_{4j}\| &= \|\gamma_{1\bar{j}}p_{1\bar{j}} - \gamma_{4\bar{j}}p_{4\bar{j}}\|^2 \\
\|\gamma_{2j}p_{2j} - \gamma_{4j}p_{4j}\| &= \|\gamma_{2\bar{j}}p_{2\bar{j}} - \gamma_{4\bar{j}}p_{4\bar{j}}\|^2 \\
\|\gamma_{3j}p_{3j} - \gamma_{4j}p_{4j}\| &= \|\gamma_{3\bar{j}}p_{3\bar{j}} - \gamma_{4\bar{j}}p_{4\bar{j}}\|^2 \\
\|\gamma_{1j}p_{1j} - \gamma_{5j}p_{5j}\| &= \|\gamma_{1\bar{j}}p_{1\bar{j}} - \gamma_{5\bar{j}}p_{5\bar{j}}\|^2 \\
\|\gamma_{2j}p_{2j} - \gamma_{5j}p_{5j}\| &= \|\gamma_{2\bar{j}}p_{2\bar{j}} - \gamma_{5\bar{j}}p_{5\bar{j}}\|^2 \\
\|\gamma_{3j}p_{3j} - \gamma_{5j}p_{5j}\| &= \|\gamma_{3\bar{j}}p_{3\bar{j}} - \gamma_{5\bar{j}}p_{5\bar{j}}\|^2
\end{aligned}$$

In a generic situation, the only difference between this system and the previous system is that it has two solutions instead of one, the solutions being related by a reflection. This system has the advantage that it is only of degree two. Setting $\gamma_{1j} = 1$ and hiding the variable γ_{2j}, we can obtain the sparse resultant using Emiris and Canny's method.[18] The resultant matrix is a square matrix of size 31,519. However, the determinant of this matrix turns out to be trivially equal to zero and so we cannot solve for γ_{2j} directly from it. But following the approach described by Emiris in Canny,[18] we can consider the maximum minor of this matrix and use it in place of the resultant.

Acknowledgments

The second author would like to thank Jie Shan for stimulating discussions. This work was funded in parts by NSF grant 0434398.

REFERENCES

1. I. M. Gel′fand, M. M. Kapranov, and A. V. Zelevinsky, *Discriminants, resultants, and multidimensional determinants*, Mathematics: Theory & Applications, Birkhäuser Boston Inc., Boston, MA, 1994.
2. A. Wallack, I. Z. Emiris, and D. Manocha, "MARS: a MAPLE/MATLAB/C resultant-based solver," in *ISSAC 98: Proceedings of the 1998 International Symposium on Symbolic and Algebraic Computation, August 13–15, 1998, University of Rostock, Germany*, O. Gloor, ed., pp. 244–251, ACM Press, (pub-ACM:adr), 1998.

3. M. Minimair, "MR: Macaulay Resultant package for Maple," *ACM SIGPLAN Notices* **39**, pp. 26–29, Apr. 2004.
4. V. L. G.-M. Greuel and H. Schönemann, "SINGULAR::PLURAL 2.1," A Computer Algebra System for Noncommutative Polynomial Algebras, Centre for Computer Algebra, University of Kaiserslautern, 2003. `http://www.singular.uni-kl.de/plural`.
5. A. Dickenstein and I. Z. Emiris, eds., *Solving polynomial equations*, vol. 14 of *Algorithms and Computation in Mathematics*, Springer-Verlag, Berlin, 2005. Foundations, algorithms, and applications.
6. E. Bézout, "Recherches sur le degré des équations résultantes de l'évanouissement des inconnues et sur le moyen qu'il convient d'employer pour trouver les équations," *Mem. Acad, Paris* , pp. 288–338, 1764.
7. B. Buchberger, "An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal," *J. Symbolic Comput.* **41**(3-4), pp. 475–511, 2006. Translated from the 1965 German original by Michael P. Abramson.
8. J.-C. Faugére, "A new efficient algorithm for computing Gröbner bases (F_4)," *J. Pure Appl. Algebra* **139**(1-3), pp. 61–88, 1999. Effective methods in algebraic geometry (Saint-Malo, 1998).
9. Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry, *An Invitation to 3-D Vision: From Images to Geometric Models*, SpringerVerlag, 2003.
10. M.-A. Ameller, L. Quan, and B. Triggs, "Camera pose revisited: New linear algorithms," *Mach. Intell.* **16**(8), pp. 802–808, 2002.
11. C. Fermüller and Y. Aloimonos, "Observability of 3D motion," *Int. J. Comput. Vision* **37**(1), pp. 43–63, 2000.
12. M. Boutin, J. Zhang, and D. G. Aliaga, "Improving the numerical stability of structure from motion by algebraic elimination," in *IS&T/SPIE joint symposium, Computational Imaging IV conference*, (San Jose, CA), January 2006.
13. J. Zhang, M. Boutin, and D. Aliaga, "Robust bundle adjustment for structure from motion," in *ICIP '06: Proceedings of the IEEE Conference on Image Processing*, (Atlanta), 2006.
14. M. Lourakis and A. Argyros, "The design and implementation of a generic sparse bundle adjustment software package based on the levenberg-marquardt algorithm," Tech. Rep. 340, Institute of Computer Science - FORTH, Heraklion, Crete, Greece, Aug. 2004. Available from `http://www.ics.forth.gr/~lourakis/sba`.
15. B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment - a modern synthesis," in *ICCV '99: Proceedings of the International Workshop on Vision Algorithms*, pp. 298–372, Springer-Verlag, (London, UK), 2000.
16. W. Zhizhuo, *Principles of Photogrammetry (with remote sensing)*, Press of Wuhan Technical University of Surveying and Mapping, Publishing House of Surveying and Mapping, 1990. Translated by Cheng Maorong.
17. P. J. Olver, *Classical invariant theory*, vol. 44 of *London Mathematical Society Student Texts*, Cambridge University Press, Cambridge, 1999.
18. I. Z. EMIRIS and J. F. CANNY, "Efficient incremental algorithms for the sparse resultant and the mixed volume," *J. Symbolic Computation* **20**(2), pp. 117–149, 1995.